Лекция 1. Место и роль блокчейнов в эко-системе цифровой экономики

Блокчейн - это гораздо больше чем криптовалюта

Цель: Определить нишу для технологии распределенных реестров в процессах современного этапа развития цифровой экономики.

Драйверами роста цифровой экономики являются современные информационные технологии, часто называемые инновационными или прорывными. Эффекты, привносимые ими, иногда бывают просто ошеломляющими. На наших глазах отдельные отрасли человеческой деятельности изменяются до неузнаваемости. Например, использование последних достижений из области Big Data, Data Mining обусловило появление новой прикладной науки – бионформатики, открывшей совершенно новые горизонты в медицине. С помощью инструментов анализа больших данных началась расшифровка генома человека, и не только его. Разработаны эффективные средства для борьбы с генетическими, наследственными заболеваниями, существенно ускорился процесс разработки новых лекарств и т.д.

Не меньшие надежды возлагаются на технологию построения децентрализованного, распределенного реестра – блокчейн. Не прошло и одиннадцати лет с момента запуска Биткоин-сети и публикации первых статей, подписанных Сатоши Накомото, а идеи децентрализации приложений и данных прочно овладели лучшими умами. Многие аналитики сулят блокчейну самое радужное будущее. В IT-сообществе рассматриваются различные области применения распределенных реестров, анализируются уже существующие платформы, оценивается технологический и экономический эффекты от их внедрения. Попытки объявить эту технологию панацеей от всех цифровых бед Россией уже начинают просто пугать. Безусловно за блокчейном большое будущее. Совершенно обосновано системы распределенного реестра в федеральной программе "Цифровая экономика Российской Федерации" включены в число основных сквозных цифровых технологий, призванных повысить конкурентоспособность России на глобальном рынке. Широта и возможности применения распределенных реестров видятся безграничными, они способны трансформировать целые сектора коммерческой или государственной деятельности. Эта технология позволяет в принципе устранить в сложившихся цепочках хозяйственных связей посредников. Перспективы ритейла в этом плане видятся весьма ограниченными. Недолгим окажется век нотариата, загсов, кадастров, множества других контор и палат, суть деятельности которых сводится к регистрации, хранению и аудиту событий, фактов, сделок, вообще коммуникаций (операций) в системах, включающих граждан, государства, предприятия, и др.

Весьма символичной представляется дата появления сети Биткоин. Она практически точно совпала с началом Четвертой промышленной революции. Предложенная Сатоши Накомото технология целиком и полностью отвечает основным трендам эпохи формирования Шестого технологического уклада: рождена на стыке экономики, психологии и нескольких научно-технических направлений; является сетевой по своей природе и обеспечивает массовые коммуникации пользователей, исключая любых посредников в общественных и бизнес-процессах;

позволяет не просто автоматизировать, а трансформировать отдельные сферы деятельности или бизнес-функции.

Полная трансформация ожидает и многие вполне успешные, и даже процветающие в настоящее время бизнесы. Например, всем известный Uber – прекрасная сетевая компания с колоссальной капитализацией. И ведет свои дела эта компания правильно, в полном соответствии с духом и законами сетевой экономики (но правда с законами, действующими на начальном этапе ее развития). Компании удалось инкапсулировать опыт клиента, современные технологии (сотовые телефоны с возможностями геолокации) и традиционный товар – доставка из одной точки в другую - в эффективную платформу, фактически предлагающую новый, информационный продукт. Они сумели сформировать электронную цепочку добавленной стоимости, и хорошо на этом заработали. Но недавно в Израиле появился проект "совместного использования автомобилей", который сразу назвали "убийцей Убера" или "анти-Убером". Сегодня ареал распространения инициативы La'Zooz уже включает Северную Америку, Западную Европу, Ближний Восток. Это первое децентрализованное приложение, основанное на применение блокчейна в сфере услуг такси. Клиент при его помощи даже сможет заработать, если будет двигаться с определенной скоростью, или помогать развитию проекта. В данном случае мы имеем дело с полной трансформацией модели взаимодействия участников в сфере частного извоза. И эта трансформация обеспечивается технологией распределенного реестра. Еще не затихла борьба профсоюзов таксистов с компаниями типа Uber и Lyft, как уже эти новые технологичные предприятия начинают вытесняться с рынка стартапами, создающими бизнес на основе распределенного реестра.

Однако, как и любая другая технология в настоящее время блокчейн имеет свои ограничения, заслуживающие самого серьезного внимания. Как минимум, теория и практика распределенного реестра требуют своего дальнейшего развития. Только в этом случае применение данной технологии в определенных сферах будет экономически выгодным и технологически оправданным.

Прежде всего, хотелось бы несколько уточнить терминологию. Часто в определении понятия блокчейн проскальзывает метафора "распределенная база данных". Это вряд ли является корректным. Распределенные базы данных появились задолго до старта сети биткоинов и подразумевает под собой рассредоточенную по узлам сети базу данных, поддерживающую свою функциональность за счет фрагментации и репликации. Здесь самым существенным аспектом является наличие единого, центрального сервера, копирующего свои данные на вспомогательные серверы (реплики). Конечно, такой подход далек от идеи децентрализации, реализованной в технологии блокчейн.

В заключение вновь вернемся к перспективам технологии распределенного реестра и тем препятствиям, которые стоят на пути его развития. Как ни странно, их достаточно много. Это и правовая неурегулированность, недопонимание места и роли этой технологии топ-менеджментом, принимающим решение об автоматизации, технические особенности (малая пропускная

способность сети, постоянный рост самого реестра) и др. Но есть еще одно. В свое время именно Биткоин стал начальной вехой в развитии блокчейн-технологий. Многие обозреватели совершенно искреннее считают его одним из самых гениальных изобретений последних десятилетий. Однако сейчас ажиотажный интерес, непомерное восхищение техническими решениями, применяемыми в сети Биткоин, оказывают медвежью услугу технологии блокчейн в целом, область применения которой гораздо шире. Попытки прямо применить техническое решение, изначально предназначенное для создания криптовалюты – очень специфичного по функционалу проекта – к решению других задач (например, в области госуслуг, медицине) ожидаемо приводят к проблемам и разочарованию. Теория блокчейна требует своего дальнейшего развития. Пока в технологиях построения распределенного реестра будут превалировать схемы, реализованные в протоколе Биткоин, мы не сможем учитывать реалии бизнес-логики, присущие другим предметным областям и практическим задачам. В связи с этим, прежде чем перейти к обсуждению протокола Биткоин, немного поговорим о технологии блокчейн в целом.

Технология блокчейн и децентрализованные приложения

Цель: Сформировать представления о соотношении между централизованными, децентрализованными и распределенными системами, блокчейном и Биткоином.

По сути реализация блокчейна, предложенная Сатоши Накомото, наполнила новом смыслом теорию и практику применения распределенных систем. Децентрализованные в том или ином смысле информационные системы конечно существовали и до сети Биткоин, однако, во-первых, область их применения была весьма ограничена и, во-вторых, с точки зрения архитектуры практически все известные разработки по сути оставались централизованными, децентрализация в них применялась на уровне отдельных технических решений. При этом доминирующим способом организации высокоуровнего сетевого взаимодействия являлась архитектура клиент-сервер. Безусловно, столь яркое появление криптовалюты было в достаточной степени неожиданным и. безусловно, послужило мощным толчком к развитию децентрализованных приложений. Именно сеть Биткоин позволила достаточно четко определить новые подходы к обеспечению целостности информации, передаваемой между абонентами, не имеющими оснований доверять друг другу. Причем цели удалось достичь без привлечения третьей, надежной стороны (центрального звена, банка, управляющего органа, выделенного сервера и т.д.) на основе распределенных вычислений, криптографических преобразований и пиринговых сетей. Пришедшее вместе с появлением протокола Биткоин осознание возможностей соблюдения прав на владение ценностей математическими методами и использования модели с ограниченными ресурсами, умноженных на вновь открывшийся потенциал одноранговых сетей, мотивировало разработчиков на создание нового класса программного обеспечения - децентрализованных приложений.

Таким образом, протокол Биткоин – частный случай категории систем, основанных на технологии блокчейн, являющихся, в свою очередь, подклассом децентрализованных приложений.

Рис. 1.1. Соотношение децентрализованных распределенных систем

В чем именно состоит сверх-задача децентрализованных приложений? Они существенно расширяют возможности многоуровневой архитектуры взаимодействий, реализованной стеком протоколов сети Интернет. Например, платформа Биткоин по сути является надстройкой (новым уровнем) над стандартной телекоммуникационной, глобальной сетью, добавляя к существующей возможности обмена данными новое качество – проведение прямых, безопасных платежей.

Появление первой криптовалюты нельзя рассматривать как очередной этап автоматизации в сфере денежного обращения. Биткоин явился следствием глубокой трансформации известных платежных систем. Децентрализованный консенсус и технология блокчейн, лежащие в основе нового протокола, позволили впервые без обращения к центральному регулятору решить задачу двойных трат, с которой несколько десятилетий не могли справиться разработчики электронных денег.

Программа майнинга биткоинов определяет их ограниченную эмиссию – всего может быть добыто лишь 21 000 000 токенов. Последняя монета сформируется в 2140 году.

Двойная трата или двойное расходование (от англ. Double spending) — повторное использование в платежной системе одних и тех же цифровых монет. В сети Биткоин теоретически может быть совершена путем создания второй транзакции, использующей в качестве источника (входа) ранее использованные средства (растраченные выходы).

Попробуем сформулировать определения для основных игроков на поле децентрализации информационных систем, используя индуктивные умозаключения. Начнем с платформы Биткоин.

Биткоин (от англ. Bit Coin – цифровая монета) – криптовалюта, которая в отличие от фиатных денег не создается и не контролируется на государственном уровне, а также является дефляционной по своей природе, благодаря строго ограниченной эмиссии. По сути сеть Биткоин представляет одноранговую платежную систему (отсутствуют какие-либо управляющие или процессинговые центры). Для учета операций используется одноименная единица (BTC, не путать с BCH, применяемой в форке Биткоина - Bitcoin Cash). Минимальная составная часть BTC – сатоши

(Satoshi, 1 сатоши = 10-8 BTC). Все сделки в сети Биткоин абсолютно прозрачны и необратимы, т.е., подтвержденная и записанная в реестр операция (транзакция) не может быть отменена. При этом, при работе непосредственно в сети Биткоин обеспечивается анонимность пользователей. Необходимым и достаточным условием для работы с платежной системой является наличие установленного клиента. Протокол платформы Биткоин и программный код базового клиента (Bitcoin Core) полностью открыты. Технологически доверие между участниками системы поддерживается с помощью децентрализованного консенсуса. Консенсус достигается на конкурентной основе, для его реализации используется механизм доказательства выполненной работы (Proof of Work).

Блокчейн (от англ. Block Chain – цепочка блоков) – структура данных, в которой информация о совершенных взаимодействиях (транзакциях) унифицирована и хранится в виде цепочки (связанной последовательности) блоков. Представляет собой децентрализованное хранилище транзакций, не требующее для совершения операций участия каких-либо посредников. Использование технологии блокчейн впервые позволило обеспечить децентрализованный консенсус. Объем цепочки блоков перманентно увеличивается по мере добавления новых блоков, содержащих самые последние записи. Блоки добавляются в блокчейн в линейном последовательно хронологическом порядке. После добавления блоки становятся навсегда неизменяемыми. Это касается также их содержания. Управление блокчейном, построенным на принципе децентрализации, в целом и в частности осуществляется сетью.

Децентрализованные приложения (от англ. Distributed Application, DApps) – приложения, построенные согласно парадигме, являющийся архитектурным антиподом централизованной (клиент-серверной) модели. В децентрализованной системе отсутствуют узлы, управляющие функционалом иных узлов.

Характерные признаки успешной децентрализованной модели:

Максимальна децентрализация. Прежде всего идеальное решение должно обеспечить децентрализацию данных (данные хранятся децентрализовано, максимально надежно, без какого-либо участия арбитра или центрального узла). Каждый пользователь обладает полностью функциональной копией всей совокупности данных. Кроме того, осуществляется децентрализация ценности, идентичности, вычислений.

Синхронизация данных всех пользователей осуществляется в автоматическом режиме на основе механизма достижения распределенного консенсуса, а не за счет "контроля сверху" (управляющего узла).

Открытость кода. Все технологические аспекты доступны любому желающему. Абсолютная прозрачность и несомненная безопасность данных лучше всех остальных доводов вызывают доверие к приложению.

Монетизация приложения осуществляется через внутреннюю криптовалюту. Схема получения прибыли разработчиком может походить на механизм, опробованной в сети Биткоин. Рост ценности платформы дал возможность хорошо заработать всем, кто стоял у ее истоков и немало способствовал ее развитию. Поэтому разработчику достаточно предусмотреть в системе дефицитные, полезные для пользователей ресурсы. Оплата доступа к ресурсам осуществляется посредством специальных токенов (коинов). Пользователи будут конкурировать за токены, чтобы получить возможность работать с системой. Владельцы дефицитных ресурсов получают плату в токенах. Перманентный рост сети при ограниченном количестве токенов, вызывает неизбежный рост ценности внутренних коинов. Токенами может поощряться любая полезная работа в рамках приложения, например, майнинг.

Отсутствие единой точки отказа. Работоспособность приложения (доступность данных) не должна нарушаться при выходе из строя или блокировке какого-то оборудования или инфраструктуры.

Эффективность работы приложения не должна зависеть от количества устройств, на которых оно активировано. Такую же природу имеет требование, согласно которому ни одно лицо или организация не может владеть "контрольным пакетом" его токенов.

Не все децентрализованные приложения удовлетворяют всей совокупности представленных признаков. Например, криптовалюта Ripple последнему критерию не соответствует.

Распределенные системы (от англ. Distributed Application) – системы, построенные с использованием технологии обмена данными между участниками, в которой вычисления распределяются между несколькими узлами.

Распределенные системы могут быть как централизованными, так и децентрализованными. Причем, приложения, использующие блокчейн и пиринговые технологии, могут быть распределенными и децентрализованными одновременно.

Приватные и публичные блокчейны

Цель: Сформулировать определения, сходства и различия, достоинства и недостатки приватных, регулируемых и публичных блокчейнов.

Системы хранения данных, построенные на блокчейне, могут отличаться схемами организации доступа к информации. Т.е., можно построить децентрализованное приложение, в котором обычные пользователи могут только знакомиться с контентом (или вообще не иметь к нему доступа), а его формирование и администрирование являются уделом исключительно привилегированных участников. В таких случаях говорят о приватных (закрытых) блокчейнах.

Итак, в приватных блокчейнах в противовес публичным (открытым) генерация новых блоков осуществляется централизованным образом. Такой подход очень удобен для автоматизации большого числа внутрикорпоративных процессов, и не только. Большая часть узлов в такой сети способно только считывать данные транзакций блокчейна. Все вопросы аудита, управления

данными, децентрализованными приложениями находятся в ведении четко очерченного круга доверенных узлов.

Такая схема достижения консенсуса обладает рядом преимуществ:

Может быть достигнута весьма низкая стоимость транзакций (валидация транзакций реализуется доверенными, как правило, высокопроизводительными узлами).

