ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

УДК 519.1

СОГЛАСОВАНО

Научный руководитель
Доцент кафедры технологий
моделирования сложных систем
к. ф.-м. наук

_____ Ю. А. Янович
«_____» _____ 2022 г.

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии канд. техн. наук

_____ В. В. Шилов
«_____» _____ 2022 г.

Выпускная квалификационная работа
(академическая)

на тему: **Распутывание биткоин-транзакций**

по направлению подготовки 09.03.04 «Программная инженерия»

ВЫПОЛНИЛ

студент группы БПИ184
образовательной программы
09.03.04 «Программная инженерия»

_____ А. Д. Курылев
«_____» _____ 2022 г.

Москва 2022

Реферат

В этом исследовании представлен анализ транзакций биткоина с самого начала блокчейна. Все транзакции классифицированы и разделены на 4 класса: простые, разделимые, неоднозначные, трудные. После классификации проведен анализ объема классов транзакций по времени, с чем может быть связано это распределение и какова динамика этих классов. Одна из целей исследования — разработать и опубликовать алгоритм распутывания Shared Send транзакций. Это требует преобразования транзакции в граф, ее упрощения, удаления повторяющихся входных и выходных данных и пересчета значений. Написание и тестирование алгоритма распутывания биткоин-транзакций сопровождается его публикацией в открытых источниках.

Данная работа состоит из 30 страниц, 3 глав, 23 рисунков. Использовано 17 источников.

Ключевые слова: биткоин; блокчейн; UTXO; транзакция; криптовалюта; анализ данных; большие данные.

Abstract

This study presents an analysis of bitcoin transactions from the very beginning of the blockchain. All transactions are classified into 4 classes: simple, separable, ambiguous, difficult. After the classification, an analysis was made of the volume of transaction classes over time, what this distribution may be related to and what is the dynamics of these classes. One of the goals of the study is to develop and publish an algorithm for disentangling Shared Send transactions. This requires transforming the transaction into a graph, simplifying it, removing duplicate inputs and outputs, and recalculating values. Writing and testing an algorithm for disentangling bitcoin transactions is accompanied by its publication in open sources.

The paper contains 30 pages, 3 chapters, 23 figures. 17 sources are used.

Keywords: bitcoin; blockchain; shared send; untangling; UTXO; transaction; cryptocurrency; data analysis; big data.

Используемые определения и термины

Shared send – метод запутывания истории монет с помощью соединения нескольких транзакций в одну большую транзакцию.

UTXO – Unspent Transaction Output, выход транзакции, который можно использовать в качестве входа новой транзакции, представляет собой некоторое количество криптовалюты, предоставленное одним аккаунтом для того чтобы быть потраченным другим аккаунтом.

Биткоин – первая блокчейн сеть и крупнейшая по капитализации криптовалюта.

Блокчейн – децентрализованное хранилище данных с устойчивым к изменениям журналом транзакций и встроенными инструментами для аудита, в котором информация сохраняется в транзакциях, сгруппированных в блоки, а блоки соединяются в последовательность (цепь) с помощью криптографии.

Криптовалюта – любая форма валюты, существующая в цифровом виде и использующая криптографические методы для обеспечения безопасности транзакций.

Криптография – наука о методах обеспечения конфиденциальности, целостности данных, аутентификации и шифровании.

Транзакция – минимальная операция, которая имеет смысл и может быть выполнена только полностью.

Требования AML – Anti-Money Laundering, противодействие отмыванию денег, требования включают в себя идентификация, хранение и обмен информации о пользователях, их доходах и транзакциях между организациями и ведомствами.

Требования KYC – Know Your Customer, знай своего клиента, принцип работы, обязывающий финансовые инструменты идентифицировать личность своих клиентов перед проведением операций.

Содержание

Введение

За последние полтора десятилетия криптовалюты и связанные с ними технологии получили колоссальную популярность во всем мире. Криптографические технологии, алгоритмы и задачи постоянно совершенствуются, идет непрерывная гонка между методами защиты и взлома информации. Первой, и наиболее популярной сегодня криптовалютой является биткоин, принцип работы которого был описан Сатоши Накамото в 2008 году [15]. Она насчитывает более 82 миллионов кошельков, около 270 тысяч транзакций совершается ежедневно. Это псевдо-анонимная криптовалюта, информация о транзакция которой хранится в публичной распределенной сети с использование блокчейн технологии. Связи между биткоин-адресами и фактическими пользователями (группой пользователей, организацией, биржей и т.д.) приватны. Из-за такой особенности биткоин блокчейна в сети существует некоторое количество биткоин адресов, через которые проходит криптовалюта, оплачивающая криминальные услуги. Так, в 2020 году Европол на конференции IOCTA представил [10] результаты анализа сети биткоин, и опубликовал данные по криминальным транзакциям: 1.1% от общего числа транзакций приходится на нелегальную активность. Во время последней на текущий момент конференции IOCTA 2021 [11] была поставлена задача на более серьезные контроль и регулирование криптовалютной сферы.

Исходя из информации, представленной в данных отчетах за последние несколько лет, как у компаний, работающих с биткоином, так и у государств, существует большая потребность в более открытой, понятной и прослеживаемой системе криптовалютных транзакций, чем есть сейчас. Глобально, существует несколько способов скрыть реальные данные транзакции: использование анонимных криптовалют, например Monero; проведение обмена через биржи, которые не в полном объеме соблюдают принцип KYC, или децентрализованной биржи. Эти способы связаны скорее не с биткоином, а с технологиями криптовалют в целом. В моем исследовании основной фокус будет направлен на методы повышения приватности, работающие непосредственно с биткоином, с его ограничениями и особенностями.

Известно два базовых метода повышения приватности путем запутывания транзакций: это Coin Mixers сервисы и Shared Send транзакции. Первый метод основывается на работе сервисов, которые принимают от пользователя криптовалюту с указанием реального адреса. Через случайное время случайными разбиениями средств данный сервис отправляет запрошенный платеж на реальный адрес получателя. Если сервисом пользуется достаточно большое количество человек, и в нем корректно выстроен алгоритм передачи платежа, такую транзакцию становится практически невозможно отследить. Внутри сервиса путаются реальные отправители и получатели, количество криптовалюты и время отправки. Однако такие сервисы требуют доверия к себе для проведения операций, и, зачастую, комиссию за сделки.

Вторым вариантом повышения приватности внутри сети биткоин является метод Shared Send транзакций. Он основывается на алгоритме CoinJoin [13], а именно на объединении некоторого количества людей и их платежей в одну транзакцию с большим количеством входов и выходов. Внутри транзакции явно не указано, какой вход соответствует какому выходу, что затрудняет привычное

прямое отслеживание денежных потоков от отправителя к получателю. Целью данной работы является разработка и публикация алгоритма, позволяющего распутывать Shared Send транзакции в сети биткоина. Также необходимо, во-первых, извлечь и предобработать транзакции из биткоин блокчейна, во-вторых, научиться выявлять транзакции, которые стали результатом Shared Send метода, в-третьих, предоставить анализ работы алгоритмов детектирования и распутывания транзакций, а также классификацию всех биткоин транзакций.

Глава 1. Обзор источников

1.1. Алгоритмы запутывания

В данной главе не будет рассматриваться такой большой сегмент алгоритмов, как "алгоритмы, требующие вмешательства в систему биткоин". Существует множество статей, которые предлагают более совершенные, анонимные или удобные методы запутывания, чем разобранные далее, но при этом эти алгоритмы требуют существенной модификации биткоин сети, что делает их неприменимыми в текущей реальности. Это однозначно теоретические исследования. В связи с чем они не будут подробно рассматриваться в этой работе.

1.1.1. CoinJoin

В 2013 году Максвелл в своем тексте [13] представил концепцию алгоритма CoinJoin, в основе которой лежало важное замечание: участие различных биткоин-адресов в одной транзакции не доказывает общий контроль над этими адресами, что, по словам Максвелла, и делает возможным CoinJoin.

Подписи внутри транзакции, по одной на вход, полностью независимы друг от друга. Это означает, что пользователи биткоин могут договориться о наборе входных данных для расходов и наборе выходных данных для оплаты, а затем индивидуально и отдельно подписать транзакцию, после чего объединить свои подписи. Транзакция недействительна и не будет принята сетью, пока не будут предоставлены все подписи, и никто не подпишет транзакцию, которая им не нравится.

Чтобы использовать такой метод для повышения конфиденциальности, некоторое число пользователей (их количество обозначим N) должны договориться об одинаковом размере выходных данных и предоставить входные данные, по крайней мере, этого размера. Транзакция будет иметь N выходов такого же размера и, возможно, еще N выходов сдачи, если некоторые пользователи введут больше, чем нужно. Все подпишут транзакцию, и тогда транзакция может быть передана. В такой схеме отсутствует риск кражи на любом этапе, так как пользователь видит, что конкретно он подписывает.

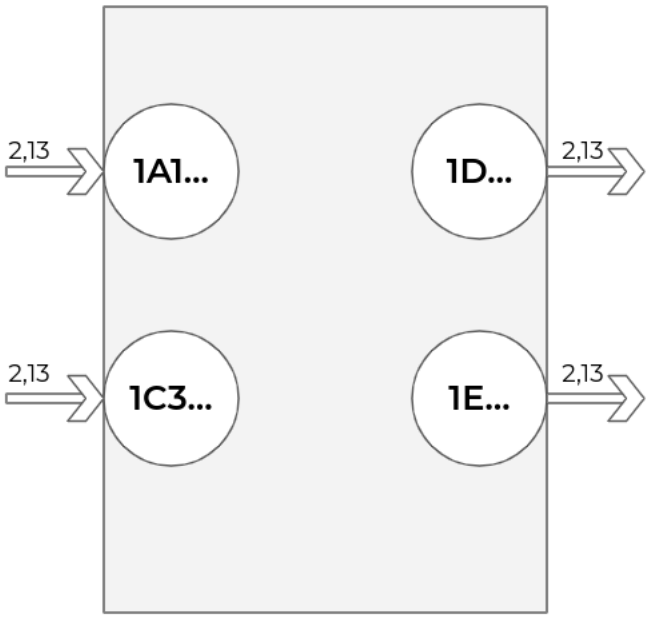


Рис. 1 — Совместная транзакция

На иллюстрации *рис. 1* присутствуют входы от 1A1 и 1C3. Скажем, мы считаем, что 1A1 — это адрес, используемый Алисой, а 1C3 — адрес, используемый Чарли. Кому из Алисы и Чарли принадлежит выход 1D, а кому 1E? Наверняка сказать нельзя.

Эту идею можно использовать и по-другому. Если нужно совершить платеж, можно найти кого-то еще, кто также хочет произвести платеж, и сделать совместный платеж. Это не сильно повышает конфиденциальность, но на самом деле делает транзакцию меньше и, следовательно, проще в сети, а также снижает комиссию. Дополнительная конфиденциальность - это преимущество.

Максвелл утверждает, что такая транзакция внешне неотличима от транзакции, созданной посредством обычного использования. Из-за этого, если эти транзакции станут широко распространенными, они улучшат конфиденциальность даже людей, которые их не используют, потому что совместная транзакция больше не будет веским доказательством общего контроля.

Автор говорит, что возможны многие варианты этой идеи, и все они могут сосуществовать, потому что эта идея не требует изменений в системе биткоин.

1.1.2. CoinSwap

Спустя несколько месяцев после публикации алгоритма CoinJoin Максвелл представил [14] протокол, в котором одна сторона (Алиса) может заплатить другой (Боб) через третью (Кэрол), а третья не может их ограбить. Протокол требует четырех опубликованных транзакций, но транзакции выглядят как обычные транзакции условного депонирования 2 из 2 в обычном случае, когда все честны. Если Алиса или Кэрол попытаются нарушить свою часть протокола, он либо полностью раскрутится, либо будут опубликованы дополнительные транзакции, чтобы протолкнуть транзакцию, но без конфиденциальности.

Ключевой концепцией этого протокола является понимание того, что транзакции могут быть защищены знанием прообраза хэша. Мы формируем защищенность транзакции хэшем в предложенном здесь протоколе, но на самом деле он не используется, если только кто-то не пытается обмануть.

Общая идея состоит в том, что Алиса и Кэрол формируют условное депонирование 2 из 2 с монетами Алисы, а Кэрол и Боб формируют условное депонирование 2 из 2 с монетами Кэрол. Эти договоренности имеют слой предварительно рассчитанных выплат по тайм-ауту на случай, если какая-либо из сторон исчезнет. Затем они устанавливают набор взаимоподписанных транзакций погашения условного депонирования, которые имеют общий хэш-замок, но не используют его. Делая это, Кэрол может быть уверена, что если Бобу заплатят, Кэрол тоже заплатят. Как только Кэрол убеждается, что ее нельзя обмануть, она передает свое условное депонирование Бобу, а затем Алиса передает свое условное депонирование Кэрол.

В результате получается несколько сложный протокол просто потому, что он состоит из многих этапов. Из-за этого это подробно объясняется на диаграмме протокола ниже (см. *рис. 2*).

Протокол CoinSwap: В протоколе предполагается, что все стороны имеют частные каналы связи.

Фаза 0. Устанавливает условное депонирование и их возврат по тайм-ауту.

Фаза 1. Делает так, что если Бобу заплатят, то Кэрол не сможет не заплатить.

Фаза 2. Просто освобождает условное депонирование напрямую, потому что все довольны тем, что обман невозможен.

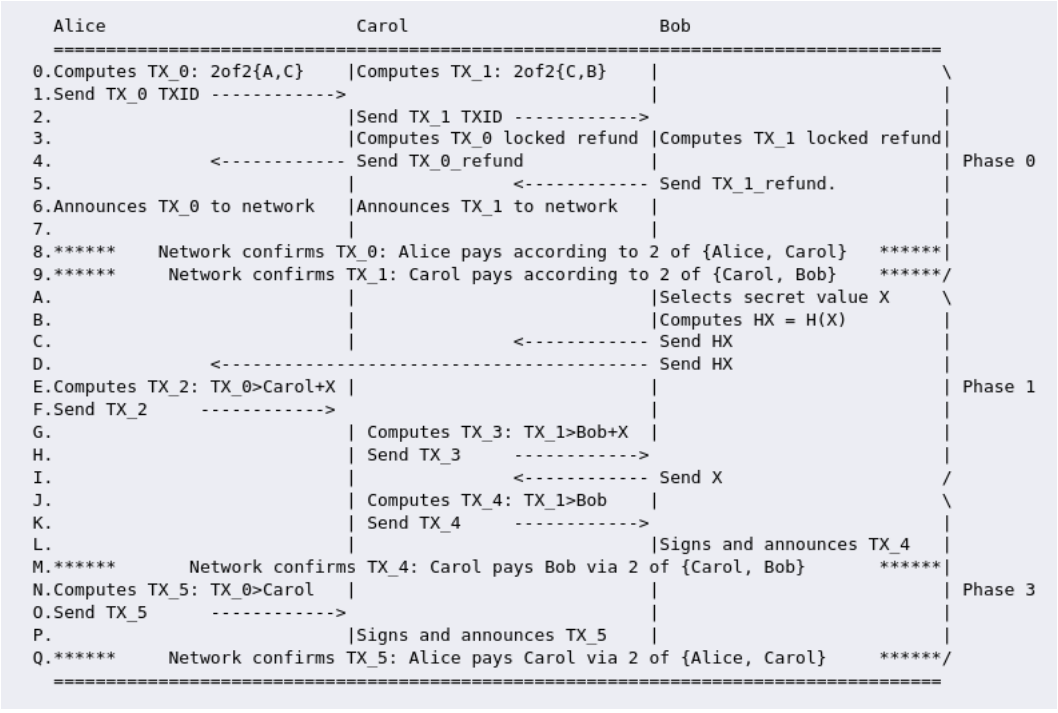


Рис. 2 — Диаграмма протокола CoinSwap

Сравнение с CoinJoin: Максвелл утверждает, что между CoinSwap и CoinJoin есть ряд сравнительных преимуществ и недостатков.

Транзакции CoinJoin эффективны — когда люди объединяются, они могут даже сэкономить немного места по сравнению с обычной транзакцией. В отличие от этого, для CoinSwap требуется как

минимум четыре транзакции, хотя два CoinSwap могут эффективно выполняться одновременно. Это означает, что CoinJoin — это то, что кошельки могут разумно использовать для многих транзакций, в то время как CoinSwap должен быть своего рода периодическим процессом.

Набор анонимности CoinJoin равен количеству участников транзакции (или каскада транзакций). Набор анонимности CoinSwap — это все операции CoinSwap, происходящие одновременно, даже если пользователи не взаимодействуют друг с другом.

Транзакции CoinSwap выглядят как обычные транзакции условного депонирования 2 из 2. Если условное депонирование 2 из 2 станет обычным явлением, то транзакции CoinSwap могут быть менее идентифицируемыми, чем крупные транзакции CoinJoin с кучей выходов одинакового размера, и, следовательно, более устойчивыми к обнаружению.

Для обеспечения конфиденциальности в транзакции CoinJoin должно быть много участников. Это несколько усложняет разработку программного обеспечения. Если Алиса играет роль Боба в CoinSwap, это двухсторонний протокол, что может упростить его реализацию, хотя он включает примерно в восемь раз больше операций по сравнению с максимально простым CoinJoin.

CoinSwaps может происходить между цепочками. CoinJoin по своей сути являются операциями с одной цепочкой.

Можно создавать криптографически сложный CoinJoin, где никто не узнает, чей выход является чьим входом (за исключением владельца каждого выхода). CoinSwap приводит к тому, что участники знают о связи.

1.1.3. MixCoin

В 2014 году группа исследователей, имея ввиду уже существующие и используемые алгоритмы микширования монет создала MixCoin [7], стратегия которого заключается в том, чтобы, опираясь на существующий феномен миксов, добавить независимый криптографический уровень подотчетности. Большая часть модификации их алгоритма состоит в следующем:

Подотчетность. Сервисы Mixcoin выдают пользователям подписанные гарантии, которые гласят: "Если Алиса пошлет мне v монет к моменту времени t_1, я отправлю v монет обратно ей к моменту t_2". После этого пользователь может уверенно отправлять средства сервису, зная, что если последний поведет себя неправильно, он может опубликовать эту гарантию, нанеся ущерб репутации сервиса и, предположительно, его бизнес-модели.

Случайная оплата услуг сервисов. Авторы показали, как оплата услуг сервисов стимулирует честное поведение, однако фиксированная плата подрывает анонимность, когда монеты смешиваются несколько раз. Вместо этого применяется рандомизированная плата, при которой сервисы сохраняют всю стоимость от небольшого процента транзакций. В статье показывается, как генерировать необходимую случайность справедливым и подотчетным образом используя непредсказуемость самой цепочки блоков биткоина.

Неразличимость миксов. Хотя пользователи взаимодействуют со специальными сервисами, одноразовые адреса обеспечивают следующее свойство: сторонние наблюдатели не могут определить,

с каким сервисом взаимодействует пользователь. В этом случае набор анонимности представляет собой множество всех пользователей, одновременно взаимодействующих с любым сервисом.

Микс-сети для биткойна. Против активного злоумышленника, который может нарушить неразличимость миксов, авторы опираются на опыт анонимных коммуникационных сетей анонимного общения, чтобы продемонстрировать, как объединение нескольких миксов в цепочку может по-прежнему обеспечивать сильную анонимность.