Можно добиться достаточно высокой пропускной способности системы (показатель TPS - transactions per second – число обрабатываемых транзакций за секунду).

Следует признать, что ряд бизнес-процессов требует повышенного контроля со стороны управляющего центра. Применение приватного блокчейна обеспечивает более контролируемую и прогнозируемую среду по сравнению с системами на основе публичного блокчейна. Как следствие, закрытые децентрализованные приложения обычно являются более гибкими, позволяющими легче менять их функциональность.

Можно использовать более "дешевые" (и быстрые) способы достижения консенсуса по сравнению методом доказательства работы. Самый простой пример такой схемы кратко можно описать следующим образом: в системе выделено определенное (достаточное) число доверенных узлов-майнеров, выполняющих функции контроля и введения новых данных в блокчейн. Для осуществления контроля и аутентификации майнеров используется механизм электронной подписи. Конкуренция среди майнеров отсутствует, они могут генерировать блоки по очереди.

Доверенные ноды, отвечающие за соблюдение консенсуса на уровне блокчейна, не станут заниматься зловредной деятельностью, например, не будут организовывать атаки 51%.

Можно обеспечить высокую скорость подтверждения транзакций.

В принципе, при необходимости в приложениях на основе приватного блокчейна можно достаточно легко осуществлять откаты системы в некоторое предыдущее целостное состояние. Подобные изменения в крайних случаях реализуются и в публичных блокчейнах. Достаточно вспомнить хард форк платформы Эфириум 20 июля 2016 после обнаружения системной ошибки в протоколе, обеспечивающем автономное регулирование инвестиционного капитала. Злоумышленники смогли похитить около 50 миллионов долларов, но не смогли их вывести из системы. Именно благодаря хард форку похищенные средства удалось вернуть (а от единой платформы отпочковался Ethereum Classic (ETC)). Однако, для публичных систем подобные изменения блокчейна воспринимаются как нечто из ряда вон выходящее и, по определению, не должны допускаться.

Приложения, сочетающие достоинства технологий криптовалют (блокчейн 1) и умных контрактов (блокчейн 2), позволяют заменить большое число централизованных сервисов, доминирующих в настоящее время в сфере кадастров (реестров) или финансовых (учетных) систем.

Платой за исключительные достоинства приватных блокчейнов является существенный отход от идеалов децентрализации, что в свою очередь приводит к снижению уровня безопасности и надежности системы.

Если информация блокчейна полностью недоступна для пользователей, интегральная безопасность системы уменьшается. Даже если приложение обеспечивает безопасность реестра данных, коммуникации с пользователями оказываются уязвимыми, например, для MitM-атак. Для публичных блокчейнов эта проблема решается за счет использования универсального механизма подтверждения транзакций, в реализации которого теоретически могут участвовать все пользователи. Пренебрежение этим важнейшим инструментом и использование централизованного подхода в закрытых системах может отрицательно сказаться на их безопасности. Поскольку транзакции в этом случае доступны ограниченному числу узлов, повышается риск вмешательства человеческого фактора в функционирование системы, причем пользователи не смогут оперативно обнаружить такое вмешательство. По сути, закрытая архитектура блокчейнов нарушает два ключевых аспекта технологии децентрализованных приложений:

- децентрализация, как гарант отсутствия единой точки отказа;
- доверие в следствии использования математически обоснованных и автоматически поддерживающихся правил обработки транзакций без какого-либо участия человека.

Вышесказанное касается в том числе и систем, в которых клиенту, предоставляется доступ только к транзакциям, касающимся исключительно самого клиента.

Вместе с этим, решения, обеспечивающие неизбирательный доступ на чтение данных блокчейна вкупе с открытым протоколом ликвидируют большинство угроз, вызванных особенностями архитектурой закрытых блокчейнов.

Атака посредника, или атака "человек посередине" (от англ. Man in the middle (MITM)) — разновидность атаки, в которой злоумышленник перехватывает и при необходимости вносит изменения в сообщения, передаваемые друг другу двумя доверенными сторонами. В случае успеха позволяет обойти защиту канала связи в виде механизма взаимной аутентификации. Для этого злоумышленнику достаточно либо уметь имитировать сообщения каждой из сторон информационного обмена, либо маскировать свое присутствие в качестве посредника.

Примерами крипто-проектов, основанных на концепции приватных блокчейнов, являются hyperledger, Mijin или ripple.

Характерной особенностью публичных блокчейнов является полный контроль над системой со стороны всех ее участников. В частности, даже разработчики системы не могут по собственной инициативе изменить ее функционал, код или данные.

Данные публичного блокчейна доступны повсеместно, невзирая на статусы и полномочия. Любой участник сети может провести некоторую операцию, фиксация которой осуществляется в виде транзакции. Транзакции подтверждаются не каким-то определённым узлом. Нельзя даже

спрогнозировать, кто окажется в роли валидатора. Смысл публичной системы в том, что ни один ее участник не обладает преимуществом или особыми полномочиями для подтверждения транзакций. Мы имеем дело с истинно демократичной системой. Каждый имеет возможность создавать смарт-контракты, переводить средства, отправлять сообщения или вносить новые данные. В таких системах участники обладают некоторой степенью анонимности. Открытые блокчейны призваны защищать особо важную, неизменную информацию.

Очевидные преимущества открытых блокчейнов, включая прозрачность данных и процессов, а также открытость базовых технологий и протоколов (что обеспечивает высокий уровень доверия к такого рода системам), в обозримом будущем должны привести к масштабной трансформации традиционных финансовых институтов и всей финансовой системы в целом.

Кроме публичных и приватных блокчейнов можно выделить некий промежуточный класс технологий, представителей которого мы назовем регулируемыми блокченами.

Сравнение основных аспектов различных типов блокчейнов приведены в таблице 1.1

Таблица 1.1. Сравнение публичных, регулируемых и приватных блокчейнов

	Приватные блокчейны	Регулируемые блокчейны	Публичные блокчейны
Прозрачность системы	Закрытая система. Число пользователей ограничено рамками организации (корпорации). Исходные коды закрыты.	Открытые данные.	Полностью открытая система, включая исходные коды, протоколы и т.д. Любое лицо может присоединиться к подобной системе.
Анонимность пользователей	Псевдо-анонимная система. Анонимность обеспечивается в рамках системы.	Полная идентификация пользователей.	Полная идентификация пользователей.
Механизм консенсуса	В системе присутствуют однозначно определенные доверенные узлы, управляющие сетью.	Смешанный.	На основе криптографической верификации, таких как proof of work, proof of stake и т.д.

Кто контролирует систему	Организация (корпорация).	Организация (корпорация). В обеспечении консенсуса (верификация данных, аутентификация доверенных узлов) могут участвовать обычные пользователи.	Все участники сети, включая разработчиков, пользователей, поставщиков услуг, майнеров.
Бизнес-схема	Управление активами.	Универсальная. Прямой доступ к чтению (созданию) транзакций для пользователей ограничен с использованием дружественных интерфейсов и приложений.	Транзакционная.
Состав майнеров	Перечень майнеров четко определен вне системы.	Перечень майнеров четко определен вне системы.	Любой участник может стать майнером.
Тип майнинга	Комбинированный майнинг.	Ротационный майнинг.	Конкурентный майнинг.
Наличие регулятора	Да	Да	Нет
Устойчивость к цензуре	Валидная транзакция может быть отринута регулятором, или даже исключена из блокчейна после внесения.	Валидная транзакция может быть отринута регулятором.	Удовлетворяющая всем правилам системы транзакция будет в конце концов обязательно добавлена в блокчейн.

Отметим еще один немаловажный факт - отношение традиционных финансовых учреждений к публичным блокчейнам, в целом, остается гораздо более прохладным чем к закрытым или регулируемым. Основные претензии к открытым системам со стороны финансовых регуляторов сводятся к отсутствию контроля за процессами обработки транзакций (в первую очередь это касается анонимности майнеров, что зачастую прямо противоречит действующему

законодательству), угрозам конфиденциальности клиентов в публичной среде, невысокой гибкости и сложности модернизации.

Сравнение процедуры традиционной банковской онлайн транзакции и транзакции в сети Биткоин

Цель: Определить перспективы использования криптовалют в качестве платежного средства.

Какие перспективы в финансовой сфере имеют криптовалюты? Прочитайте два следующих абзаца и решите сами.

Традиционная схема банковской онлайн транзакции выглядит примерно так. Посредством POS-терминала, в целях аутентификации держателя, информация о карте из терминала передается в банк-эквайрер, обслуживающий данный терминал, и имеющий соглашение с владельцем торговой точки. В зависимости от договоренностей торговая точка оплачивает банку комиссию за его участие в обработке транзакции. Далее банк-эквайрер передает информацию в платежную систему, обслуживающую данную карту. Там данные попадают в операционный центр, к которому подключены банки-участники платежной системы. В этом центре проходит проверка на предмет наличия или отсутствия платежных данных карты в стоп-листе и в зависимости от полученного результата в транзакции отказывается или она одобряется с дальнейшим направлением в банк-эмитент, выпустивший данную карту, и обслуживающий привязанный к ней банковский счет/счета клиента. Здесь она попадает в процессинговый и авторизационный центр, в котором проводятся расширенные проверки на легальность обрабатываемой транзакции. При подозрении на мошенничество или нарушение условий обслуживания дается отказ. В зависимости от типа карты (дебетовая или кредитная) и установленного банком приоритета авторизации здесь может проводиться проверка доступного остатка средств на счете или платежного лимита, а также сверяться авторизационный PIN-код держателя. При удовлетворении всем проверкам эмитент одобряет операцию и в рамках транзакции, также через платежную систему, ответ дается в торговую точку. Путем взаиморасчетов с платежной системой эмитент перечисляет эквайреру сумму запрашиваемых по транзакции средств, а также комиссию платежной системы за обработку транзакции. В свою очередь с клиентского счета банк списывает оплачиваемую и подтвержденную клиентом к оплате сумму денег (для дебетовых карт) или уменьшает доступный платежный лимит, тем самым резервируя часть средств к последующему списанию (для кредитных карт). Транзакция завершается в момент поступления обратно в торговую точку ответа с одобрением или отказом.

Терний на пути транзакции в сети Биткоин существенно меньше. Программное обеспечение клиента пользователя, выполняющего платеж криптовалютой, отправляет транзакцию в сеть Интернет (причем могут использоваться самые разные каналы от Wi-Fi до мобильных). Транзакция подтверждается нодами (узлами) сети Биткоин и в конце концов одним из майнеров включается в блок. В свою очередь блок добавляется в блокчейн. На всякий случай

следует подождать добавления еще 6 блоков (ожидание займет не больше часа). Все. Платеж проведен. На самом деле наверняка времени на операцию уйдет больше, поскольку мы пренебрегли задержкой, в течение которой транзакция в составе блока попадет в блокчейн. Но согласитесь, действий потребовалось гораздо меньше, а прозрачность системы – несравненно выше.

Несколько утрированный взгляд на платеж в сети Биткоин приведен на рисунке 1.2. Здесь некая Марина расплачивается за выпитый кофе биткоинами, направляя их владельцу кофейни Сергею.

увеличить изображение

Рис. 1.2. Жизненный цикл транзакции

Полагаю, Вы согласитесь с простым доводом, что всегда найдутся предметные области, для которых использование криптовалюты гораздо предпочтительнее фиатных денег. Еще раз продекламируем один из тезисов этой книги – криптовалюта отнюдь не призвана сразу и навсегда заменить фидуциарные деньги. По крайней мере сейчас. Но право на свои ниши криптомонеты безусловно уже заслужили.

Преимущество биткоин-транзакции перед банковской онлайн транзакцией:

Избавление от посредников (банков-корреспондентов).

Высокая скорость проведения.

Меньшая цена транзакции.

Низкая вероятность возникновения ошибок.

Децентрализация.

Потрясающая устойчивость, отсутствие единых точек, выход из строя которых может обрушить систему.

P2P-взаимодействие.

Биткоин-транзакцию отменить невозможно.

Краткое описание и основные термины

Цель: Положить начало формированию понятийного и категориального аппарата предметной области, сформировать начальное представление о децентрализованном консенсусе и функционировании сети Биткоин.

Так что же все-таки такое блокчейн или распределенный реестр? Рассмотрим основные аспекты этой технологии на примере родоначальника цепочек блоков - криптовалюты Биткоин.

Платформа Биткоин — это набор концепций и технологий, образующих цифровой фундамент для экосистемы электронных денег. Токены (цифровые монеты), известные во всем мире как биткоины (BTC), используются для хранения или передачи ценностей между участниками сети (платформы). Пользователи криптовалютной платежной системы контактируют друг с другом посредством протокола Биткоин, реализуемым поверх сети Интернет. Однако, система может быть развернута в рамках какой-либо другой сети. Стек протоколов платформы Биткоин доступен в виде программного обеспечения с открытым исходным кодом и может быть запущен на различных устройствах, включая стационарные компьютеры, ноутбуки и мобильные устройства, что делает доступ к платежной системе повсеместным и легко осуществимым.

Оборот биткоинов в сети напоминает движение традиционных, фиатных денег. Владельцы BTC могут свободно их покупать, тратить на приобретение товаров, передавать друг другу и т.д. Покупка или обмен биткоинов на фидуциарные деньги или другие альткоины может производиться на специализированных криптовалютных биржах.

Несмотря на сходство в использовании биткоинов и фидуциарных денег, между ними имеется принципиальное отличие. BTC полностью виртуальны, т.е., не существует ни какого аналога физических купюр, ни даже монет в цифровом формате (как это бывает в случае электронных денег или чеков). Есть только записи (транзакции) в огромной бухгалтерской книге, фиксирующие передачу ценности от одного владельца другому.

Купюры определенного номинала декларируются в момент передачи ценности от одного участника сети другому или во время очередной эмиссии новой порции токенов. Каждый участник сети Биткоин владеют набором ключей, с помощью которых он способен доказать права владения на определенную сумму BTC в сети Биткоин. Эти ключи хранятся в специальных программных клиентах (а иногда и в физических устройствах), называемых цифровыми кошельками. Обладание ключами – единственное необходимое требование для совершения платежа (создания транзакции). Т.е., в сети Биткоин контроль над денежными средствами полностью ложится на самих пользователей, влияние какого-нибудь центрального звена (например, банка) принципиально исключается.

Ключ (от англ. key) - фраза (последовательность битов), на основе которой криптографическая система может выполнить шифрование и дешифрование криптограмм, манипуляции с электронной подписью (подписание документа и верификация подписи), а также решить задачи идентификации и аутентификации. В зависимости от типа криптосистемы ключ может быть симметричным (один и тот же ключ применяют для шифрования и дешифрования) и ассиметричные (используется криптопара: открытый (публичный) и закрытый (приватный) ключи).

Криптография - наука о технологиях обеспечения конфиденциальности (сокрытия контента от посторонних лиц), целостности (поддержание актуального состояния данных) и аутентификации (подтверждение подлинности (аутентичности) субъектов или объектов).