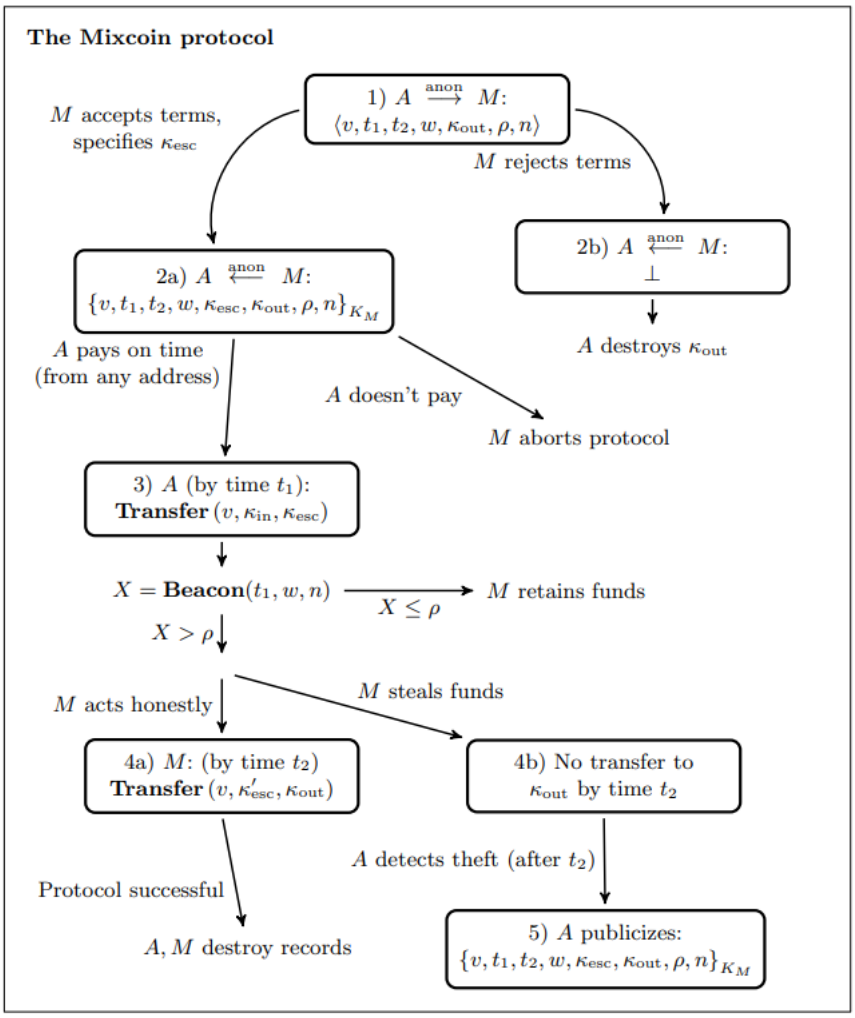


Рис. 3 — Протокол работы MixCoin

Основной протокол, изображенный на *рис. 3* - это общая конструкция, позволяющая клиентам и сервисам указывать множество различных параметров. Авторы ожидают, что, поскольку анонимность любит общность, эти параметры будут сходиться к глобальным значениям. В частности, ожидается, что смешивание будет завершено за несколько часов с комиссией за смешивание менее 1%.

1.2. Инструменты распутывания

1.2.1. Crystal Blockchain

Сервис Crystal Blockchain [9] обеспечивает анализ и мониторинг криптовалютных транзакций на блокчейне, предлагая биржам, банкам и финансовым учреждениям решения по соблюдению требований AML и управлению рисками. Как утверждают создатели данной системы, Crystal анализирует и отслеживает 98% транзакций блокчейна. Данные в режиме реального времени обеспечивают ясность и точность, помогая снизить риски и принимать решения, основанные на соответствии требованиям. Crystal ранжирует риски и уведомляет о потенциально нестандартных действиях. Компания помогает проводить расследования, аудиты и исследовательские отчеты. Они следят за изменениями в региональных и глобальных нормативных актах, чтобы обеспечить их соблюдение. Визуализация работы системы изображена на *рис. 4*.

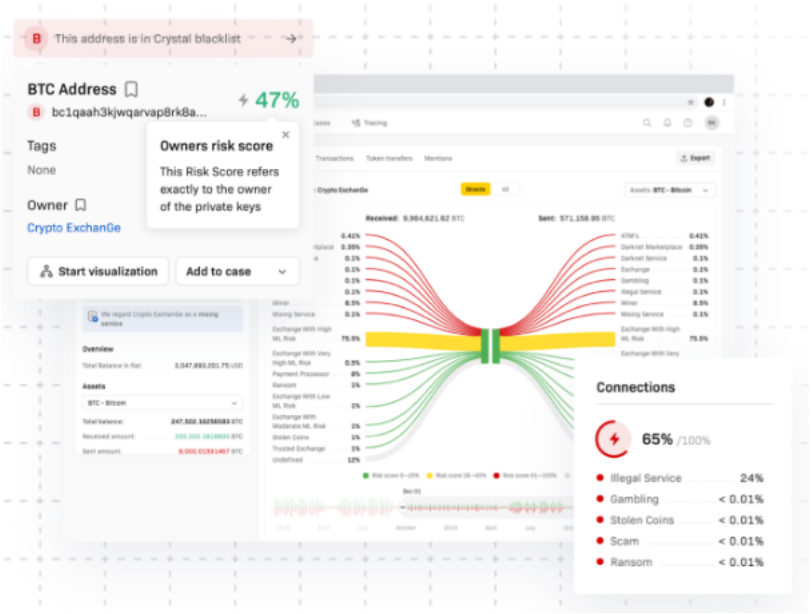


Рис. 4 — Визуализация с сайта crystalblockchain.com

Crystal Blockchain работает с банками и финансовыми институтами, государственными организациями, платежными сервисами, предоставляя контроль за соблюдением требований AML, глобального и регионального законодательства, а также защищенности клиентских данных.

1.2.2. Chainalysis

Chainalysis [8] предоставляет услуги по анализу блокчейн сети, благодаря чему, как утверждается у них на сайте, можно установить связь между биткоин кошельком и реальной личностью, что поможет узнать реальный источник валюты и оценить риск контрагента. Chainalysis используется более 700 компаниями в более чем 70 странах для защиты клиентов, расследования финансовых преступлений и соблюдения нормативных требований. Этот сервис работает преимущественно с финансовыми учреждениями, криптовалютными компаниями и государственными структурами.

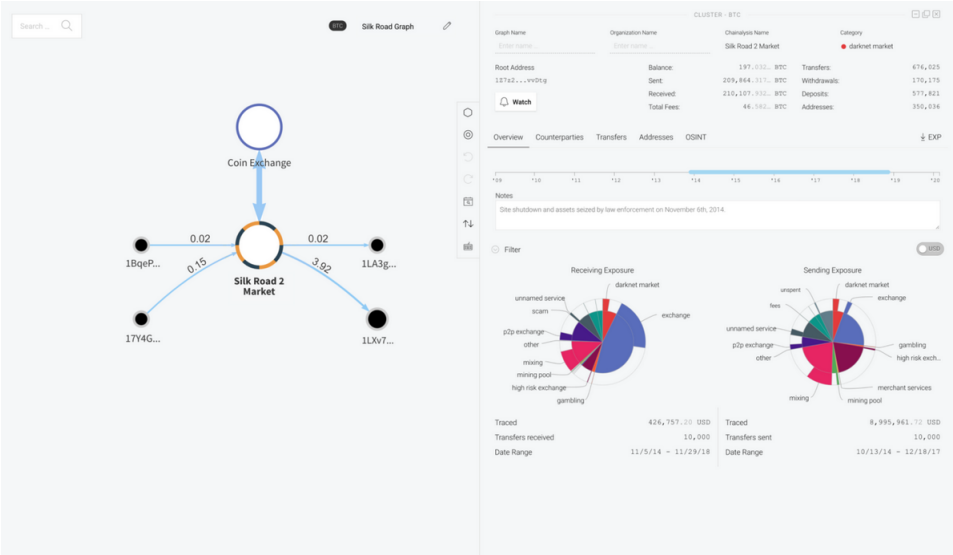


Рис. 5 — Пример работы Chainalysis с кейсом Silk Road

Благодаря инструменту Chainalysis агенты правохранительных органов произвели крупнейшую конфискацию криптовалютных активов в размере 1 миллиарда долларов. В результате работы алгоритмов системы Chainalysis Reactor были выявлены и доказаны зависимости между крупнейшими криптовалютными кошельками даркнет сервиса Silk Road. На *рис. 5* представлена визуализация с сайта chainalysis.

Выводы по главе

Как заметно из описания вышепредставленных алгоритмов их же авторами, изначально они создаются для повышения анонимности между пользователями блокчейна сети биткоин. В данной работе я буду называть Shared Send транзакциями те транзакции, которые совершались по принципу, описанному в пункте 1.1.1. К сожалению, с большей анонимностью и запутанностью сети биткоин повышается и число нелегальных пользователей, которые пользуются повышенной конфиденциальностью для сокрытия своих транзакций от финансовых систем или государств. Используя информацию, полученную от самих авторов статей, которые в некоторых случаях делают важные замечания и оговорки по поводу работы своих алгоритмов, можно попробовать снизить анонимность транзакций, искусственно повышенную применением подобных методов. Это сильно поможет в стабилизации надежности биткоина и повысит раскрываемость преступлений.

Глава 2. Теоретические результаты

2.1. Предобработка транзакций

2.1.1. Отсечение лишних данных

В транзакции, включенной в сеть биткоин, содержится много информации. Для данной работы часть этих данных будет лишней, будет расходовать память системы и не использоваться. В связи с этим было принято решение сокращать хранимые в памяти данные о транзакции. Было принято решение хранить транзакции в блоках, за каждым блоком закреплять время его создания, что поможет в дальнейшем при анализе данных, а также количество хранимых внутри транзакций. В хранимой версии транзакции внутри блока останутся только следующие значения:

- Количество входных адресов
- Входные адреса
- Стоимость входных адресов
- Количество выходных адресов
- Выходные адреса
- Стоимость выходных адресов

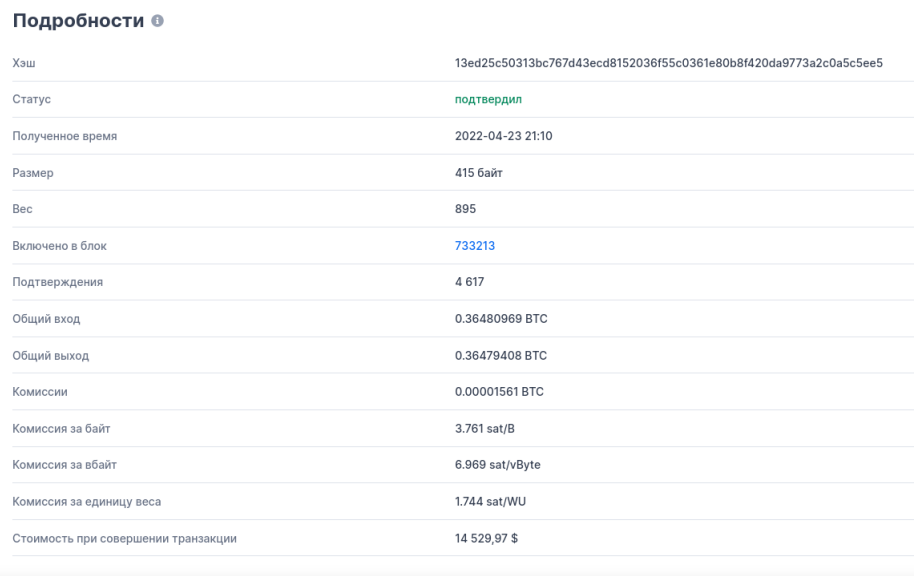
Разность между суммами стоимостей входных и выходных адресов будет комиссией и не будет учитываться в алгоритме. Для примера рассмотрим следующий блок и транзакцию в нем:

Номер блока: 733213

Хэш транзакции: $13ed25c50313bc767d43ecd8152036f55c0361e80b8f420da9773a2c0a5c5ee5$

Хэш	00000000000000000000727f8c4f910699f81e9f4ee4526316cdb347b4f3c6f20
Подтверждения	4 617
Отметка времени	2022-04-23 21:30
Высота	733213
Майнер	Poolin
Количество транзакций	3 085
Сложность	28 225 928 151 211,10
Корень Меркла	d39a397e2aec9b6248346d886b1a8230c1911dcd83eb85f347b39f260e875c1e
Версия	0x20000004
Биты	386 529 497
Вес	3 999 487 WU
Размер	1 538 761 bytes
Nonce (одноразовый код)	1 372 416 824
Объем транзакции	125480.62442808 BTC
Вознаграждение за блок	6.25000000 BTC
Вознаграждение комиссии	0.11510335 BTC

Рис. 6 — Пример блока



Подробности

Хэш	13ed25c50313bc767d43ecd8152036f55c0361e80b8f420da9773a2c0a5c5ee5
Статус	подтвердил
Полученное время	2022-04-23 21:10
Размер	415 байт
Вес	895
Включено в блок	733213
Подтверждения	4 617
Общий вход	0.36480969 BTC
Общий выход	0.36479408 BTC
Комиссии	0.00001561 BTC
Комиссия за байт	3.761 sat/B
Комиссия за вбайт	6.969 sat/vByte
Комиссия за единицу веса	1.744 sat/WU
Стоимость при совершении транзакции	14 529,97 $

Рис. 7 — Пример транзакции

Для просмотра данных будем использовать сервис [6]. Как видно (*рис. 6* и *рис. 7*), в блоке и транзакции указано много лишних для текущего эксперимента данных. После процедуры отсечения лишней информации останется следующее:

Время создания блока: 1650681000

Количество транзакций в блоке: 3085

Входной адрес 1: $bc1qwfgdjyy95aay2686fn74h6a4nu9eev6np7q4fn204dkj3274frlqrskvx0$

Стоимость 1: 36480969

Выходной адрес 1: $3NYbY7nAxMuw8GWtqs96yXKprGsyVvsx8z$

Стоимость 1: 2512000

Выходной адрес 2: $17ewxY9kMiWwN2ajaN7xBPaJAwdUZM8DKX$

Стоимость 2: 21361

Выходной адрес 3: $bc1qwfgdjyy95aay2686fn74h6a4nu9eev6np7q4fn204dkj3274frlqrskvx0$

Стоимость 3: 33946047

Необходимо отметить единицы измерения некоторых полей. Так, время создания блока указано в формате Unix Time, а стоимость входов и выходов - в сатоши, минимальной доле биткоина.

2.1.2. Упрощение транзакции

Упрощением транзакции будем называть операцию преобразования транзакции по следующему алгоритму:

1) Объединить все входы, принадлежащие одному адресу

2) Объединить все выходы, принадлежащие одному адресу

3) Если во множестве входов есть адрес, который также есть и во множестве выходов, необходимо вычесть величину выхода из величины входа

4) Если результат предыдущего действия положительный, удалить соответствующий выход, а на входе заменить его величину модулем разности

5) Если результат отрицательный, удалить соответствующий вход и на выходе обновить его значение модулем разности

Пример применения операции упрощения транзакции из статьи [17] и показан на *рис. 8*.

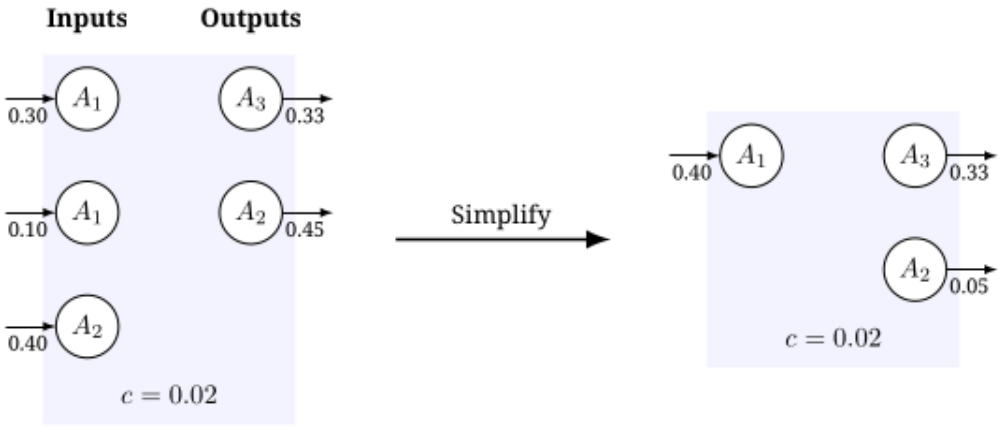


Рис. 8 — Пример упрощения транзакции

2.2. Детектирование Shared Send транзакций

Согласно документу [17] все транзакции в сети биткоин можно поделить на 4 типа:

- Простая транзакция - транзакция, которая не является результатом работы операции смешивания платежей, то есть в ней задействован один пользователь. Все остальные типы транзакций являются смешанными.

- Разделяемая транзакция - вид запутанный транзакции, которую возможно разложить на отдельные под-транзакции, представляющие денежные потоки конкретных пользователей, единственным образом

- Неоднозначная транзакция - вид запутанной транзакции, которую можно разложить на денежные потоки как минимум двумя различными способами

- Трудная транзакция - транзакция, которую не удалось отнести к одному из трех других типов из-за вычислительных ограничений (в том же документе авторами было доказано NP-полнота этой задачи)

Первый тип транзакции распознать просто: если транзакция после проведения вышеописанного преобразования (упрощения транзакции) имеет два и более входов, два и более выходов - это транзакция относится к классу Shared Send. Если же данное условие не соблюдается - транзакция классифицируется как простая и не подлежит дальнейшему анализу.

Так называемые трудные транзакции также легко выявляемы - при использовании метода распутывания транзакций устанавливается таймер, равный некоторому числу секунд, которые даются на распутывание отдельной транзакции. Если за выделенное время алгоритм не справляется, то транзакция считается трудной. Конечно если считать, что в условиях работы нашего алгоритма есть бесконечные вычислительные мощности или бесконечное время, то типов транзакций становится всего три, а трудные транзакции определяются как один из вышеперечисленных типов, однако в данном исследовании будем придерживаться реальных условий.

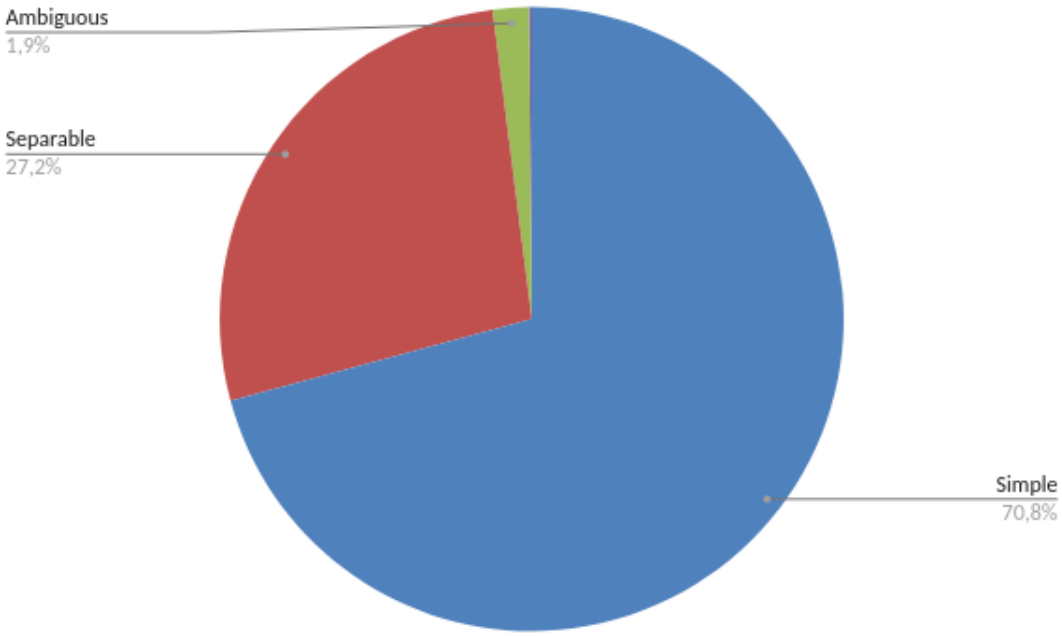


Рис. 9 — Соотношение разных видов транзакций в блокчейне

На *рис. 9* и *рис. 10* можно увидеть соотношение различных типов транзакций в сети биткоин. На графике simple показывает процент простых транзакций, separable - разделимых, ambiguous - неоднозначных, intractable - трудных. Как видно, количество трудных транзакций настолько мало, что на девятом рисунке их даже не видно.