Выход в сеть в 2009 году нового платежного инструмента – платформы Биткоин - явился кульминацией многолетних исследований в области криптографии, математики и распределенных систем. В качестве опорных технологий новой платежной системы выступают:

Децентрализованная пиринговая (P2P, peer-to-peer) сеть (протокол Биткоин).

Доступный всем реестр (бухгалтерская книга) транзакций (так называемый блокчейн).

Децентрализованная математически и детерминистически эмиссия денег (так называемый майнинг).

Система децентрализованной проверки (валидации) транзакций.

С технологиями разберемся несколько позже. А пока обозначим две концепции, на которых во многом основана идея распределенных реестров. Это конечно же уже упоминавшаяся децентрализация и механизм достижения консенсуса среди не доверяющих друг другу пользователей системы (а также ее полная прозрачность).

Начнем наш экскурс с децентрализации. Идея сам по себе простая, но реализовать ее в полном объеме не удавалось долго. Суть в том, что каждый участник, имеющий доступ к распределенному реестру, хранит у себя его полную и актуальную копию. Если кто-то вносит изменения в блокчейн, то его копия тут же (на самом деле после проверки) рассылается всем остальным пользователям системы. С такой архитектурой система получается чрезвычайно отказоустойчивой. Может поломаться все, но, если остался хотя один участник блокчейн-сети с неразрушенным реестром, имеется принципиальная возможность полностью восстановить работоспособность платформы без каких-либо потерь. Замечательное качество технологии блокчейн.

К сожалению, это же самое свойство вызывает определенные трудности в реализации децентрализованных приложений. Поскольку в реестр изменения могут вносить все, то кто гарантирует достоверность хранимой в нем информации? Напомню, что мы имеем дело с системой, в которой между участниками нет доверия. В централизованных системах консенсус обеспечивается каким-нибудь внешним арбитром, например, головным сервером или банком, если речь идет о платежной системе. Участники могут не доверять друг другу, но в отношении

надежности арбитра у них никаких сомнений нет. В полностью децентрализованной системе мы сталкиваемся с так называемой задачей византийских генералов. И в 2008 году ее решение было предложено Сатоши Накомото, а в 2009 реализовано в первой в истории криптовалюте – Биткоин. Собственно, с этого момента и появился блокчейн как отдельное направление.

В сети Биткоин гарантом консенсуса между участниками выступает математика, а точнее криптография. Вот уж действительно случай, когда можно говорить о полной беспристрастности и объективности.

Чтобы разобраться в этом вопросе, прежде всего нужно вспомнить, что из себя представляет распределенный реестр. Почему мы говорим о технологии block chain? На самом деле именно с цепочкой блоков мы и имеем дело. Если речь идет о биткоине, то в реестре хранится информация об эмиссии биткоинов и передаче имеющихся в сети биткоинов между ее участниками. Каждая такая одиночная операция называется транзакцией. Объединять в цепочку транзакции – не самый оптимальный вариант. Поэтому еще пока не проведенные, ждущие своей очереди транзакции предварительно сводятся в блоки. Т.е. перевод с одного кошелька на другой в биткоин-сети не актуализируется сразу же после выполнения операции. Сформированная транзакция передается в сеть и какое-то время считается не завершенной. Только после включения в распределенный реестр нового блока с данной транзакцией можно говорить о завершении перевода (на самом деле рекомендуется дождаться формирования последовательности еще из пяти-шести блоков).

Занимаются сборкой блоков особые участники сети Биткоин – майнеры. Как только блок сформирован (параметры сети Биткоин (числа Target и Nonce см. Таблицу 1.2) настраиваются так чтобы это происходило в среднем один раз в десять минут) он распространяется по сети для подтверждения другими майнерами. И в случае успеха добавляется в конец цепочки.

Таблица 1.2. Упрощенная структура блока в блокчейне

Prev_block	Хеш предыдущего блока
Merkle_root	Хеш транзакций, включенных в блок
Timestamp	Временная метка (дата и время создания этого блока)
Bits	Число, называемое Target – значение, автоматически задаваемое самой биткоин-сетью, регулирующее сложность задачи майнинга. Подстраивается таким образом, чтобы на вычисление параметров нового блока уходило в среднем 10 минут
Nonce	Изменяемый майнером параметр в попытках сформировать такой блок, чтобы его Хеш отвечал заданному формату - был меньше чем target (при каждой новой попытке просто увеличивается на 1, начиная с 0)
Txn_count	Количество транзакций в блоке

Далее идет список транзакций

Очень важный аспект формирования цепочки заключается в том, что каждый новый блок рассчитывается на основании предыдущего, а фактически – всей сформированной на данной момент цепочки. Добиться этого помогает криптографии, а точнее хеш-функции. Делается это примерно так. Блок включает в себя список проверенных майнером транзакций (а вдруг кто-то отправил на адрес своего приятеля несуществующие биткоины) и заголовок, содержащий небольшой набор служебных полей. Важнейшим среди которых является ссылка на хеш предыдущего блока. Для вновь созданного блока вычисляется его хеш. При этом учитываются и включенные в блок новые транзакции (на самом деле для каждой транзакции предварительно считается ее хеш) и заголовок блока. Таким образом, в каждом новом блоке, присутствует вся предыстория блокчейна, включая самый первый блок (блок генезиса, в котором первые в истории 50 биткоинов Сатоши Накомото отправил сам себе). Благодаря этому практически невозможно выдернуть какой-нибудь блок из середины и вставить вместо него свой – содержащий неправильные транзакции. Хеш следующего блока тут же престанет соответствовать цепочке, поскольку в момент создания он рассчитывался, исходя из других предпосылок. Факт подмены, благодаря прозрачности распределенного реестра, тут же будет обнаружен другими участниками. Более того, не получится заменить не только "атакованный" блок, но и всю последующую за ним цепочку блоков, чтобы подмена в новом реестре не обнаруживалась. Злоумышленнику просто на это не хватит вычислительной мощности и времени. Вспомните про десять минут, через которые появится следующий новый блок. Оказывается, что формирование блока в биткоин-сети чрезвычайно затратная и сложная операция. На самом деле майнеру недостаточно просто вычислить хеш собранного блока. Этот хеш должен удовлетворять определенным условиям. Для хеширования в биткоине предусмотрен алгоритм SHA-256. Т.е., результирующий хеш будет представлять число - последовательность 256 бит. Представьте себе, что в системе принимаются хеши только определенного вида, например, первые 72 бита обязательно должны быть нулевыми. Для того чтобы варьировать хеши в заголовке предусмотрен специальный числовой параметр (Nonce), меняя который майнер и пытается достичь желаемого результата. Каким-то образом вычислить нужное значение параметра (фактически взломать алгоритм SHA-256) за такой короткий промежуток времени практически невозможно. Счастливчик, которому удалось решить эту сложную задачу, отправляет "проект" нового блока другим майнерам и в качестве доказательства своих честных намерений и проделанной большой работы (proof of work) предъявляет подобранное значение параметра - Nonce. Остальным майнерам остается только проверить действительно ли при таком заявленном параметре Nonce получается хеш блока заданного формата (меньше чем число Target). Эта задача существенно проще и не требует много времени и вычислительных затрат. Именно поэтому говорят, что консенсус в сети Биткоин основан

на доказательстве работы. В качестве которого выступает подобранный майнером параметр Nonce для собранного им нового блока-кандидата.

Следует отметить, что в других криптовалютах консенсус может достигаться иным путем. Известны подходы, основанные на подтверждении доли, подтверждении важности и др. В любом из этих случаев затея подделать реестр оказывается экономически нецелесообразной.

Высотой блока называют общее число блоков в цепочке, выстроенных после блока генезиса. Высота блокчейна – высота самого последнего блока в распределенном реестре.

В задачу пиринговых сетей входит обеспечение надежных коммуникаций всех участников сети друг с другом напрямую, минуя всевозможные центральные серверы. Работая поверх стека протоколов TCP/IP, такая сеть обеспечивает транспортировку транзакций, блоков-кандидатов и самого распределенного реестра. Фактически именно одноранговая peer-to-peer – сеть является технологической основой для достижения децентрализации блокчейна. Сетевые сервисы, использующие идею равноправия участников известны достаточно давно. Наибольшую популярность приобрели частично децентрализованные файлообменные системы, такие как eDonkey, KaZaA, BitTorrent и др. Но появление платформы Биткоин безусловно заставило посмотреть на сети peer-to-peer с иной точи зрения, и вдохнуло новую жизнь в эти протоколы.

И последний аспект сети Биткоин, который мы затронем в самом вначале, это транзакции. Собственно, как раз, ради их хранения и создавался реестр. Каждая транзакция – совершенно прозрачная, т.е. доступная всем участникам сети Биткоин запись о передачи некоторой суммы BTC с одного кошелька на другой. На самом деле в этой системе даже привычное для нас понятие счета отсутствует. Т.е. нельзя просто взять и посмотреть в блокчейне какую-нибудь ячейку и определить сколько на Вашем счету электронных денег. Для решения этой задачи программное обеспечение кошелька должно пробежаться по всему реестру в поиске транзакций, в которых в качестве получателя перевода указывался Ваш кошелек (на самом деле достаточно просмотреть только так называемый UTXO pool – список (множество) еще нерастраченных или незакрытых выходов транзакций).

Итак, что у нас входит в транзакцию?

Список входов. Количество входов может быть любым, но большим нуля разумеется. Каждый вход указывает откуда мы берем биткоины, т.е. выход одной из ранее проведенных в реестре транзакций.

Список выходов. Выходов тоже может быть несколько, на каждом указывается некоторая сумма. Каждый из выходов однозначно ассоциируется с кошельком получателя данной суммы.

И конечно же, каждая транзакция подписывается электронной подписью участника сети, осуществляющего данный перевод.

Поясним процедуру перевода на примере (см. Рисунок 1.3).

Предположим владелец кошелька Марина собирается перевести владельцу кошелька Сергею 5 биткоинов. На всякий случай упомянем, что криптовалюта биткоин – анонимная. Т.е., кто такая эта Марина или Сергей никто не знает. По крайней мере в сети Биткоин такой информации (персональных данных) точно нет.

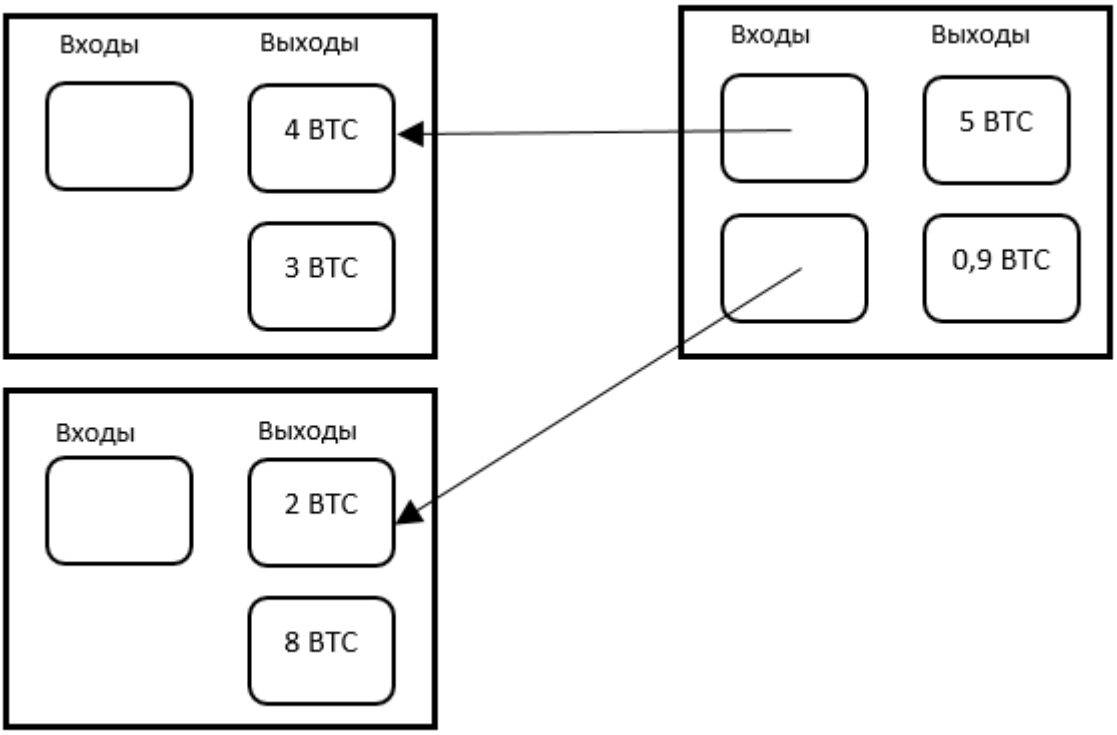

Рис. 1.3. Перевод биткоинов

Сначала Марина должна решить с выходов каких транзакций из реестра она должна взять эти биткоины. Есть ограничение – нельзя использовать только некоторую часть суммы. Вся наличность, имеющаяся на выбранном выходе транзакции, должна тратиться полностью. Ну и конечно эти деньги не должны быть уже потрачены, т.е. мы говорим только о тех выходах транзакций, которые входят в UTXO pool (конечно кошелек не позволит потратить несуществующие деньги, мошенникам придется идти другим путем). Пусть Марина владеет выходами двух транзакций с достаточной суммой биткоинов. С помощью программного обеспечения кошелька она создает транзакцию с двумя входами, ссылающимися на два ассоциированных с Мариной выхода из множества нерастраченных выходов. Соответственно, первый выход транзакции предназначен Сергею (и после включения транзакции в блокчейн только Сергей сможет распоряжаться этими 5 биткоинами), а второй выход достанется снова Марине. Поскольку к оплате было привлечено больше биткоинов, чем требовалось, следовательно, необходимо сформировать сдачу.

Внимательный читатель легко обнаружит пропажу 0,1 BTC или 10 000 000 сатоши. Куда же делась эта десятая часть биткоина? Ошибки никакой нет. Сумма входов транзакции может

превышать сумму ее выходов (но не наоборот). Разница будет выдана майнеру, который эту транзакцию обработает, в качестве вознаграждения. И если Вы хотите, чтобы Ваш платеж прошел поскорее (транзакция оказалась в блокчейне), предусмотрите достойную награду майнерам. Будьте уверены, что за транзакцию с таким хорошим вознаграждением они возьмутся в первую очередь.

Сатоши – минимальная, неделимая цифровая единица ценности в платформе Биткоин. 1 BTC = 100 000 000 сатоши.

Виды транзакций

Самая распространенная форма транзакции – конечно же простой платеж с одного адреса на другой, включающая второй выход с целью начисления сдачи отправителю (один вход и два выхода).

Другим распространенным видом платежа является транзакция, объединяющая несколько входов в один выход (аналог обмена множества мелких монет на одну крупную купюру в реальной жизни).

Довольно часто выполняется по сути обратная по смыслу финансовая операция - распределение средств одного входа по нескольким выходам, содержащим адреса нескольких разных получателей. Таким образом удобно расплачиваться с группой сотрудников, выполнивших определенный объем работы.