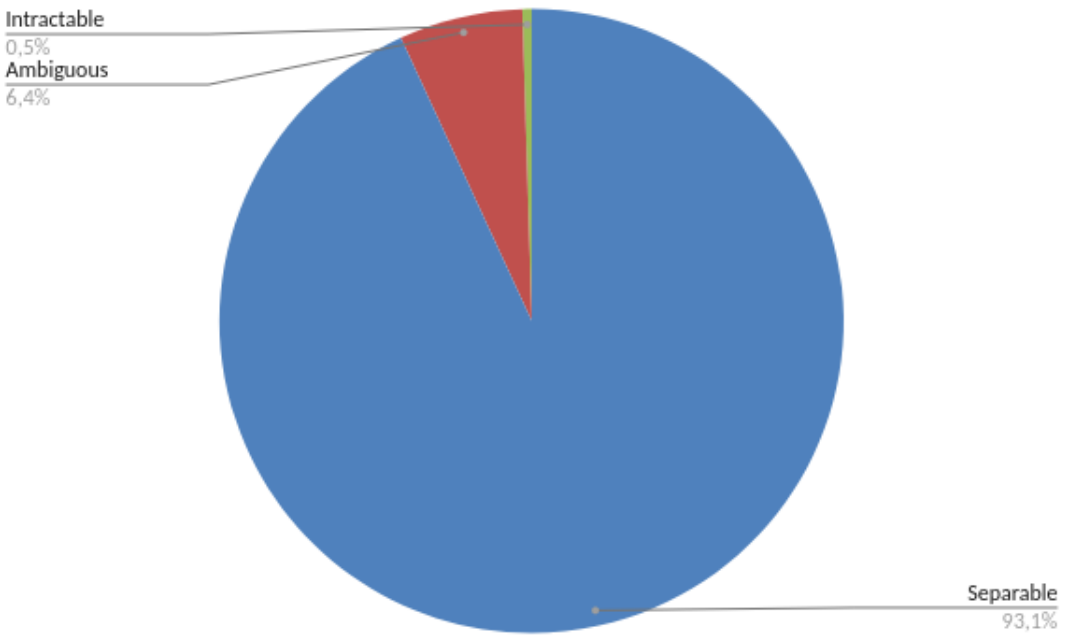


Рис. 10 — Соотношение видов запутанных транзакций в блокчейне

Оставшиеся два типа можно различить формально. Для этого в статье [17] были представлены две леммы. Согласно лемме 1 из статьи, транзакция является неоднозначно разделимой, если соблюдается как минимум одно из нижеперечисленных условий:

1) Существуют два различных подмножества входов $A_1, A_2 \subset A$ и подмножество выходов $B_1 \subset B$ такие, что соединения (A_1, B_1) и (A_2, B_1) возможны.

2) Существуют два различных подмножества выходов $B_1, B_2 \subset B$ и подмножество входов $A_1 \subset A$ такие, что соединения (A_1, B_1) и (A_1, B_2) возможны.

Согласно лемме 2, транзакция является неоднозначно разделимой, если выполняется хотя бы одно из условий:

1) Существует две пары подмножеств $A_1, A_2 \subset A$ и $B_1, B_2 \subset B$ такие, что A_1 и A_2 различные и имеют ненулевое пересечение, а пары (A_1, B_1) и (A_2, B_2) могут быть соединены и при этом минимальны.

2) Существует две пары подмножеств $A_1, A_2 \subset A$ и $B_1, B_2 \subset B$ такие, что B_1 и B_2 различные и имеют ненулевое пересечение, а пары (A_1, B_1) и (A_2, B_2) могут быть соединены и при этом минимальны.

Если же говорить менее формальным языком, то транзакция называется неоднозначно разделимой тогда, когда ее можно распутать как минимум двумя способами.

2.3. Распутывание Shared Send транзакций

Для распутывания транзакций нам необходимо рассмотреть все возможные варианты составления подмножеств входов данной транзакции, таких, что $A' \subset A$. Далее на координатной прямой отобразим сумму каждого подмножества A'. Аналогичным способ поступим со всеми возможными подмножествами выходов $B' \subset B$. На *рис. 11* A'_i и B'_i обозначают сумму элементов множеств A' и B' соответственно.

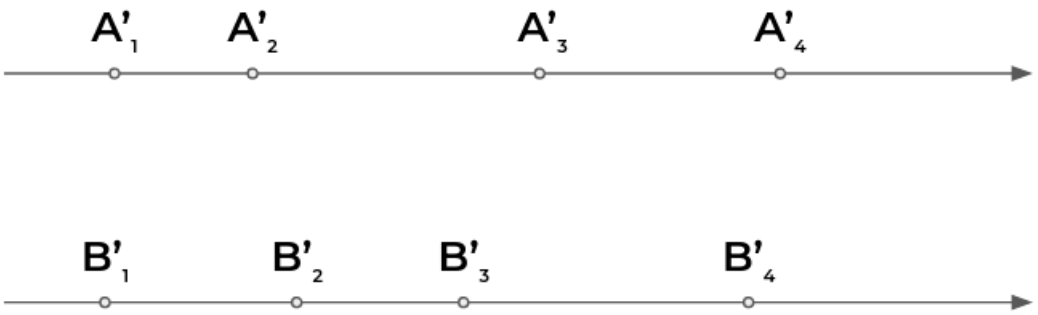


Рис. 11 — Визуальное представление сумм подмножеств

После этого необходимо найти такие пары (A'_i, B'_i), для которых верно следующее условие:
$Sum(B'_i) \leq Sum(A'_i) \leq Sum(B'_i) + c$, где c - размер комиссии за транзакцию

Данное условие обеспечивает один из принципов работы биткоина, что сумма значений входов не должна быть больше чем сумма значений выходов, включая комиссию за транзакцию. На *рис. 12* можно увидеть подмножества, соответствующие условию.

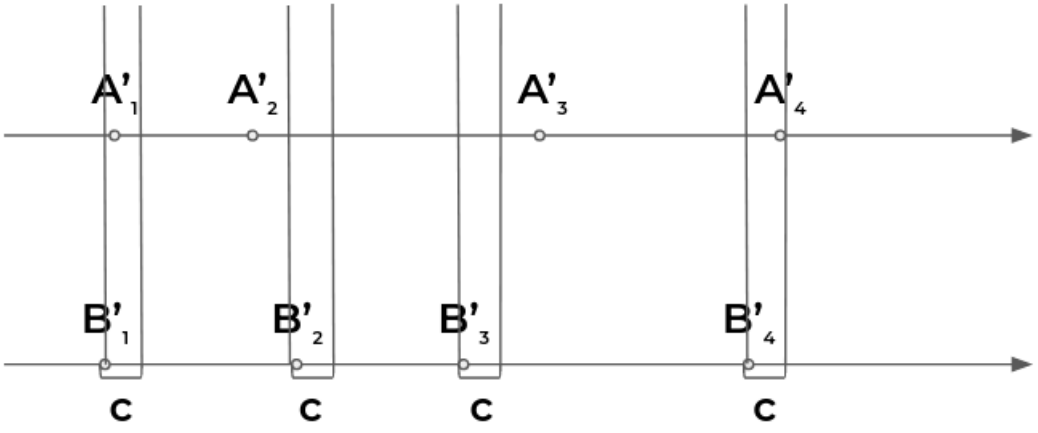


Рис. 12 — Визуальное распутывание транзакции

Если внутри отрезка $[B'_i, B'_i + c]$ попадает точка A'_i, то образуется пара (A'_i, B'_i), являющаяся соединяемой. При этом, воспользовавшись леммами, указанными выше, можно заметить, что если выполняется хотя бы одно из следующих условий:

1) Существуют два разных подмножества A'_i и A'_j такие, что $A'_i \in [B'_k, B'_k + c]$ и $A'_j \in [B'_k, B'_k + c]$

2) Существует два различных отрезка $[B'_i, B'_i + c]$ и $[B'_j, B'_j + c]$ такие, что $A'_k \in [B'_i, B'_i + c]$ и $A'_k \in [B'_j, B'_j + c]$

3) Существуют различные подмножества A'_i и A'_j и различных отрезка $[B'_k, B'_k + c]$ и $[B'_m, B'_m + c]$ такие, что $A'_i \in [B'_k, B'_k + c]$ и $A'_j \in [B'_m, B'_m + c]$, но при этом $A'_i \cap A'_j \neq \emptyset$

То данная транзакция будет считаться неоднозначно разделяемой, так как в каждом из случаев можно представить как минимум два различных варианта корректного разделения. Если же ни одно из вышеуказанных условия не выполняется, транзакция считается разделяемой.

Данное решение легко представить визуально, оно наглядное, но неэффективное, так как для предоставления вариантов распутывания транзакций необходимо перебрать все возможные варианты входов и выходов алгоритма. Количество подмножеств данного множества можно узнать по формуле 2^n, где n - размер исходного множества. Для транзакций сети биткоина вполне стандартная ситуация, когда количество входов транзакции исчисляется десятками или сотнями, а с экспоненциальной формулой пользоваться данным методом в реальных условиях ограниченного времени и вычислительных мощностей будет невозможно.

В репозитории [1] представлен код алгоритма, используемого в данном эксперименте при распутывании транзакций.

Глава 3. Анализ данных

Все рассчеты, эксперименты и анализ данных будут проводиться с использованием языков программирования Python версии 3.10, C++17 и технологии IPython Notebook, которая позволяет проводить и фиксировать эксперименты в формате записной книжки.

3.1. Хранение информации

Для начала работы необходимо загрузить и синхронизировать Full Bitcoin Node [5], содержащую в себе данные о всей истории биткоин блокчейна. Файла блоков хранятся в формате .blk, в двоичном виде. Для их расшифровки понадобится программа [16]. После преобразования файла блока можно посмотреть на формат хранения данных.

```
6210905
Start ./blocks/blk02803.dat in 2022-05-25 15:33:32.346311
Magic number = D9B4BEF9
Block size = 00169CCE
SHA256 hash of the current block hash = 00000000000000000007F5E68562098E8A67867F758298DCE68C100293E624
Version number = 3FFF0004
SHA256 hash of the previous block hash = 00000000000000000008C298864A07C62172183E0E78DC784E01FF9282018DA5
MerkleRoot hash = 156219E1A29D9565029C24D713DD7E7EB1644CB1DAA22E7ACC6B803B640B9462
Time stamp = 618EE3F0
Difficulty = 170CFECF
Random number = 853DDCF9
Transactions count = 2759

TX version number = 00000001
Inputs count = 01
TX from hash = 0000000000000000000000000000000000000000000000000000000000000000
N output = FFFFFFFF
Input script = 0336D30A2CFABE6D6DF20FD8091BE5070F5EF6F4C258C295FC5A54A3466450148AF266736F1A6CE44610000000F09F909F082F4632506F6F6C2
                F0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000050045240000
Sequence number = 00000000
Outputs count = 5
Value = 0000000026620913
Output script = 76A914C825A1ECF2A6830C4401620C3A16F1995057C2AB88AC
Value = 0000000000000000
Output script = 6A24AA21A9EDA91D95EB754C74D3043635A48E817A26E837E2E863AF69B3941142CF90FAC5AF
Value = 0000000000000000
Output script = 6A2448617468D42067E8693075C9C92B8AD48A23CB0323302BCC07D0FC62AE4217A22AF44556
Value = 0000000000000000
Output script = 6A4C2952534B424C4F434B3A0A9EFEE1AA68295C1A6B79EB715E353E81AA911E878FCA54EBDEBF2E003AA5F5
Value = 0000000000000000
Output script = 6A24B9E11B6D5252E0C410C81B949A312CD4346B214ABA03195329E5AAEE3057B29E8636E9EB
Witness 0 0 32 0000000000000000000000000000000000000000000000000000000000000000
Lock time = 3DE518A1
TX hash = 8D3194173E95F1FD77803D388211F5FD5320AC1F0FE65C8C61FEA42A0F2E1A8C
```

Рис. 13 — Обработанный файл блока

На *рис. 13* можно увидеть примерный формат хранения данных в биткоин сети. В нем присутствует много лишний для данного исследования данных, а также фактически отсутствует входные адреса, есть только script. Для подсчета входных адресов транзакций нужно с использованием Python Bitcoin RPC [12] обойти весь блокчейн и извлечь из публичного скрипта необходимый адрес. В листинге **??** представлен необходимый код.