Разница между полными нодами и облегченными кошельками

Нода (от англ. Node — узел) – любая вычислительная система, которая подключается к сети Биткоин, используя протоколы пиринговой сети, позволяющие нодам коммуницировать друг с другом и распространять по сети транзакции и блоки.

Ноды, полностью реализующие все правила платформы Биткоин, называют полными. Такие узлы полностью синхронизированы с блокчейном. Следовательно, они обязаны хранить полный набор файлов распределенного реестра (более 230 GB). Именно полные ноды составляют основу платформы Биткоин.

Полные ноды загружают каждый блок и транзакцию и проверяют их на основе правил консенсуса сети Биткоин. Например, эмиссия ограничена определенным количеством биткойнов, транзакции должны иметь стандартный формат и содержать корректные подписи для соответствующих выходов, исключение двойных трат и так далее.

Если в рамках проверки будет выявлена нелегитимность транзакции или блока, то нода признает такую структуру данных недействительной, даже если остальные узлы вынесут положительный вердикт. Более того, нода на определенное время перестанет принимать данные от источника - инициатора запрещенной операции. Таким образом полные ноды, благодаря своей полной беспристрастности и скрупулезности, обеспечивают высокую степень безопасности платформы.

Облегченные ноды таким качеством не обладают. В основном они доверяют результатам работы майнеров и полностью на них полагаются. Как результат – на некоторое время их можно ввести в заблуждение в отношении легитимности некоторого блока или транзакции. Полные ноды являются единственным надежным способом обеспечения максимальной безопасности криптомонет. Обладая полным объемом информации о блокчейне, они способны гарантировать неукоснительное соблюдение всех правил.

Не стоит путать сетевые роли полной ноды и узла майнера. Само по себе владение полной нодой и участие в проверке транзакций не принесет Вам дохода. Однако, именно от стабильности работы полных нод зависит стабильность платформы Биткоин.

Большинство приложений кошельков поддерживает небольшую базу данных неизрасходованных выходов транзакций – пул UTXO, заблокированных закрытыми ключами кошелька. Ноды кошельков, работающие в режиме полных нод, на самом деле содержат копию каждого неизрасходованного выхода. Благодаря этому кошелек способен не только выбирать входы для транзакций, но и оперативно проверять корректность входов транзакций. Однако, поскольку хранение всего блокчейна требует значительных ресурсов, большинство пользователей используют облегченные кошельки, отслеживающие только неизрасходованные выходы конкретного пользователя.

Впрочем, у облегченных кошельков всегда имеется возможность запросить всю необходимую информацию у платформы Биткоин, используя API различных провайдеров, или послав JSON RPC-запрос программному обеспечению полной ноды.

Пользователи платформы постоянно создают новые транзакции, отражающие движение денежных средств. В сети они добавляются к временному пулу непроверенных транзакций, поддерживаемому каждой нодой. В процессе формирования блока каждый майнер обращается к этому пулу, выбирает необходимое число транзакций и рассчитывает остальные параметры блока.

Краткие итоги

Системы распределенного реестра в федеральной программе "Цифровая экономика Российской Федерации" включены в число основных сквозных цифровых технологий, призванных повысить конкурентоспособность России на глобальном рынке. Широта и возможности применения технологии блокчейн видятся весьма значительными, с ее помощью можно трансформировать целые сектора коммерческой, общественной или государственной деятельности. Например, устранить посредников в сложившихся цепочках хозяйственных связей. При этом на пути развития децентрализованных приложений возникает множество разнообразных препятствий.

Распределенные системы могут быть как централизованными, так и децентрализованными. Причем, приложения, использующие блокчейн и пиринговые технологии, могут быть распределенными и децентрализованными одновременно.

В сети Биткоин гарантом консенсуса между участниками выступает криптография. Важнейшим аспектом формирования блокчейна является то, что каждый новый блок рассчитывается на основе предыдущего, а фактически – всей сформированной на данной момент цепочки блоков. Занимаются сборкой блоков особые участники сети Биткоин–майнеры.

В реестре сети Биткоин хранится информация об эмиссии биткоинов и передаче ранее выпущенных биткоинов между ее участниками. Каждая такая одиночная операция называется транзакцией. В состав транзакции в сети Биткоин входят список входов и список выходов. Каждая транзакция подписывается электронной подписью владельца средств. UTXO pool – список (множество) еще нерастраченных выходов транзакций.

Минимальной, неделимой цифровой единицей ценности в платформе Биткоин является сатоши. 1 BTC = 100 000 000 сатоши.

Нода – любая вычислительная система, которая подключается к сети Биткоин, используя протоколы пиринговой сети, что позволяет узлам коммуницировать друг с другом и распространять по сети транзакции и блоки. Ноды, полностью реализующие все правила платформы Биткоин, включая хранение полного набора файлов распределенного реестра на локальном носителе, называют полными. Именно полные ноды, наряду с майнерами, составляют основу платформы Биткоин.

Лекция 2. Криптографические ключи, адреса, кошельки

Почему это важно?

Цель: Сформировать понимание важности криптографических алгоритмов для криптовалют.

Право владения токенами, в том числе, биткоинами устанавливается через криптографические (цифровые) ключи, Биткоин-адреса и цифровые подписи. Закрытые криптографические ключи не перемещаются по сети. Они генерируются и хранятся пользователями в специализированном клиенте (кошельке). Цифровые ключи в кошельке пользователя являются абсолютно независимыми от протокола Биткоин, генерируются и управляются с помощью программного обеспечения кошелька пользователя без обращений к блокчейну или к сети Интернет. Благодаря такой стратегии управления ключами становятся возможными многие из важнейших свойств сети Биткоин, в том числе децентрализованные консенсус и контроль, подтверждение владения, и модель безопасности, основанная на математическом (криптографическом) доказательстве.

Каждая транзакция в сети Биткоин должна быть подписана подлинной электронной подписью, которая может быть получена только при наличии валидных цифровых криптографических ключей. Следовательно, любой, получивший копии данных ключей, имеет точно такой же контроль над средствами, ассоциированными с этой учетной записью, как и ее истинный владелец.

Ассиметричная схема шифрования подразумевает наличие криптопары: частного (закрытого) и публичного (открытого) ключей. Публичный ключ можно сравнить с номером банковского счета, тогда приватный ключ выполняет функцию PIN-кода или подписи на банковском чеке, обеспечивая полный доступ к учетной записи. Эти цифровые ключи почти никогда не попадаются на глаза пользователям сети Биткоин. В основном, они хранятся в файлах бумажника, а манипуляции ключами выполняет программное обеспечение кошелька – разумеется по инициативе владельца.

Если рассматривать процедуру платежа, то в соответствующей транзакции фигурирует Биткоин-адрес получателя, являющийся фактически цифровым отпечатком открытого ключа. Сравним его с именем получателя денежных средств на банковском чеке.

В большинстве случаев биткоин-адрес формируется на основе публичного ключа. По крайней мере, это утверждение справедливо в отношении всех обладателей собственных аккаунтов. Забегая вперед скажем, что получателями средств могут быть сценарии. Их биткоин-адреса определяются по иной схеме.

Таким образом, в рамках сети Биткоин адреса обеспечивают анонимность пользователей. Не существует способа ассоциировать биткоин-адрес с конкретным физическим или юридическим лицом. Но эта анонимность поддерживается только в рамках сети Биткоин. Использование токенов на криптобиржах или магазинах, принимающих криптовалюту, в большинстве случае потребует

стандартной идентификации владельца криптосредств. Из выше сказанного следует, что в отличие от стандартной схемы цифровой подписи даже публичный ключ не распространяется по сети и не передается другим пользователям. Вместо него для решения задач авторизации используются биткоин-адреса.

Криптография с открытым ключом

Цель: Определить "точки пересечения" ассиметричной криптографии и протокола Биткоин.

Криптография с открытым ключом была изобретена в 70-х годах прошлого века. Именно ассиметричные криптоалгоритмы традиционно стали использоваться в качестве математической основы при построении систем компьютерной и информационной безопасности. В процессе становления криптографии с открытым ключом было найдено несколько классов математических функций, получивших название односторонние. К ним, в частности, относятся возведение в степень простого числа, умножение эллиптических кривых и др. Необратимость означает, что прямые значения такого рода функций вычисляются достаточно просто, а вот обратные рассчитать практически невозможно. На основе односторонних функций разработано довольно много алгоритмов цифровых шифров и криптостойких электронных подписей.

В сети Биткоин для создания криптопары, контролирующей доступ к счету, используется умножение на эллиптических кривых. Так же, как и в других ассиметричных криптосистемах пара ключей состоит из закрытого ключа и производного от него, уникального открытого ключа. Публичный и приватный ключи однозначно связаны математическим соотношением. Такая связь обеспечивает возможность подписи сообщения приватным ключом с одной стороны, и последующую проверку правильности подписи при помощи публичного ключа, с другой. При этом приватный ключ не раскрывается. В платежных операциях открытый ключ используется для получения биткоинов (как основание для определения адреса), а с помощью закрытого ключа подписываются транзакции, предназначенные для траты имеющихся на балансе счета средств (на самом деле понятие баланса в привычном понимании в сети Биткойн не применяется). Фактически, любой пользователь, намереваясь потратить имеющиеся у него биткоины, должен опубликовать в сети свои открытый ключ и электронную подпись транзакции, полученную с помощью соответствующего закрытого ключа. После публикации этих данных любой участник сети (и, в первую очередь, майнеры) могут проверить правомочность совершаемых финансовых действий и признать транзакцию действительной. Сделка будет считаться совершенной после того как проверенная транзакция в составе очередного валидного блока будет включена в блокчейн.

Поскольку открытый ключ в любой момент может быть вычислен на основе закрытого ключа, некоторые кошельки хранят только приватные ключи.

Управление ключами в сети Биткоин

Цель: Сформировать понимания процессов преобразования цепочки криптографических преобразований: Закрытый ключ → Открытый ключ → Биткоин-адрес.

При создании биткоин-кошелька на компьютере создается специальный файл, содержащий в себе две записи: private key (закрытый ключ) и public key (открытый ключ). Обычно это что-то типа wallet.dat. И если закрытый ключ генерируется случайным образом, то открытый ключ создается путем криптографического преобразования закрытого ключа. Если быть точным в Биткойн-сети используется алгоритм эллиптической криптографии secp256k1 – вариация широкоизвестного Алгоритма Цифровой Подписи с Эллиптическими Кривыми (ECDSA).

Закрытый ключ - случайное число длиной в 256 бит, генерируемое для каждого счета пользователя. Чтобы доказать окружающим, что приватный ключ у пользователя имеется, и не раскрыть его при этом, вычисляется второе число —публичный ключ. Для этих целей как раз и используется криптосистема ECDSA. Преобразование это одностороннее, т.е. выполнение обратной операции - вычисление закрытого ключа по открытому – является практически невыполнимой задачей.

Наконец, есть еще и третье число: так называемый "адрес кошелька". Как и в любой другой системе адрес нужен для целей идентификации пользователей. Отправлять биткоины "на деревню, дедушке" конечно же в голову никому не придет. В принципе публичный ключ сам мог бы послужить адресом, более того, первый протокол платформы Bitcoin именно это и предполагал. Но достаточно быстро было принято решение перейти на другую схему вычисления адреса. Он стал короче по сравнению с ключами - всего 160 бит и, одновременно, безопасней. Адрес вычисляют на основе публичного ключа путем последовательного двукратного вычисления хеш-функций (сначала SHA-256, а затем RIPEMD-160).

Связь между криптопарой (закрытым и открытым ключами) и биткоин-адресом продемонстрирована на рисунке 2.1.

Рис. 2.1. Приватный ключ, публичный ключ, и биткоин-адрес

Рассмотрим эти вопросы подробнее.

Закрытые ключи

Цель: Сформировать четкое понимание процесса и развить практические навыки получения закрытых ключей в сети Биткоин.

Закрытый ключ — это просто число, взятое наугад. Только владелец закрытого ключа контролирует средства, связанные с определенным биткоин-адресом.

Секретный ключ используется для создания электронной подписи, необходимой в качестве неопровержимого доказательства права владения средствами в транзакции. Секретный ключ должен храниться в строжайшем секрете. Его несанкционированное копирование по сути означает передачу контроля над средствами в чужие руки.

Следует позаботиться о создании резервной копии секретного ключа. В случае его потери доступ к соответствующим средствам также будет утрачен. Они окажутся замороженными в блокчейне сети Биткоин навсегда. Не существует никакого легального способа ими воспользоваться. Ни у кого в целом мире, включая Сатоши Накомото.

Секретный ключ — это просто целое число. Можно сформировать валидный секретный ключ, прибегнув к дедовскому способу - подбрасыванию монеты. Правда подбрасывать ее придется 256 раз. Записав результаты этого эксперимента Вы получите двоичную запись случайного секретного ключа, который можно использовать в биткоин-кошельке. Разумеется, протокол платформы Биткоин предусматривает более современные способы генерации приватных ключей.

Задача осложняется тем, что большинство привычных инструментов генерации случайных чисел выдают, так называемые, псевдослучайные последовательности, не основанные на надежных источниках энтропии. Самый надежный способ состоит в использовании квантовых генераторов случайных бинарных последовательностей, примеры практической реализации которых уже известны. Основаны такие устройства на принципиально ином источнике случайности, чем большинство традиционных генераторов. А именно, квантовой неопределенности. Согласно фундаментальным законам квантовой механики, у электрона или другой частицы нет траектории, которую можно проследить. Есть лишь вероятность обнаружить частицу в той или иной области пространства. И подобную энтропию невозможно устранить даже теоретически. На самом деле такая неопределенность характерна не только для движения элементарных частиц, но и присуща, по сути, всем квантовым процессам. В обозримом будущем, наверняка, именно такие генераторы существенно снимут остроту проблемы. Но пока на практике прибегают к более предсказуемым, а потому ненадежным инструментам генерации случайных чисел.

Создание секретного ключа для платформы Биткоин, по сути, аналогично выбору числа в диапазоне от 1 до 2256 (на самом деле используется число, несколько меньшее чем 2256, верхняя граница диапазона задается числом n - 1, где n = 1,158 · 1077 - порядок эллиптической кривой, используемой в платформе Биткоин). Программное обеспечение сети Биткоин использует штатные генераторы случайных чисел, входящие в состав операционных систем. Как правило, генератор

случайных чисел стандартной операционной системы инициализируется источником энтропии, исходящим непосредственно от пользователя. Обычно, это сводится к тому, что пользователю предлагается некоторое время шевелить мышкой. Далее к результату оцифровки сформированной траектории применяется хеш-функция SHA256, на выходе которой мы получаем 256-битное число. Остается только сравнить его с верхней границей n. Если полученное число меньше величины n - 1, тогда мы нашли подходящий закрытый ключ для вновь образуемого счета в сети Биткоин (сравнение нового значения с уже сформированными в сети закрытыми ключами не производится!).

Ниже приведен случайным образом сгенерированный секретный ключ (k) в шестнадцатеричном формате (64 шестнадцатеричных цифры):

1E88423A4ED27609A15A2616A2B0E8E52CED330AC530EDCC32C9FFC6A526AEDD

Отметим, что пространство возможных приватных ключей платформы Биткоин (от 1 до 2256) имеет колоссальные размеры. Если обратиться к более привычной для большинства людей десятичной системе счисления, то это примерно от 1 до 1077. По имеющимся оценкам видимая часть Вселенной состоит всего-то из 1080 атомов – вполне сравнимые объемы.

Отображение секретного ключа в виде числовой последовательности не всегда является удобным. Часто прибегают к более прогрессивному формату WIF (Wallet Import Format). Значение в таком формате получается довольно просто. Двоичную запись ключа предваряют специальным префиксом – числом 128 (0x80 в шестнадцатеричной системе счисления). Полученное значение записывают в формате Base58Check. О кодировке Base58Check чуть позже мы поговорим подробнее. Пока представим пример записи секретного ключа в формате WIF:

секретный ключ = 0a56184c7a383d8bcce0c78e6e7a4b4b161b2f80a126caa48bde823a4625521f

секретный ключ в формате WIF = 5HtqcFguVHA22E3bcjJR2p4HHMEGnEXxVL5hnxmPQvRedSQSuT4.

Открытые ключи

Цель: Сформировать четкое понимание процесса получения открытых ключей в сети Биткоин.

Открытый ключ K получается из секретного путем одностороннего скалярного умножения на эллиптических кривых базовой точки на значение секретного ключа.

$K = k \cdot G$,

где k — это приватный ключ, G — базовая точка,

Обратная операция вычисления k (нахождение дискретного логарифма) при известном K возможна только при помощи полного перебора k, т.е. весьма непродуктивной атаки типа "brute force". Прежде, чем мы перейдем к генерации публичных ключей немного поговорим о криптографии на эллиптических кривых.

Криптография на эллиптических кривых

Криптография на эллиптических кривых — это вид асимметричной криптографии или криптографии с открытым ключом, основанная на проблеме дискретного логарифмирования на эллиптических кривых.

На рисунке 6 продемонстрирован пример эллиптической кривой, аналогичной той, которая используется в платформе Биткоин.

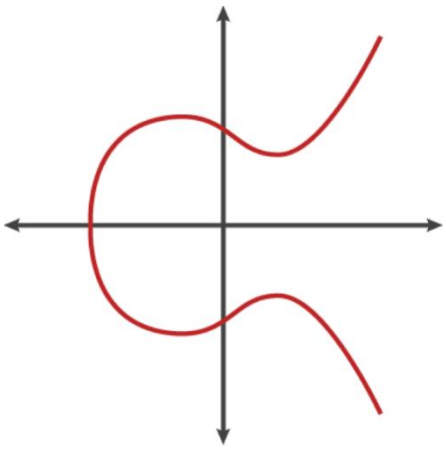

Рис. 2.2. Пример эллиптической кривой

Эллиптические кривые обладают целым набором полезных свойств. Например, любая наклонная прямая, пересекающая эллиптическую кривую в двух точках, всегда будет пересекать ее также в третьей точке. Следующим фактом является то, что любая наклонная прямая, являющаяся касательной к кривой в одной из точек, обязательно пересечет кривую еще ровно в одной точке. Эти свойства окажутся востребованными в криптосистемах.

В сети Биткоин используется конкретная эллиптическая кривая и набор математических констант из стандарта под названием secp256k1 установленного Национальным Институтом Стандартов и Технологий (NIST).

Уравнение эллиптической кривой: y2 = x3 + 7

Определенной над полем y2 mod p = x3 + 7 (mod p)

Операция нахождения модуля простого числа p - (mod p) показывает, что эллиптическая кривая определена над конечным полем простого порядка p, где p = 2256 - 232 - 29 - 28 - 27 - 26- 24 - 1является очень большим простым числом.

Простой модуль p = 2256 - 232 - 29 - 28 - 27 - 26 - 24 - 1 = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFC2F

Базовая точка:

X = 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

Y = 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8

Порядок = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141 - количество точек кривой над конечным полем.

В ECDSA секретный ключ — это случайное число между единицей и значением порядка.

Открытый ключ = секретный ключ · значение базовой точки.

Постараемся в этом разобраться подробнее.

Эллиптическая кривая над полем K — это кубическая кривая над алгебраическим замыканием поля K, задаваемая уравнением третьей степени с коэффициентами из поля K и "точкой на бесконечности". Одной из форм эллиптических кривых являются кривые Вейерштрасса

$$y2 = x3 + a*x + b$$

Для коэффициентов a = 0 и b = 7 (используемых в платформе Биткоин), график функции изображен на рисунке 2.2.

В эллиптической криптографии используется такая же кривая, но определенная над некоторым конечным полем.

$$y2 \bmod p = x3 + a*x + b \pmod p$$

Конечное поле в контексте эллиптической криптографии можно представить, как предопределенный набор целых, положительных чисел, которому должен принадлежать результат любого вычисления. Например, 11 mod 8 = 3. Здесь мы имеем конечное поле от 0 до 7, и все операции по модулю 8, над каким бы значением они ни осуществлялись, приведут к результату из этого диапазона.

Поскольку эллиптическая кривая определена над конечным полем простого порядка, а не вещественных чисел, она выглядит как узор из точек, рассеянных в двух измерениях, что достаточно трудно визуализировать. Тем не менее, математика идентична математике эллиптической кривой в вещественных числах. В качестве примера, на рисунке 2.3 отображены эллиптические кривые над полем очень скромных конечных простых порядков (p = 17, p = 59), иллюстрирующая то, каким образом располагаются точки на координатной сетке. Эллиптическая кривая, используемая в криптосистеме платформы Биткоин, secp256k1 представляет собой гораздо более сложную совокупность точек, распределенных на неизмеримо большей координатной сетке.

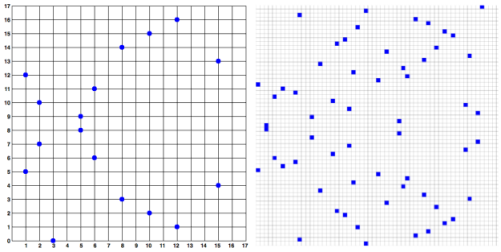

Рис. 2.3. Криптография эллиптических кривых: эллиптическая кривая над F(p)

при p = 17 и при p = 59

Эллиптическая кривая на множестве вещественных чисел (непрерывные числовые значения, в том числе и дробные) не позволяет получить однозначное соответствие (биекцию) исходных данных их образам (хешам). Чтобы не усложнять, и без того не простые, расчетные методы еще и округлением, для целей криптографии используются только точки с координатами из множества конечных полей. Исходя из этого, на эллиптической кривой берутся точки со значениями координат попадающими в конечное поле.

Вспомним основные свойства эллиптических кривых. И постараемся понять, как их можно использовать в криптографии.

Эллиптическая кривая над полем есть неособая кубическая кривая на проективной плоскости над алгебраическим замыканием поля, задаваемая уравнением 3-й степени с коэффициентами из поля и "точкой на бесконечности"

В зависимости от значений параметров a и b график данной функции может выглядеть по-разному:

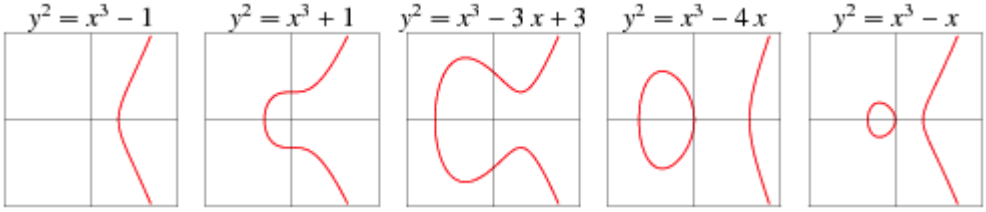

Рис. 2.4. Примеры эллиптических кривых

Математики давно заинтересовались эллиптическими кривыми. Первые упоминания о них находят в трудах Диофанта. А в 17 веке, ее свойства исследовал Ньютон. Именно его труды послужили основой для формализации правил сложения точек на эллиптической кривой.

Пусть на некой эллиптической кривой f определены две точки P,Q ∈ f. Их суммой называется точка R ∈ f, которая в простейшем случае определяется следующим образом. Проведем прямую через точки P и Q. Пересечение этой прямой с кривой f даст нам точку -R. Отобразив точки -R симметрично относительно оси x, получаем искомую точку R, которую будем называть суммой P + Q = R.

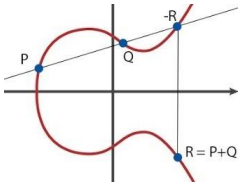

Рис. 2.5. Сумма точек на эллиптических кривых

Считаю необходимым отметить, что мы именно таким образом определяем операцию сложения. Попытка сложить точки согласно традиционного правила, то есть суммируя соответствующие координаты, даст совершенно другую точку, которая, скорее всего, не имеет ничего общего с точкой R и даже не лежит на кривой f.

В случае, когда складываются две точки, имеющие координаты вида P(a, b) и Q(a, -b), прямая пройдет параллельно оси ординат (третья слева ситуация на Рисунке 2.6).

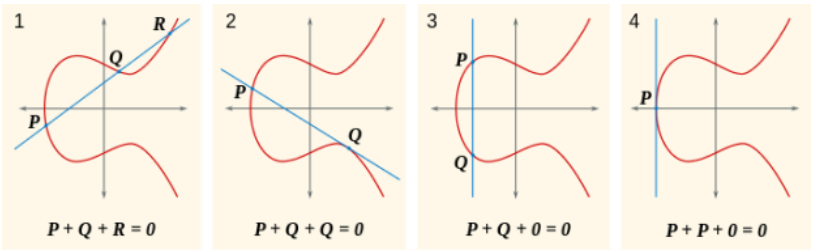

Рис. 2.6. Различные варианты сложения точек на эллиптических кривых

Как следует из рисунка, в этом случае отсутствует пересечение с кривой f в точке, которое мы называли -R. Для того, чтобы избежать подобной ситуации, введем так называемую точку в бесконечности (point of infinity), обозначаемую обычно O или просто 0. Установим, что в случае отсутствия пересечения P + Q = O.

Если P1 является точкой в бесконечности, тогда сумма P1 + P2 = P2. Аналогично, если P2 является точкой в бесконечности, тогда P1 + P2 = P1. Данное свойство демонстрирует, как точка на бесконечности играет роль нуля в эллиптической математике.

Особый интерес для нас представляет вариант, когда мы хотим сложить точку саму с собой (вторая слева ситуация на рисунке 10, точка Q). В этом случае сначала проводится касательная к точке Q, а полученная точка пересечения отображается относительно оси ординат.

Оказывается, что операция сложения на эллиптических кривых ассоциативна. Следовательно, (A + B) + C = A + (B + C). Т.е., мы можем написать A + B + C без скобок и такая запись будет восприниматься совершенно однозначно.

А теперь мы подошли к самому главному – умножению точки G на какое-то натуральное число k. В результате получим новую точку K = G · k, т.е., K = G + G + G + G + G + …, k раз. Алгоритм умножения точки на число на эллиптической кривой f проиллюстрирован на рисунке 2.7.

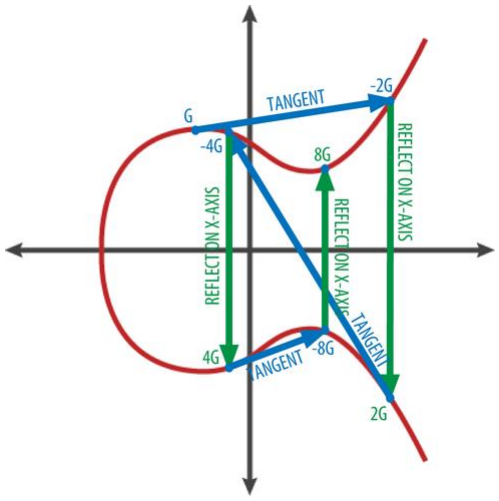

Рис. 2.7. Операция умножения точки на число на эллиптических кривых

Таким образом, скалярное умножение на эллиптических кривых это тоже самое, что и суммирование точки G самой с собой k раз подряд. В свою очередь, каждое "удвоение" точки G является эквивалентом построения касательной к упомянутой точке, нахождения

пересечения построенной прямой с эллиптической кривой и отображения найденной точки пересечения относительно оси абсцисс.

Создание открытого ключа

Цель: Сформировать практические навыки получения открытых ключей в сети Биткоин.

Пусть k — наш закрытый ключ, G — базовая точка, тогда открытый ключ K = k · G. То есть, фактически, открытый ключ — это некоторая точка, лежащая на кривой secp256k1. Поскольку, базовая точка одинакова для всех пользователей платформы Биткоин, закрытый ключ k при умножении на G всегда дает один и тот же открытый ключ – K.

Отметим два важных момента. Во-первых, операция вычисления открытого ключа определена однозначно, то есть конкретному закрытому ключу всегда соответствует ровно один единственный открытый ключ. Во-вторых, обратная операция является вычислительно трудной и, в общем случае, получить закрытый ключ из открытого можно только путем полного перебора первого – задача практически невыполнимая (по крайней мере, в настоящее время). На самом деле, точно также связаны открытый ключ владельца счета и биткоин-адрес (правда в этом случае результат достигается за счет необратимости хеш-функций). Именно поэтому биткоин-адрес (полученный из K) можно передавать кому угодно без риска дискредитировать закрытый ключ k и как результат - лишиться части своих криптозапасов.

Открытый ключ K определяется как точка на эллиптической кривой K = (x, y).

Если

k = 1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD, то

K = (x, y)

где,

x = F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A

y = 07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB.

Таким образом, открытый ключ в алгоритмах электронной подписи, основанных на использовании эллиптических кривых, является координатами точки на этих самых кривых. Т.е., это два числа – координаты X и Y. На самом деле используется несколько вариантов записи открытого ключа. Пока мы остановимся на так называемом uncompressed формате - по 32 байта для каждой из осей координат. Чтобы не возникало путаницы, используется специальный префикс 04 (0x04 в шестнадцатеричной системе), в итоге получается 65 байт.

Например, для секретного ключа k = 0a56184c7a383d8bcce0c78e6e7a4b4b161b2f80a126caa48bde823a4625521f

публичный ключ будет выглядеть следующим образом:

045fbbe96332b2fc2bcc1b6a267678785401ee3b75674e061ca3616bbb66777b4f946bdd2a6a8ce419eacc5d05718bd718dc8d90c497cee74f5994681af0a1f842,

где первый байт = 04 – префикс, означающий, что мы имеем дело с открытым ключом.

На всякий случай, вспомним как работают стандартные алгоритмы электронной подписи, а немного позже отметим особенности, присущие их реализации в сети Биткоин.

Электронная подпись - это криптосистема, обеспечивающая решение следующих задач:

контроль целостности электронных документов. Любое случайное или преднамеренное изменение содержимого документа автоматически сделает подпись недействительной.

убедительное доказательство подлинности авторства документа. Поскольку доступ к закрытому ключу есть исключительно у его владельца, следовательно, только он может создать корректную электронную подпись. Рассуждая аналогичным образом, электронную подпись также можно использовать для обеспечения неотказуемости от авторства документа.