После предобработки файл блока содержит только необходимую информацию о самом блоке и транзакциях в нем. Пример такого файла можно увидеть на *рис. 14*.

```
1439611249
453
>1
Inputs:2
1AvK36bXoak4fGsYu4Jg5Q7nbG7xe42Zf7:11090750000
1AvK36bXoak4fGsYu4Jg5Q7nbG7xe42Zf7:10159460000
Outputs:2
1MZh1i8mrMBakMRd2i997sD95ZSwQLZ6hC:21200000000
1AvK36bXoak4fGsYu4Jg5Q7nbG7xe42Zf7:50060800
>2
Inputs:4
15Kezo54LCLpYhHszkqMqa2TNXXLBtTP4V:32300000
15Kezo54LCLpYhHszkqMqa2TNXXLBtTP4V:105600000
15Kezo54LCLpYhHszkqMqa2TNXXLBtTP4V:129900000
15Kezo54LCLpYhHszkqMqa2TNXXLBtTP4V:142000000
Outputs:2
1QGhb1BT6SLNXeprNqJaNDCPPrj8T581wX:18430024
134PxXJ5oK9kPc7GqvSuxojaoHaors8Xwn:391359976
```

Рис. 14 — Финальный файл блока

3.2. Упрощение

Алгоритм упрощения транзакции был описан в пункте 2.1.2. Входы и выходы могут содержать одинаковые адреса как внутри, так и между друг другом. После упрощения транзакции гарантируется, что внутри одной транзакции каждый адрес встречается не более одного раза как на входе, так и на выходе.

```
Before simplify:
==========Transaction info==========
Simple: False
In addresses: ['1PWiUdt2H4c9aFP9mrZo7rWhcsmNnwAyDk', '1PWiUdt2H4c9aFP9mrZo7rWhcsmNnwAyDk']
In values: [120, 230]
Out addresses: ['bc1qxz4zn9a55yuvcwpze8avepwtendlklrh0arupf', '343Ez3orvtreoQsjkuaXGURBbFdXyQi5HX']
Out addresses: [100, 250]

After simplify:
==========Transaction info==========
Simple: True
In addresses: ['1PWiUdt2H4c9aFP9mrZo7rWhcsmNnwAyDk']
In values: [350]
Out addresses: ['bc1qxz4zn9a55yuvcwpze8avepwtendlklrh0arupf', '343Ez3orvtreoQsjkuaXGURBbFdXyQi5HX']
Out addresses: [100, 250]
```

Рис. 15 — Выявление дублирования входных адресов

```
Before simplify:
==========Transaction info==========
Simple: False
In addresses: ['1PWiUdt2H4c9aFP9mrZo7rWhcsmNnwAyDk', '1PWiUdt2H4c9aFP9mrZo7rWhcsmNnwAyDk']
In values: [120, 230]
Out addresses: ['bc1qxz4zn9a55yuvcwpze8avepwtendlklrh0arupf', '1PWiUdt2H4c9aFP9mrZo7rWhcsmNnwAyDk']
Out addresses: [100, 250]

After simplify:
==========Transaction info==========
Simple: True
In addresses: ['1PWiUdt2H4c9aFP9mrZo7rWhcsmNnwAyDk']
In values: [100]
Out addresses: ['bc1qxz4zn9a55yuvcwpze8avepwtendlklrh0arupf']
Out addresses: [100]
```

Рис. 16 — Выявление одинаковых адресов на входе и на выходе

На *рис. 15* и *рис. 16* можно увидеть результат работы метода упрощения транзакции. Сначала показано изменение транзакции после выявления двух одинаковых адресов в списке входных адресов, а после - когда два одинаковых адреса находятся на входе и на выходе. В обоих случаях, как можно заметить, пересчитываются также и значения отправленных монет. Также обе транзакции после преобразования были помечены простыми.

3.3. Распутывание

На *рис. 17* можно увидеть работу алгоритма на примере распутывания простой транзакции. Этот пример взят из 376229-го блока биткоин, транзакция номер 4.

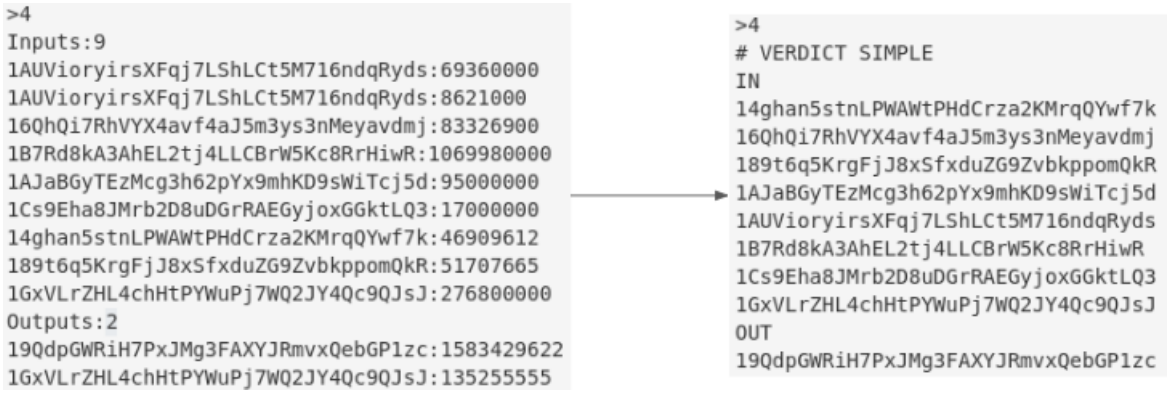


Рис. 17 — Работа алгоритма на примере простой транзакции

Как можно заметить, среди входов и выходов транзакции есть повторяющиеся адреса, а именно адрес, заканчивающийся на (..QJsJ). С помощью описанного алгоритма упрощения транзакции программа преобразует начальную транзакцию следующим образом: так как замечено дублирование адресов со входа и с выхода, необходимо определить, какое значение больше - на входе или на выходе. При вычитании 276800000 и 135255555 получается 141544445, следовательно, после отработки алгоритма транзакция упростится до адреса (..QJsJ) на входе со значением 141544445. Далее видно, что у транзакции мощность множества выходов равняется одному, следовательно, транзакция клас-

сифицируется как простая, что и видно на рисунке, под номером транзакции написано "VERDICT SIMPLE".

Далее рассмотрим пример из 376229-го блока биткоин, транзакция номер 29. В работе представлены наименее трудные для восприятия транзакции, чтобы сместить фокус с разбора большого числа случаев на непосредственные результаты работы алгоритма.

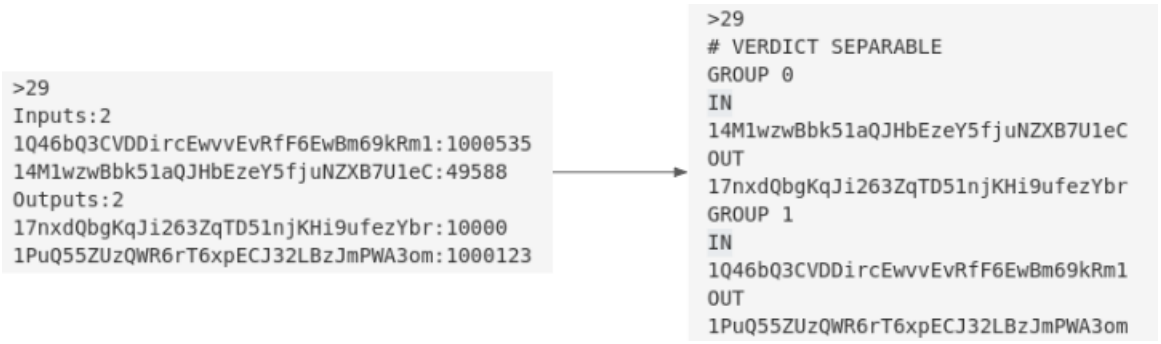


Рис. 18 — Пример разделяемой транзакции

На *рис. 18* видно, что алгоритм распутать транзакцию единственным образом: (..U1eC) → (..zYbr), (..kRm1) → (..A3on). Других вариантов распутывания здесь быть не может, рассмотрим, почему. С первого входного адреса средства могут пойти как на первый, так и на второй выходной адрес, так как оба значения на выходных адресах меньше значения входного. Рассмотрим теперь второй входной адрес. С него средства могут быть направлены только на первый выходной, так как только значение этого адреса меньше значения рассматриваемого нами адреса. Следовательно, если средства со второго входного адреса идут на первый выходной, значит, средства первого входного идут на второй выходной. Комиссия составляет $1000535 + 49588 - 10000 - 1000123 = 40000$ сатоши.

На *рис. 19* представлена неоднозначно распутываемая транзакция под номером 66 из блока 376229.