Есть говорить о технической стороне вопроса, то на практике превалируют решения, реализующие схемы с открытым ключом. Поэтому в качестве иллюстрации на рисунке 12 приведен именно ассиметричный алгоритм электронной подписи.

Процесс начинается с фазы подписания документа электронной подписью. В реальных системах для получения электронной подписи полный текст документа обычно не используется, иначе мы столкнемся с большими временными и аппаратными издержками. Как известно, ассиметричные криптосистемы предъявляют гораздо больше требований к аппаратным ресурсам и работают медленнее по сравнению с симметричными. Поэтому текст документа перед подписанием подвергается предварительному сжатию с помощью функции хеширования. Полученный результат называют дайджестом. Свойства хеш-функции гарантируют следующие качества дайджеста:

Стандартные размеры - результат хеширования не зависит от объема текста на входе.

Уникальность для любых уникальных наборов символов – исключена возможность формирования одинаковых хешей для разных массивов данных.

Невозможность по его значению восстановить исходный текст – преобразование посредством хеш-функции является односторонним.

Рис. 2.8. Принцип работы ассиметричного алгоритма электронной подписи

Затем дайджест шифруется на закрытом ключе лица, подписывающего документ. Результатом этой операции является собственно электронная подпись – специальный, цифровой реквизит документа. Сведения о владельце ключа и зашифрованный хеш помещаются в специальный контейнер, прикрепляемый к документу. Так же в документ (но уже в другой контейнер) вкладывается открытый ключ подписанта. Весь указанный массив информации (документ и два контейнера) далее отправляется адресату.

Верификация электронной подписи реализуется следующим образом. Вычисляется дайджест полученного документа. Параллельно с этим дешифруется с использованием открытого ключа подписанта блок электронной подписи из контейнера. Если они совпадут, тогда результат верификации признается успешным. Это означает, во-первых, что подпись подлинная (основание для такого решения – открытый ключ соответствует закрытому, хранящемуся только у лица, подписавшего документ) и, во-вторых, что после подписания текст документа не менялся (иначе бы вычисленный и полученный путем дешифрации дайджесты не совпали).

Наиболее известными алгоритмами создания электронной подписи являются:

Криптосистема RSA, пригодная и для шифрования, и для электронной подписи. Такие популярные криптографические решения как PGP, S/MIME, TLS/SSL, IPSEC/IKE и ряд других построены на основе RSA.

DSA (основан на вычислительной сложности исчисления логарифмов в конечных полях) и ECDSA (основан на аппарате эллиптических кривых) - алгоритмы с открытым ключом, являющиеся американскими стандартами электронной подписи.

Схема Шнорра и др.

Биткоин-адреса

Цель: Сформировать четкое понимание процесса и развить практические навыки получения биткоин-адресов, включая подробное обсуждение кодировки Base58Check.

Решение использовать в качестве адреса хеш-функцию, а не открытый ключ основывалось на двух соображениях:

Криптосистемы на эллиптических кривых уязвимы для модифицированного алгоритма Шора при решении задачи дискретного логарифма на эллиптических кривых. Это означает, что появившиеся в будущем квантовые компьютеры смогут получить закрытый ключ из открытого ключа. Публикуя открытый ключ только тогда, когда монеты были потрачены (и при условии, что адреса не используются повторно), такая атака потеряет смысл.

С дайджестом ограниченного размера будет легче работать: печатать; встраивать его в небольшие носители информации, такие как QR-коды.

Биткоин-адрес представляет собой строку из цифр и символов латинского алфавита. Адреса, полученные на основе открытых ключей, начинаются с цифры 1 (так называемые pay-to-public-key hash адреса). Ниже приводится пример подобного биткоин-адреса:

1J7mdg5rbQyUHENYdx39WVWK7fsLpEoXZy.

Чаще всего в качестве получателя средств в транзакциях Биткоин фигурирует именно биткоин-адрес. Если сравнить транзакцию в сети Биткоин с бумажным чеком, то адрес получателя явился бы тем, что мы пишем в строке "Получатель платежа". В качестве получателя на бумажном чеке может значиться название фирмы, учреждения, или даже указание обналичить чек. Именно возможность использования абстрактного имени в качестве получателя на бумажном чеке обеспечивает значительную гибкость чековым финансовым системам. Платформа Биткоин использует сходную абстракцию. Кроме упомянутых биткоин-адресов, определяющих владельца счета (т.е., обладателя соответствующей криптопары) в сети Биткоин адресатом может быть назначен некий сценарий, который определяет, кто может распоряжаться выходами транзакции (pay-to-script hash, такие адреса начинаются с цифры 3). Платформа Биткоин поддерживает также мульти-подписи, когда основной сценарий требует более одной подписи для доказательства права собственности и, следовательно, предоставления возможности распоряжения средствами, ассоциированными с подобным счетом. В общем случае мульти-подпись M-из-N, реализованная в сети Биткоин, требует для разблокирования выхода транзакции наличия M подписей из Nвозможных. Разумеется, $M \leq N$.

Начнем с анализа самого простого случая, когда биткоин-адрес получается из публичного ключа с помощью односторонней криптографической функции хеширования. Заметим, что криптографические хеш-функции широко задействованы в реализации целого ряда принципиальных инструментов сети Биткоин: при создании биткоин-адресов или адресов сценариев, в алгоритме доказательства работы (Proof of work) в майнинге.

Адрес из публичного ключа получается в результате последовательного применения двух алгоритмов криптографического хеширования. А именно, Secure Hash Algorithm (SHA) и RACE Integrity Primitives Evaluation Message Digest (RIPEMD). Чтобы быть совсем точным, упомянем о том, что на первом этапе длина получаемого дайджеста составляет 256 бит (т.е., используется алгоритм SHA-256), а на втором – 160 бит (RIPEMD-160).

Процесс получения биткоин-адреса отображен на рисунке 13.

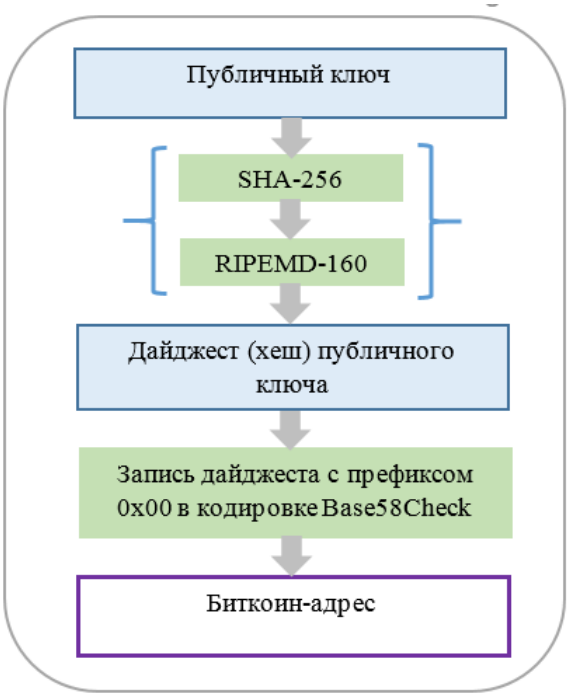

Рис. 2.9. Алгоритм формирование биткоин-адреса

Кодировка Base58Check

Для записи биткоин-адреса почти всегда используется кодировка Base58Check, мощность алфавита которой составляет 58 символов. Назначение ее очень простое – представить последовательность байт в простом и удобочитаемом формате и максимально снизить при этом вероятность возможных опечаток или ошибок. Понятно, почему это важно. С биткоин-адресами работают люди, которым вообще свойственно ошибаться. При работе с большими, не имеющими традиционного смысла наборами символов вероятность ошибок резко возрастает. При этом, чтобы безвозвратно потерять средства, достаточно ошибиться всего лишь в одном символе адреса. Правда разработчики платформы Биткоин в свое время об этом подумали и предусмотрели несколько механизмов защиты от ошибок. И первый из них – это использование для записи адресов кодировки Base58Check. Заметим, что вторая по популярности криптовалюта Эфириум такой защиты не имеет, хотя и появилась позже.

Само по себе представление Base-64 используется для передачи бинарных данных в "символьных средах", наподобие электронной почты. Алфавит формата Base-64 состоит из 26 прописных букв, 26 заглавных букв, 10 цифр, и двух дополнительных символов, которые в различных вариациях могут меняться. В свою очередь кодировка Base58 — это подмножество Base64, использующее прописные и заглавные буквы, цифры, но из набора символов исключаются те, которые часто путают, а именно 0 (ноль), O (заглавная буква O), l (маленькая L), I (большая i). Не включены в алфавит и два выше упомянутых дополнительных символа.

Т.е., алфавит Base58 включает следующие символы:

123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz

От опечаток при наборе адреса призвана спасти кодировка Base58Check. Фактически это расширение строки, записанной с помощью Base58, путем добавления кода проверки ошибок. Четыре байта вычисленной контрольной суммы, полученной путем хеширования исходных данных, добавляются в конец числовой последовательности. Программное обеспечение кошелька вычисляет контрольную сумму сравнивает ее с контрольной суммой из вводимого в систему адреса. Если данные не совпадают, значит при вводе была допущена ошибка. Подобная проверка позволяет предотвратить отправку средств по несуществующему биткоин-адресу.

Предварительно к конвертируемым в формат Base58Check данным добавляется префикс (байт версии), обеспечивающий категоризацию типов данных. Префикс версии в кодировке Base58Check позволяет создавать различные форматы, легко различимые визуально по определенному набору символов в начале строки. Например, для биткоин-адреса префикс равен 0 (0x00 в шестнадцатеричной системе счисления), для закрытого ключа - 128(0x80 в шестнадцатеричной системе счисления). Соответственно, в кодировке Base58Check запись биткоин-адреса будет начинаться с цифры 1, а закрытого ключа формата WIF - с цифры 5. Список основных префиксов версий типов данных приводится в таблице:

Таблица 2.1. Типы данных и префиксы, используемые в операциях кодирования Base58Check в платформе Биткоин

Категория данных	Префикс в шестнадцатеричной системе счисления	Результат кодирования с помощью Base58Check
Биткоин-адрес	0x00	1
Pay-to-Script-Hash адрес	0x05	3
Адрес для тестовой Биткоин сети	0x6F	m или n
Приватный ключ WIF	0x80	5, K или L
Зашифрованный приватный ключ	0x0142	6P
Расширенный публичный ключ	0x0488B21E	xpub

Суммируя вышеизложенное, сформулируем алгоритм работы системы кодирования Base58Check:

К кодируемым данным в соответствии с их типом добавляется специальный префикс (в начало числовой последовательности).

Вычисляется контрольная сумма путем двойного применения алгоритма хеширования к результату предыдущего этапа. Результатом хеширования является 32-х байтный дайджест, из которого мы выбираем только первые четыре байта. Именно эти 32 бита будут служить основанием

для выявления ошибок, т.е., контрольной суммой. Формально это можно записать следующим образом:

контрольная сумма = первые четыре байта (SHA-256(SHA-256(префикс + данные))), где + означает операцию слияния.

Затем контрольная сумма записывается в конец числовой последовательности.

Полученная последовательность (префикс + данные + контрольная сумма) кодируется с использованием системы Base58.

Данный процесс иллюстрируется схемой, изображенной на рисунке 14.

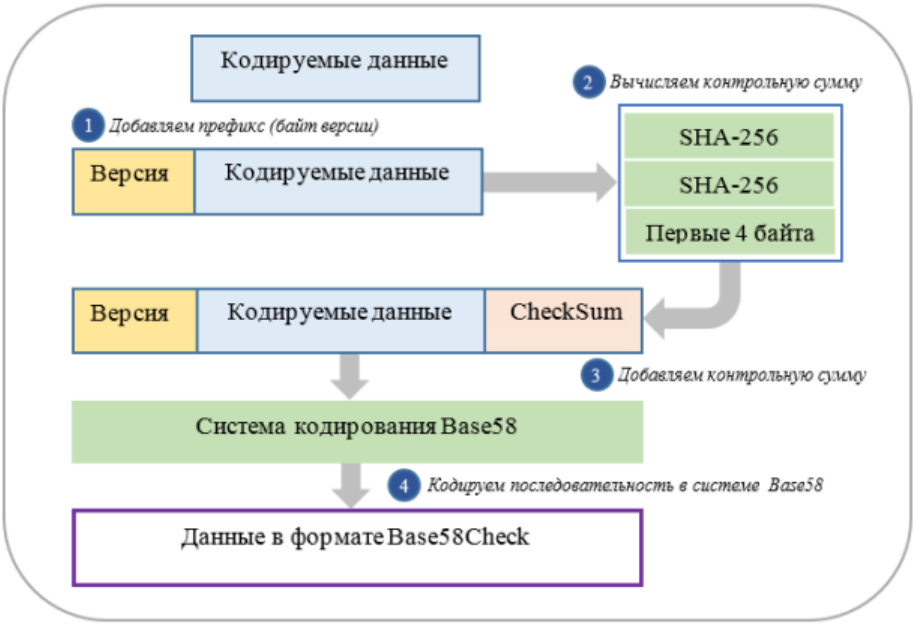

Рис. 2.8. Процесс кодирования данных в формат Base58Check

Большинство данных в сети Биткоин, с которыми работает пользователь, представлены в кодировке Base58Check. Такая форма записи удовлетворяет сразу трем немаловажным требованиям: компактность, удобство для восприятия и дополнительная защита от ошибок.

Пример формирования биткоин-адреса

Цель: Исследовать процесс получения биткоин-адреса на конкретном примере.

В качестве иллюстрации рассмотренных процедур приведем пример создания биткоин-адреса:

Начинаем с открытого ключа (65 байт, первый байт = 0x04, следующие 32 байта соответствуют координате X и заканчивается последовательность 32 байтами, соответствующими координате Y):

04678afdb0fe5548271967f1a67130b7105cd6a828e03909a67962e0ea1f61deb649f6bc3f4cef38c4f35504e51ec112de5c384df7ba0b8d578a4c702b6bf11d5f

Применяем к открытому ключу хеш-функцию SHA-256:

261c1eb21fc4708c6acbe1cfc6d4565652e9e768b620782898936b93000a6c02

Применяем к результату предыдущего шага хеш-функцию RIPEMD-160:

62e907b15cbf27d5425399ebf6f0fb50ebb88f18

Добавляем байт версии в начало результата предыдущего шага (0x00 биткоин-адрес основной сети):

0062e907b15cbf27d5425399ebf6f0fb50ebb88f18

Применяем к результату предыдущего шага хеш-функцию SHA-256:

9b90f16de7f0e580c07735dac15ffe23e2f8f8e103914e509aa91913ffdb9fb6

Применяем к результату предыдущего шага хеш-функцию SHA-256 (для нахождения контрольной суммы):

c29b7d937e3049e279391e62fdf00c12def7444013ddf6215808d10e9f2d5996

Находим контрольную сумму (первые 4 байта от дайджеста, полученного на предыдущем шаге):

c29b7d93

Добавляем контрольную сумму в конец числовой последовательности (расширенный дайджест хеш-функции RIPEMD-160), полученной в пункте 4. Результатом является 25-байтный двоичный биткоин-адрес:

0062e907b15cbf27d5425399ebf6f0fb50ebb88f18c29b7d93

Результат предыдущего пункта конвертируем в строку формата Base58. Получаем самый популярный формат записи биткоин-адреса (34 символа).

1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa

Сжатые открытые ключи

Цель: Сформировать четкое понимание процесса и развить практические навыки получения сжатых открытых ключей в сети Биткоин.

Оказывается, даже педантичным компьютерам и телекоммуникационным сетям, не путающимся в цифрах, не очень удобно работать с открытыми ключами в оригинальном виде. Правда, претензии у них несколько иного рода. Публичные ключи фигурируют в составе большинства транзакций. Это необходимо для проверки учетных данных владельца при совершении платежа. Каждый открытый ключ требует 520 битов памяти для хранения. Каждый день в сети Биткоин регистрируются десятки тысяч транзакций в день, существенно раздувая размеры блокчейна. Для распределенной системы, подразумевающей, что копия реестра должна храниться у каждого участника сети (пока мы не рассматриваем более простые вари анты с легкими кошельками) такая динамика становится очень накладной и грозит стать запредельной. Отметим, что на начало февраля 2019 года размер блокчейна платформы Биткоин превысил 236.68 GB. Вы сами можете оценить перспективы роста на основе графика динамики размера блокчейна сети Биткоин за 10 лет ее существования:

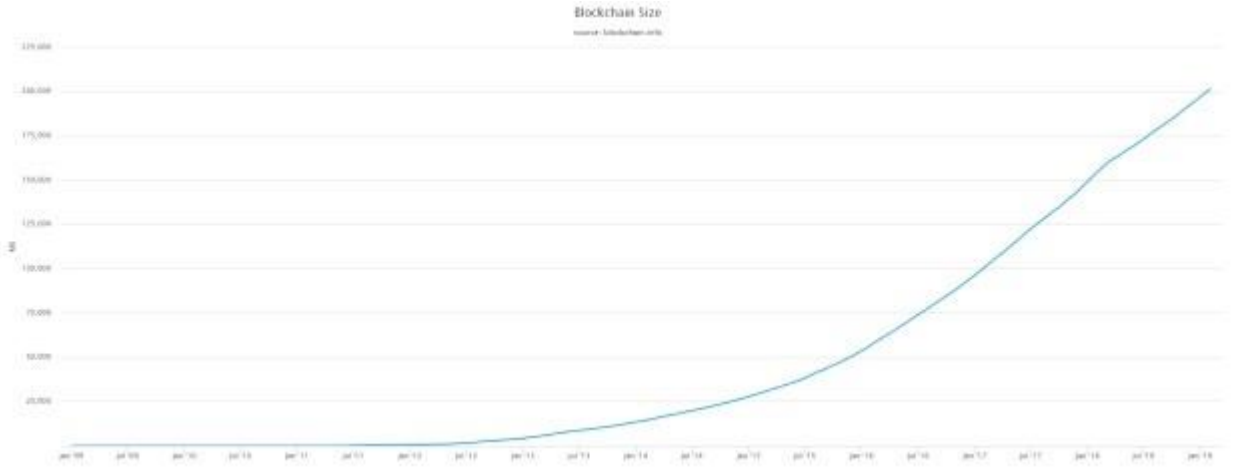

Рис. 2.9. Размер блокчейна платформы Биткоин

Сжатые открытые ключи были разработаны с целью уменьшения размера транзакций и сокращения размера блокчейна. Как мы уже знаем, открытый ключ — это точка на эллиптической кривой. Поскольку кривая является реализацией математической функции, точка на кривой представляет собой решение уравнения, а значит, если мы знаем x координату, то мы можем вычислить y координату. Для этого достаточно решить известное уравнение

$$y2 \bmod p = x3 + 7 \pmod p$$

Это соображение позволяет построить систему, в которой хранится только одна координата, входящая в открытый ключ. Экономия составляет ровно 256 бит, почти 50%. А вычислением недостающей координаты вполне может заняться программное обеспечение платформы Биткоин, например, же кошельки.

Чтобы не запутаться в таком многообразии ключей, разработчики снова прибегли к системе префиксов. Если несжатые открытые ключи предваряются префиксом 04, то сжатые начинаются с 02, или с 03. Выясним причину появления двух возможных префиксов. Проблема заключается в том, что, если точка лежит на эллиптической кривой, то для заданной X координаты очевидно существуют два решения уравнения (в формуле у нас фигурирует y2 и эллиптическая кривая симметрична относительно оси X). В данном случае можно воспользоваться следующим свойством функции, определенной над конечным полем: если для X координаты существуют решения уравнения, то одна из точек будет иметь четную Y координату, а вторая — нечетную. Для первого случая был зарезервирован префикс 0x02 (в шестнадцатеричной системе счисления), а для второго - 0x03 (в шестнадцатеричной системе счисления).

Небольшая иллюстрация на рисунке 16 позволяет визуализировать этот процесс.

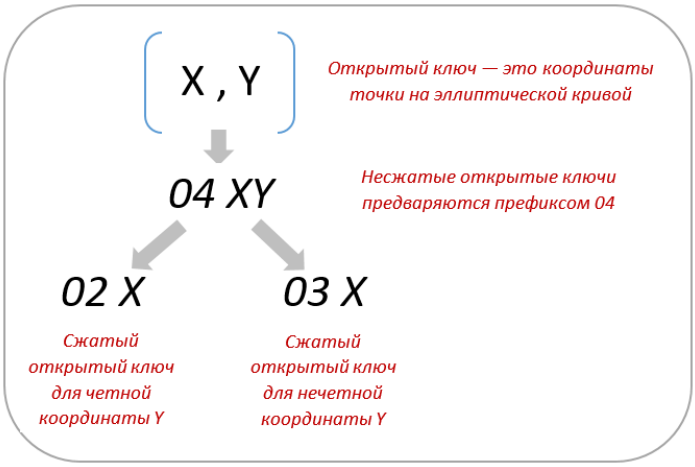

Рис. 2.10. Формирование сжатых открытых ключей

Приведем также пример в цифрах. Пусть у нас имеется открытый ключ, представленный двумя координатами (каждая представлена 256-битным числом или 64 шестнадцатеричными цифрами):

x = F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A

y = 07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB

Запишем его в несжатом виде (520-битное число или 130 шестнадцатеричных цифр с префиксом 0x04):

K =

04F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A07CF33DA18BD734C600B96A72BBC4749D5141C90EC8AC328AE52DDFE2E505BDB

А теперь представим его в сжатом виде (264-битное число или 66 шестнадцатеричных цифр с префиксом 0x03, указывающим на нечетность значения второй координаты):

K = 03F028892BAD7ED57D2FB57BF33081D5CFCF6F9ED3D3D7F159C2E2FFF579DC341A

Это сжатый открытый ключ точно также соответствует исходному закрытому ключу и может быть из него получен. При этом выглядит он иначе. Более того, если мы сконвертируем этот сжатый открытый ключ в биткоин-адрес, используя двойное хеширование (RIPEMD-160(SHA-256(K))), то получим другой биткоин-адрес. Это может привести к путанице, так как означает, что от одного и того же приватного ключа может быть получен публичный ключ, выраженный в двух различных форматах (сжатом и несжатом), которым соответствуют два разных адреса Биткоин. Тем не менее, приватный ключ является единым для обоих адресов.

Несжатые открытые ключи постепенно вытесняются в платформе Биткоин. Современное клиентское программное обеспечение по умолчанию оперирует именно сжатыми открытыми ключами, что, безусловно благотворно сказывается на размерах транзаккций и, как следствие, блокчейна. Тем не менее, пока еще не все клиентское программное обеспечение поддерживает сжатые открытые ключи. С другой стороны, новые клиенты, поддерживающие прогрессивную схему сжатия открытых ключей, одновременно должны уметь работать с транзакциями,

созданными в старой парадигме. Это особенно важно, когда приложение кошелька импортирует секретные ключи из другой программы, потому что новый кошелек должен просканировать блокчейн в поиске транзакций, соответствующих этим импортируемым ключам. В условиях неопределенности задачи поиска (Какие именно биткоин-адреса следует искать? На основе каких открытых ключей они были получены – сжатых или несжатых?) новое программное обеспечение кошельков позволяет пометить особым образом закрытые ключи, на основе которых были получены сжатые открытые ключи и, соответственно, сжатые биткоин-адреса.

Сжатые закрытые ключи

Цель: Сформировать четкое понимание процесса и развить практические навыки получения сжатых закрытых ключей в сети Биткоин.

На самом деле ни о каком сжатии в том смысле, который вкладывался в процесс сжатия открытых ключей, речь совершенно не идет. Более того, размер сжатого закрытого ключа окажется на один байт больше, чем размер обычного, несжатого, за счет дополнительного суффикса 0x01, добавленного с правого края числовой последовательности. Этот суффикс подчеркивает тот факт, что используется новая схема вычисления открытых ключей и для данного закрытого ключа могут формироваться только сжатые открытые ключи.

Особо отметим, что данные форматы ключей не взаимозаменяемы. В современных кошельках, которые используют сжатые открытые ключи, закрытые ключи будут экспортированы только сжатыми (с префиксом K или L). Если кошелек старого образца и не использует сжатые открытые ключи, закрытые ключи будут экспортированы в формате WIF (с префиксом 5). Актуальной остается задача просигнализировать импортирующему кошельку, что он должен просканировать блокчейн на предмет сжатых или несжатых публичных ключей и адресов.

Если программное обеспечение кошелька поддерживает сжатые публичные ключи, именно они будут использоваться во всех транзакциях. На их основе также вычисляются биткоин-адреса, которые, в свою очередь, используются в транзакциях. При экспорте приватных ключей из современного кошелька, поддерживающего сжатые публичные ключи, используется модифицированный Wallet Import Format с добавлением одного байта суффикса 0x01 к закрытому ключу. Такой закрытый ключ в кодировке Base58Check называется "сжатый WIF" и начинается с буквы K или L, а не с 5, как в случае с несжатыми ключами WIF формата из более старых кошельков. В таблице 3 приведены примеры записи одного и того же закрытого ключа в разных форматах.

Таблица 2.2. Варианты кодирования закрытого ключа

Формат	Закрытый ключ
Шестнадцатиричный	1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD

WIF	5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2JpbnkeyhfsYB1Jcn
Сжатый шестнадцатиричный	1E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD01
Сжатый WIF	KxFC1jmwwCoACiCAWZ3eXa96mBM6tb3TYzGmf6YwgdGWZgawvrtJ

Кошельки

Цель: Сформировать представление о различных типах кошельков, используемых в сети Биткоин.

Типы криптовалютных кошельков

Выбор кошелька для криптовалюты давно перестал быть вопросом исключительно безопасности или простоты. Использование разных кошельков для разных целей становится обычным делом, как и использование разных счетов при работе с фиатными деньгами. Разработчики программного обеспечения активно эксплуатируют эту концепцию и непрерывно предлагают новые программные средства, в том числе и вполне работоспособные. Описание этого вышедшего из берегов бурного потока разработок никак не укладывается в парадигму книги, поэтому информацию для выбора криптовалютного кошелька для собственных целей предлагаю почерпнуть в Интернете – благо ее там сейчас предостаточно. Нас же в первую очередь будут интересовать архитектурные аспекты их реализации.

В отличие от традиционных банковских счетов, работающих с фиатными деньгами, криптокошельки не хранят наличность в буквальном смысле. Они содержат открытые и секретные цифровые ключи, предоставляющие доступ к биткоин-адресам, и позволяют подписывать транзакции. При совершении платежных операций в блокчейне появляется запись о переходе прав на определенную сумму криптовалюты к новому владельцу, фактической передачи ценностей при этом не происходит.

По способу хранения наиболее критичной информации (конечно же речь идет о ключах) кошельки принято делить на "горячие" и "холодные". К первым относятся кошельки с постоянным доступом к интернету: онлайн-кошельки, мобильные приложения, программы для персональных компьютеров, имеющих постоянное подключение к сети Интернет. Безусловно, они очень удобны в использовании, обеспечивают синхронизацию блокчейна и готовы мгновенно обрабатывать транзакции. При этом они подвержены хакерским атакам и вирусам. И уж тем более не стоит серьезно относиться к кошелькам, предлагаемым криптовалютными биржами. Доверять секретные ключи посреднику - не самая лучшая стратегия.

В системах, реализующих принцип холодного хранения, критичные данные хранятся в оффлайн-режиме на физическом носителе, не имеющим прямого соединения с Интернетом. Разумеется, такой способ обеспечивают гораздо более высокую степень защиты криптовалютной

наличности. При этом в плане проведения транзакций они не столь удобны. Другой угрозой является отличная от нуля вероятность поломки спец.техники или ее потери. Но здесь есть традиционный и достаточно простой способ защиты – своевременное создание резервных копий критичных данных. Холодный способ хранения реализуется в бумажных кошельках (имеются даже зашифрованные бумажные кошельки), системах с дополнительным физическим носителем и специальных аппаратных кошельках (например, Trezor, Keepkey или Ledger Nano S).

В заключение приведем достаточно простой вариант классификации криптовалютных кошельков, учитывающий в первую очередь технологию изготовления:

Веб-кошельки (или онлайн кошельки).

Идеально подходят для новичков, не желающих самостоятельно администрировать собственное программное обеспечение, или обладателей "маломощной" вычислительной техники.

Популярные онлайн (браузерные) кошельки: Матби; Coinbase; Blockchain.info; Cryptopay; Xapo; Bitpay и др.

Достоинства:

Отсутствует необходимость в скачивании, хранении и последующей синхронизации блокчейна (очень серьезно экономит время и место на постоянных носителях);

Управлять Веб-кошельком можно с самых разных устройств. Единственное условие наличие свободного доступа к Интернету.

Дополнительные сервисы и профессиональная поддержка.

Недостатки:

Низкий уровень защиты (секретный ключ хранится на стороне, у какого-то дяди, пусть даже и очень авторитетного).

Мобильные кошельки (устанавливаются на смартфоны и т.п.).

Мобильные клиенты позволяют совершать платежи по схеме scan-and-pay. Нет необходимости прокатывать карту, набирать PIN-код или вносить данные вручную. Единственное, что нужно для приема платежа, это открыть QR-код в своем мобильном кошельке и показать его контрагенту, чтобы он просканировал код своим мобильным телефоном, или просто поднести телефоны друг к другу (если они поддерживают технологию NFC). Мобильные кошельки, чтобы не скачивать весь блокчейн, хранят только его облегченную версию. А при транзакциях обращаются к доверенным нодам. Делается это, чтобы сэкономить столь значимые для телефона ресурсы как память и трафик.

Популярные кошельки для мобильных устройств: Blockchain Mobile; Bitcoin Wallet; Electrum; Mycelium; Coin Pocket; Xapo и др.

Очевидное достоинство - удобство использования при оплате и постоянный доступ к кошельку.

В качестве недостатка отметим высокую вероятность кражи самого телефона или потери. Дискредитации ключей в этом случае может помешать высокая степень безопасности (двухуровневая аутентификация, смс-оповещения), которую обеспечивают мобильные криптовалютные кошельки.