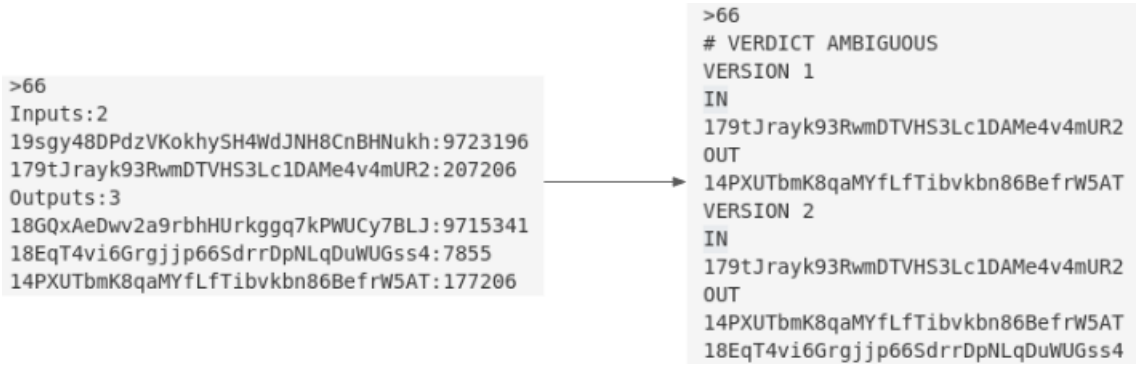


Рис. 19 — Пример неоднозначной транзакции

Как можно заметить, алгоритм предлагает два варианта распутывания. В первом случае адрес (..mUR2) связан с (..W5AT), а во втором - с (..W5AT) и (..Gss4) сразу. Рассмотрим легитимность первого случая: описанная связь адресов корректна, так как значение на выходе меньше значения на входе. Следовательно, неописанные адреса также связаны между собой. При вычитании значений

выходных адресов из значения входного адреса получается 0, а значит такая связь также возможна. Рассмотрим второй случай. Проведя те же операции можно заметить, что такая связь корректна, так как на входе значение больше, чем сумма значений на выходе, и в оставшейся связи значение на входе также меньше значения на выходе. Из-за этого транзакция может быть распутана двумя способами: можно взять разные наборы адресов, связи между которыми будут корректными. В данном случае комиссия составляет 30000 сатоши.

Последней рассмотренной транзакцией будет сложная транзакция номер 37 из блока 376214. Из *рис. 20* видно, что у транзакции большое число входов - 21. При проведении эксперимента стояло ограничение по времени на попытку распутывания транзакции: 10 секунд.

```
>37
Inputs:21
1BcqWbptFFrFwvcfoLrfFMhDCjLbQzJhJT:18480165
1Apb7kfhYoDGtBBFy2pXAkvxZq9UcJ3Uu3:201870
1JEExerr2VR2rFJ3WKiogrt56evAMA1GiN:42777000
19TR3u5EMithBGRpg4r4KWydke4HETVtDU:73907290
1PTpDzdwDmTR7r6tWH1FBtiuQ1GWzKtRAX:41753400
1MThoH5Nj9CWxGAKnaKjpNeesZtozMAT4k:55500000
1Er5fo5uebd8HPJdmWWxGn4nZ945GePnGq:27333026
192A8ZqXx42BnSbFNnXVwgQVRnw2S5749L:386222571
176JTegaPjzeqY1jvH6UHYyf8ht2eMTULx:14061667
14cdzwLWYedKiwCBorP6T16cGtBX6yCwML:15218152
1ByHAvRuFmNYqiiUPDvRSZuJBvb2WaZgQv:532000000
14RWZSY3CK38RBfqpqDEe6zTeSixvXKJQb:31110
142mGbwJVteMtDcbHTdzNWSQ36vod1WjxS:1000000000
16Udp27JJxfhBeY2n8hBEXVQPiyqeQJZa2:21291998
1A5jkUevhoDHDdBtdb7wVmda499FCaJASE:38560411
1Mc5bWXGNY1ThCEmdJMvDqZTQxNVfVG3uW:170524000
1JxZSJrHBQTwvV2wyfYvBZ8BUCGzyP5fBU:1079352
12V4wriLhtcBmrwhZqVMqGvEgA5fBGYdL1:4804605
176JTegaPjzeqY1jvH6UHYyf8ht2eMTULx:25311000
1EYhQDJQ2zRBRjBa3qZRUcsfyvy6Bwi7UZ:394325788
1AQhcKZAzRW2T6i1HLWQskXkNBWATApUyr:40867370
Outputs:2
1DcjQ2U3tKeHBkMnHDngBBxDBYWjtx1w3G:2903220424
15MYc2Qev4GqxWdSNq3FzUz4mC4khxTchB:1000388
```

Рис. 20 — Пример нераспутанной сложной транзакции

Видно, что пересечение множеств входных и выходных адресов пустое, из чего следует вывод, что транзакция не может быть простой. За предоставленное время алгоритм не предоставил решения задачи распутывания, так как нужно было перебрать большое число комбинаций.

3.4. Численный эксперимент

По результатам работы программы обсчета блокчейна [12] [2], алгоритма распутывания транзакций [1], а также программ сбора [3] и агрегации данных [4] в результате исследования есть ниже-представленные графики.

Первая пара - зависимость количество транзакций разных типов от времени (см. *рис. 21*). По оси абсцисс располагается дата в формате кварталов разных лет. На оси ординат - количество транзакций. Типы транзакций здесь и далее представлены в следующем виде: simple - транзакции простого типа, separable - разделяемая, ambiguous - неоднозначная, intractable - трудная. На первом графике

представлены все типы транзакций, на втором - только типы запутанных транзакций для наглядности, не включая простой тип. Можно увидеть, что в целом, видна тенденция рост количества транзакций, как общего числа, так и пропорционально по типам.

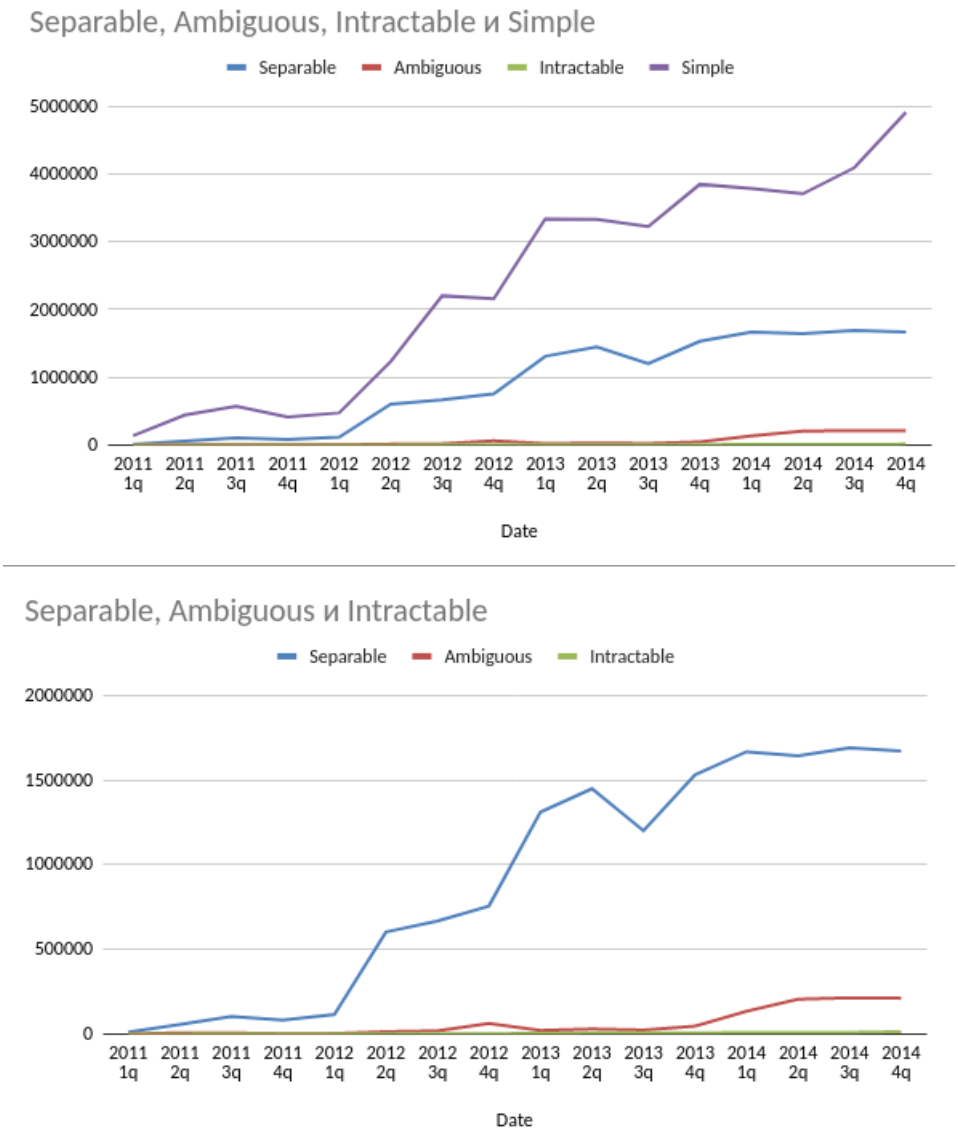


Рис. 21 — Графики количества типов транзакций по кварталам

Следующая пара графиков (см. *рис. 22*) основывается на тех же данных, что и предыдущая, однако теперь вертикальная ось - это логарифм количества транзакций. Графики аналогично представлены по типам, с простыми транзакциями и без них. Можно заметить, что порядок числа распутываемых и простых транзакций схож и растет приблизительно одинаково. В то же время заметно, что те же показатели у неоднозначных и трудных транзакций заметно ниже. У трудных транзакций динамика роста примерно похожа на простые и распутываемых, однако у неоднозначных транзакций рост идет рывками.

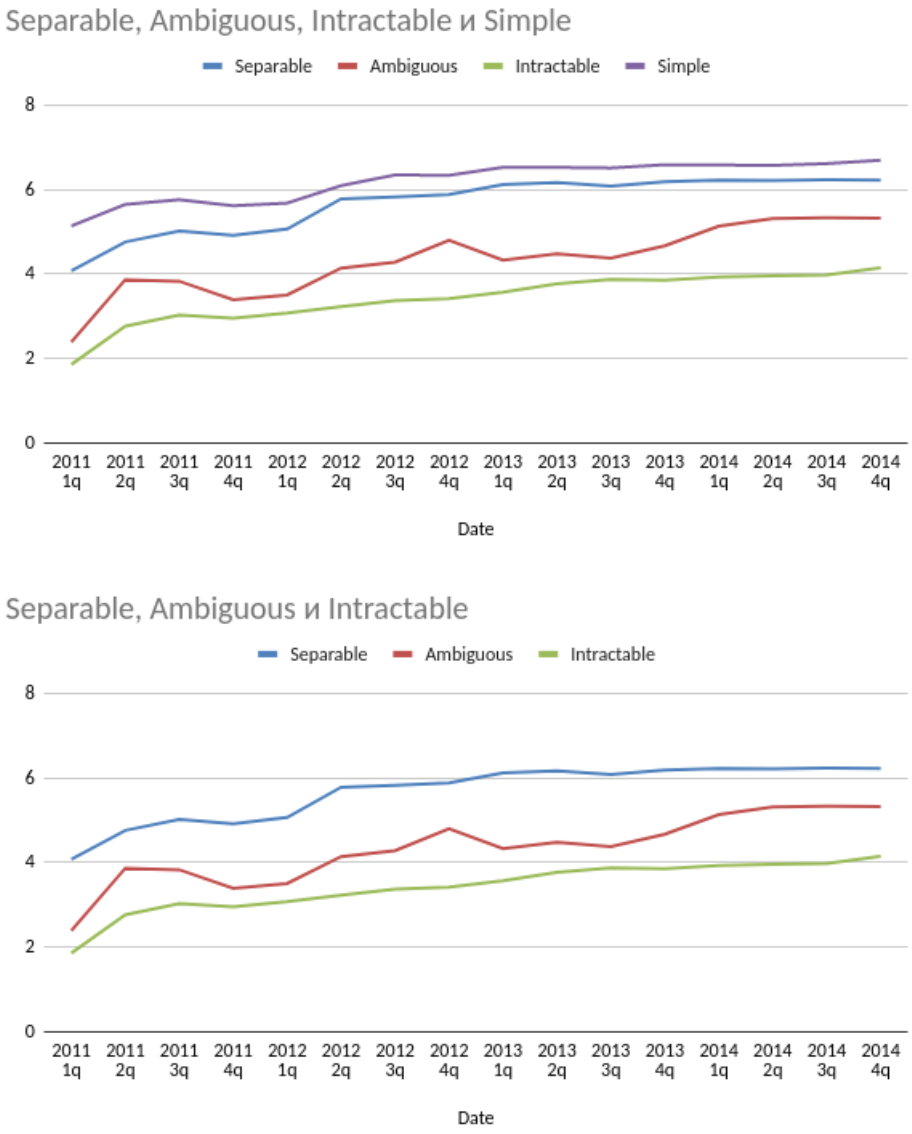


Рис. 22 — Прологарифмированный график количества типов транзакций по кварталам

На *рис. 23* можно увидеть динамику роста числа транзакций биткоина в целом, как напрямую, так и прологарифмированную. Заметно, что рост идет рывками. Скорее всего, это связано с в то время только набирающей обороты популярностью биткоина, когда каждое упоминание в средствах массовой информации вызывал интерес к технологии.

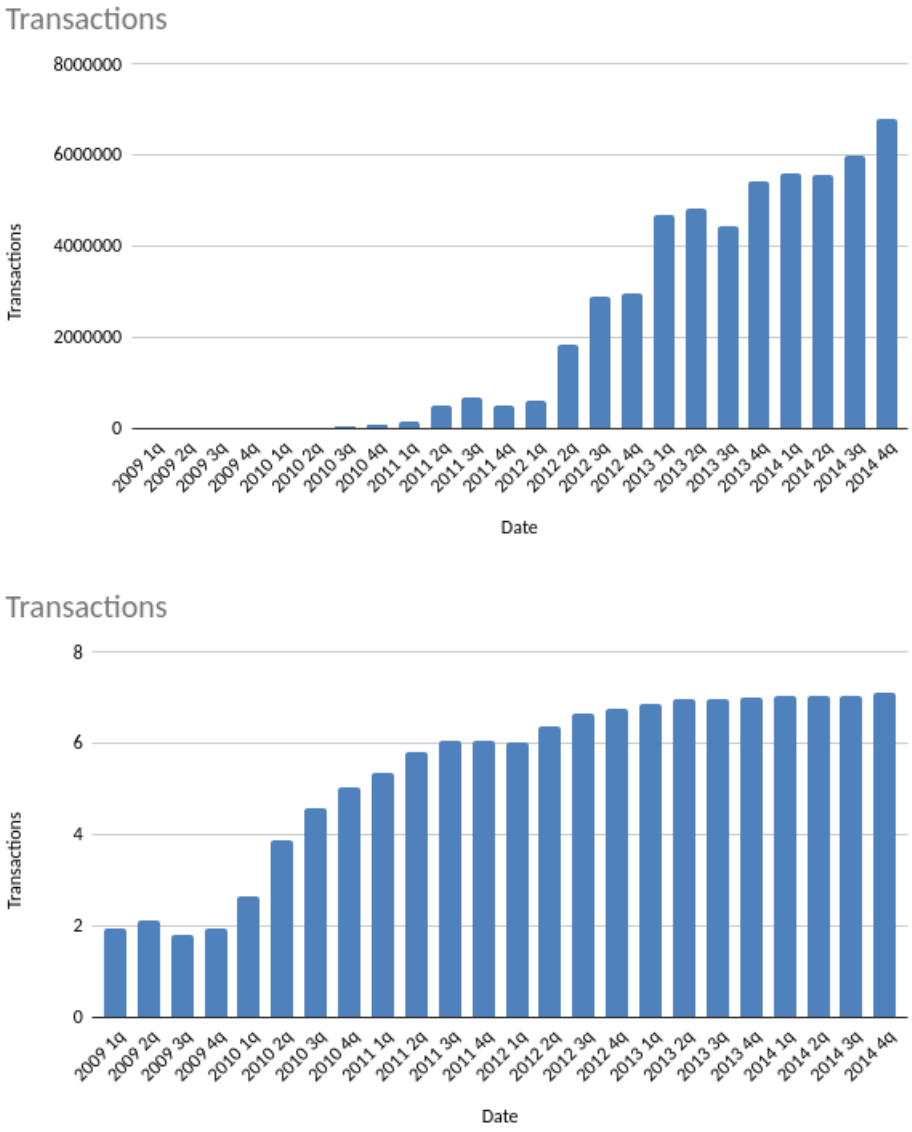


Рис. 23 — График общего числа транзакций по кварталам

Также в разделе 2.2 представлены графики соотношения различных типов транзакций между собой. На данный момент обработано более 53 миллионов транзакций, собрана детальная статистика по каждой из них и составлены соответствующие графики.

Заключение

В ходе данного исследования разработан, реализован и опубликован псевдополиномиальный алгоритм распутывания методом динамического программирования. Его код представлен в репозитории [1]. Создана локальная инфраструктура для распутывания транзакций сети биткоин блокчейна, запущено распутывание с использованием Full Bitcoin Node [5], пред- и постобработки на Python [2] [3] и распутывания на C++ [1]. Обработаны все транзакции в период с 2009 года (начало работы биткоина) до 2014 года, по результатам вычислительных экспериментов можно сделать вывод, что в сети биткоина около 30% запутанных транзакций, 93% которых можно распутать однозначным способом, используя представленный алгоритм.

В главе 1 рассмотрены различные методы анонимизации в сети биткоин блокчейна, их история, недостатки и преимущества. Представлены аналоги, рассмотрены их возможности. Во второй главе описана теория обработки транзакций, их предобработка, классификация и само распутывание. В главе 3 продемонстрирована работа всех описанных алгоритмов, программ и скриптов, а также приведена статистика и графики по более чем 53 миллионам транзакций биткоин сети.

Список использованных источников

1. adkurylev [электронный ресурс] : Untangling algorithm. — URL: https://github.com/adkurylev/graduation_paper_code/blob/master/transaction_solver.cpp (дата обр. 24.05.2022).
2. adkurylev [электронный ресурс] : Bitcoin blocks preparation code. — URL: https://github.com/adkurylev/graduation_paper_code/blob/master/Bitcoin_data_preparation.ipynb (дата обр. 24.05.2022).
3. adkurylev [электронный ресурс] : Untangling script. — URL: https://github.com/adkurylev/graduation_paper_code/blob/master/read.py (дата обр. 24.05.2022).
4. adkurylev [электронный ресурс] : Untangled transactions information aggregation. — URL: https://github.com/adkurylev/graduation_paper_code/blob/master/aggregate.py (дата обр. 24.05.2022).
5. Bitcoin Project [электронный ресурс] : Full Bitcoin Node. — URL: https://bitcoin.org/en/full-node (дата обр. 24.05.2022).
6. Blockchain.com [электронный ресурс] : Explorer. — URL: https://www.blockchain.com/explorer (дата обр. 24.05.2022).
7. *Bonneau J.* Mixcoin: Anonymity for bitcoin with accountable mixes // International Conference on Financial Cryptography and Data Security. — Springer, Berlin, Heidelberg, 2014. — с. 486—504.
8. Chainalysis [электронный ресурс] : The Blockchain Data Platform. — URL: https://www.chainalysis.com (дата обр. 24.05.2022).
9. Crystal Blockchain [электронный ресурс] : Analytics for Crypto Compliance. — URL: https://crystalblockchain.com/ (дата обр. 24.05.2022).
10. European Union Agency for Law Enforcement Cooperation. 2020. [электронный ресурс] : Internet organised crime threat assessment (IOCTA) 2020. — URL: https://www.europol.europa.eu/sites/default/files/documents/internet_organised_crime_threat_assessment_iocta_2020.pdf (дата обр. 27.02.2022).
11. European Union Agency for Law Enforcement Cooperation. 2021. [электронный ресурс] : Internet organised crime threat assessment (IOCTA) 2021. — URL: https://www.europol.europa.eu/cms/sites/default/files/documents/internet_organised_crime_threat_assessment_iocta_2021.pdf (дата обр. 14.04.2022).
12. jgarzik [электронный ресурс] : Python Bitcoin RPC. — URL: https://github.com/jgarzik/python-bitcoinrpc (дата обр. 24.05.2022).
13. Maxwell, G., 2013. [электронный ресурс] : CoinJoin: Bitcoin privacy for the real world. — URL: https://bitcointalk.org/index.php?topic=279249.0 (дата обр. 27.02.2022).
14. Maxwell, G., 2013. [электронный ресурс] : CoinSwap: Transaction graph disjoint trustless trading. — URL: https://bitcointalk.org/index.php?topic=321228.0 (дата обр. 20.04.2022).
15. *Nakamoto S.* Bitcoin: A peer-to-peer electronic cash system // Decentralized Business Review. — 2008. — с. 21260.
16. ragestack [электронный ресурс] : Bitcoin Blockchain Parser. — URL: https://github.com/ragestack/blockchain-parser (дата обр. 24.05.2022).
17. *Yanovich Y., Mischenko P., Ostrovskiy A.* Shared send untangling in bitcoin // bitfury.com. — 2016. — с. 1—25.