Локальные кошельки (установленные на персональный компьютер или ноутбук).

Подразделятся на два вида. "Толстые" или "тяжелые" – такие кошельки загружают на компьютер весь блокчейн, возводя пользователя в ранг полноценного участника сети Биткоин и занимая при этом большой объем памяти на постоянном носителе (на начало февраля 2019 года – уже более 236 GB). В отличие от тяжелых криптокошельков "тонкие" или "легкие" не скачивают весь блокчейн на персональный компьютер, а обращаются к нему через API сторонних агрегаторов. Занимают значительно меньше места по сравнению с толстыми, несколько быстрее проводят транзакции, но из-за обращений к стороннему ресурсу уступают толстым в безопасности. Хотя приватные ключи в обоих случаях хранятся у пользователя.

Популярные локальные кошельки: Bitcoin Core; Electrum; Jaxx; Exodus и др.

Обеспечивают достаточно высокий уровень надежности кошелька и защищенности средств. Но при этом могут занимать очень много места на жестком диске и долго синхронизируются с блокчейном.

Аппаратные кошельки (в виде отдельного устройства).

При инициализации таких устройств вводится код, ограничивающий доступ к аппаратному кошельку в случае его потери или кражи. Взломать данное устройство невозможно. Все транзакции совершаются в защищенной среде прибора. Даже, если подключить аппаратный кошелек к взломанному персональному компьютеру, злоумышленникам не удастся получить контроль над кошельком.

Единственным недостатком является довольно высокая цена (порядка $100).

Бумажные кошельки.

Это обычный листок бумаги, на которой распечатаны приватные и открытые ключи в виде QR-кода. Для практического использования этих ключей их следует отсканировать. Безусловно, в плане удобства уступают аппаратным кошелькам, но тем не менее обеспечивают сравнимый с ними высокий уровень безопасности.

Рис. 2.10. Бумажные и аппаратные криптовалютные кошельки

Детерминированные и недетерминированные (случайные) кошельки

Рассмотрим кошелек, с точки зрения организации процедуры хранения закрытых ключей. Недетерминированные кошельки реализуются в форме структурированных файлов или простейших баз данных. В ответ на желание владельца независимым (случайным) образом программным обеспечением кошелька генерируется очередной закрытый ключ, помещаемый в хранилище. Например, клиент Bitcoin Core в момент инициализации генерирует сразу 100 закрытых ключей. Разумеется, в случае необходимости формируются дополнительные ключи. Такая стратегия создает препятствия для операций резервного копирования, импорта/экспорта. Необходимо хранить довольно большой набор несвязанных случайных данных. Массовый выпуск приватных ключей даже не приводит к повышению конфиденциальности, поскольку появляются основания для отслеживания связи между различными адресами и транзакциями. Поэтому разработчики, сохраняя в своих клиентах возможность использования недетерминированного механизма, настоятельно рекомендуют использовать детерминированный способ генерации ключей.

В этом случае каждый новый закрытый ключ формируется путем применения односторонней хеш-функции к значению последнего, ранее сгенерированного закрытого ключа. В результате образуется цепочка секретных данных (вполне в духе технологии блокчейн). Для того чтобы воссоздать всю последовательность, достаточно знать только первый ключ (называемый зерном или мастер-ключом).

На практике применяются довольно много различных методов генерации ключей и структур кошелька, использующих эту идею. Мы остановимся только на двух технических решениях.

Детерминированные кошельки (с зерном)

Что именно должно представлять собой зерно? Зерно — это случайное число, которое в сочетании с другими параметрами, такими как номер индекса или "код цепи" позволяет получить всю цепочку приватных ключей. При такой стратегии для восстановления всех закрытых ключей достаточно знать только зерно. Следовательно, достаточно единожды сформировать его резервную

копию в момент создания, а не дублировать данные каждый раз, когда генерируется новый ключ, как это имело место в случае недетерминированного кошелька. Аналогичным образом упрощаются операции импорта/экспорта. Достаточно оперировать одним зерном, что кардинально снижает затраты и риски в процессах миграции ключей между различными реализациями кошельков.

Мнемонические кодовые слова

Компьютер замечательно запоминает числа, а вот человек лучше оперирует словами, еще лучше, если эти слова наделены смыслом. Именно на этом простом факте основан один из подходов к формированию зерна. Мнемонические коды — это некоторая строка, состоящая из английских слов, которая уникальным образом представляет (по сути - кодируют) случайное число, которое может быть использовано в качестве зерна для детерминированного кошелька. Этой строки вполне достаточно для того, чтобы заново создать зерно, а, используя его, воссоздать кошелек полностью, включая все производные ключи.

Работает это следующим образом. Приложение детерминированного кошелька с мнемоническим кодом в момент инициализации представит пользователю последовательность слов (числом от 12 до 24). Эта строка является резервной копией кошелька и может быть использована для восстановления и повторного создания всех ключей в таком же или любом другом совместимом клиенте.

Теоретически мнемоническую фразу можно придумать самому, но это небезопасно, так как человек плохо справляется со случайной генерацией. Лучший способ обезопасить свой кошелек – взять мнемоническую фразу, созданную с помощью специального генератора фраз.

Опишем процедуру создания мнемонического кода и зерна:

Создаем случайную числовую последовательность (энтропию) (от 128 до 256 бит).

Применяем к числу, полученному на первом шаге, хеш-функцию SHA-256. Выбираем первые несколько битов из вычисленного дайджеста. Будем называть их контрольной суммой.

Добавляем контрольную сумму в конец случайной последовательности.

Разделяем результат предыдущего шага на части длиной в 11 бит, и используем их в качестве индекса по словарю из 2048 заранее определенных слов.

Выбираем по индексам от 12 до 24 слов из словаря. Полученная фраза и будет представлять мнемонический код.

Англоязычный список слов для стандарта BIP39 (Mnemonic code for generating deterministic keys) содержит 2048 слов. Если фраза состоит из 12 слов, тогда число возможных комбинаций составляет 2048 в 12-й степени, или 2 в 132-й степени, то есть фраза обеспечит 132 бита безопасности. Для фразы из 24 слов энтропия возрастет до 256 бит.

Итак, мнемонический код представляет собой последовательность от 128 до 256 бит, которая впоследствии используется для получения более длинного (512 бит) зерна путем применения функции удлинения ключа PBKDF2 (Password-Based Key Derivation Function - стандарт

формирования ключа на основе пароля). Полученное зерно готово для создания детерминированного кошелька и соответствующих ключей.

В таблице 4 представлены примеры некоторых мнемонических кодов и зерен, которые они производят

Таблица 2.3. Примеры мнемонического кода и результирующего зерна

Энтропия на входе	Мнемонический код	Зерно (512 бит)
0c1e24e5917779d297e14d45f14e1a1a(128 бит)	army van defense carry jealous true garbage claim echo media make crunch (12 слов)	3338a6d2ee71c7f28eb5b882159634cd46a8984 63e9d2d0980f8e80dfbba5b0fa0291e5fb888a599 b44b93187be6ee3ab5fd3ead7dd646341b2cdb8d 08d13bf7
2041546864449caff939d32d574753fe684d3c947c3346713dd8423e74abcf8c (256 бит)	cake apple borrow silk endorse fitness top denial coil riot stay wolf luggage oxygen faint major edit measure invite love trap field dilemma oblige (24 слова)	3972e432e99040f75ebe13a660110c3e29d131a2c 808c7ee5f1631d0a977fcf473bee22fce540af281bf 7cdeade0dd2c1c795bd02f1e4049e205a0158906 c343

Иерархические детерминированные кошельки

Наиболее эффективной формой детерминированных кошельков является иерархический детерминистический кошелек или HD-кошелек, описанный в стандарте BIP0032 (Hierarchical Deterministic Wallets). Иерархические детерминированные кошельки содержат ключи в виде древовидной структуры, так что из родительского ключа можно вывести последовательность производных ключей, от каждого из которых, в свою очередь, также получается последовательность производных ключей и так далее без ограничения глубины вложений. HD-кошельки предлагают два основных преимущества по сравнению с недетерминированными. Во-первых, структура дерева может быть послужить в целях дополнительной организации, например, когда одна ветвь ключей используется для получения входящих платежей, а другая ветвь определена для получения сдачи для исходящих платежей. Ветвление ключей также может быть использовано в корпоративной среде для различных отделов, филиалов, конкретных функций, или категорий бухгалтерского учета.

Следующее преимущество HD-кошельков состоит в том, что их пользователи могут создавать последовательности открытых ключей, не имея доступа к соответствующим закрытым ключам. Это обстоятельство позволяет использовать HD-кошельки на небезопасных серверах или в режиме только приема средств с выдачей нового публичного ключа для каждой новой транзакции. Открытые ключи не должны быть предварительно загружены или получены заранее и сервер не содержит закрытые ключи, которые могут использоваться для доступа к средствам.

Шифрованные закрытые ключи

Как это часто бывает в практике обеспечения информационной безопасности, два ее важнейших аспекта – доступность и конфиденциальность - в отношении хранения закрытых ключей предъявляют противоречивые требования. Шифрование могло бы кардинально повысить секретность закрытых ключей. Однако, безсистемное применение этого мощного инструмента существенно осложняет выполнение многих необходимых в платежных системах операций, таких как создание резервных копий, миграция ключей из одного кошелька в другой, обновление программного обеспечения клиентов и т.д.

Эти соображения побудили IT-сообщество, участвующее в развитии и совершенствовании платформы Биткоин, разработать общий стандарт шифрования закрытых ключей - Предложения по улучшению Биткоина BIP0038. Предлагаемый универсальный алгоритм одновременно является простым, удобным и эффективным. Для шифрования закрытых ключей используется кодовая фраза, результат кодируется с помощью Base58Check. Таким образом обеспечивается надежность хранения секретной информации на резервных носителях, адекватность алгоритмов передачи ключей между кошельками, не столь критичными становятся ситуации, в которых ключи могут быть обнаружены. Стандартом для шифрования был выбран Advanced Encryption Standard (AES) - симметричный алгоритм блочного шифрования, принятый в National Institute of Standards and Technology (NIST).

Криптосистема, рекомендованная BIP0038, принимает на входе приватный ключ, обычно кодируемый в WIF-формате (Base58Check-строка с префиксом 5). Параллельно с этим вводится ключевая фраза - длинный пароль, обычно состоящий из нескольких слов или представляющий собой последовательность алфавитно-цифровых символов. Результатом работы криптосистемы является приватный ключ в кодировке Base58Check с префиксом 6P. Всегда, когда Вам попадается закрытый ключ, начинающийся с 6P, это означает, что он зашифрован и должен быть преобразован обратно в формат WIF (префикс 5). Для дешифрования потребуется упомянутая выше секретная фраза. Программное обеспечение большинства современных кошельков поддерживает стандарт BIP0038.

С зашифрованными закрытыми ключами работают также сторонние приложения, например, https://www.bitaddress.org

Очень часто приведенная технология шифрования ключей используется в бумажных кошельках. Владельцу достаточно выбрать сильную секретную фразу, чтобы обеспечить бумажному кошельку с зашифрованными (по схеме BIP0038) закрытыми ключами обеспечивает чрезвычайно высокий уровень безопасности и является весьма эффективным способом холодного хранения.

Таблица 2.4. Пример зашифрованного закрытого ключа

Закрытый ключ (WIF)	5J3mBbAH58CpQ3Y5RNJpUKPE62SQ5tfcvU2Jpbnk
Секретная фраза:	MyTestPassphrase
Зашифрованный закрытый ключ	6PRTHL6mWa48xSopbU1cKrVjpKbBZxcLRRCdctLJ3z5yxE87MobKoXdTsJ

Pay-to-Script Hash (P2SH) адреса и мульти-подписи

Цель: Сформировать начальное представление о P2SH-транзакциях и мультиподписных платежах, акцентируясь на особенностях, представляющих их биткоин-адресах.

Pay-to-Script Hash адреса

Ранее рассматривалась стандартная процедура формирования адреса по схеме закрытый ключ -> открытый ключ -> биткоин-адрес. Такие адреса начинаются с цифры "1". Каждый участник сети Биткоин может оправить средства на такой адрес. А вот для того чтобы их потратить, необходимо обладать соответствующим закрытым ключом. Тут, как говорится, без вариантов. Способ P2PKH (Pay-to-Public-Key-Hash) являются основной формой реализации транзакций в сети Биткоин. Такие транзакции "блокируют биткоины" на своих выходах. Впоследствии средства могут быть разблокированы и тут же снова заблокированы последующими транзакциями. Подобным образом биткоины легко мигрируют с одного адреса на другой. Транзакции, отправляющие средства на биткоин-адрес, содержат скрипты P2PKH, разрешающиеся с помощью открытого ключа и электронной подписи, созданной на основе соответствующего закрытого ключа. В случае стандартной транзакции скрипт включается в ее выход.

Большинство нестандартных платежей, например, мульти-подписи или проверка условия, что определенный выход транзакции не может быть потрачен до определенного момента времени в будущем (CheckLockTimeVerify), используют более сложные схемы, с общим названием P2SH (Pay to Script Hash - "оплата по хешу скрипта"). Такие биткоин- адреса, начинаются с цифры "3".

P2SH транзакции были введены Гэвином Андресеном в январе 2012 года. Цель нововведения по словам Андресена состояла в том, чтобы "перенести ответственность за предоставление условий по выкупу транзакции с отправителя средств на того, кто данные средства тратит".

При использовании способа оплаты P2SH биткоины также блокируются в скрипте. Однако сам скрипт не включается в выход транзакции. Вместо этого сценарий разблокировки хешируется,

а результат этот операции записывается в выход транзакции. Как и принято в криптографии, полученный дайджест нельзя использовать для восстановления оригинального скрипта. Для хеширования pay-to-script используется тот же алгоритм, который используется при создании биткоин-адреса (двойное хеширование), только применяется он к сценарию, а не к открытому ключу. Полученный дайджест кодируется при помощи Base58Check с версией префикса 0x05, что дает закодированный адрес, начинающийся с 0x03.

Для разблокировки выхода P2SH в последующей транзакции, недостаточно выполнения сформулированных в скрипте условий. Нодам в сети Биткоин вообще доступен только хеш скрипта, а не сам сценарий. Поэтому, никакая нода не может подтвердить выполнение предусмотренных в скрипте условий. Кроме той, которой известен весь сценарий блокировки. Именно поэтому следующая транзакция, расходующая биткоины, должна включать в себя скрипт целиком, а также сформулированные в нем условия (блокировка (скрипт) + ключ (условия) для разблокировки средств).

Иначе говоря, начиная с 2012 года при использовании решения P2SH только новый держатель средств знает, как можно их потратить, остальные участники сети не знают условий блокировки средств (скриптов), вплоть до их разблокирования - траты. Достигается это путем включения в блокчейн только хеш-дайджестов исходных скриптов. Именно хеши регулируют права собственности на наличность. В момент траты владелец предоставляет исходный текст скрипта и ключ для расшифровки хеш-дайджеста одновременно. После этого каждый пользователь сети Биткоин может проверить истинность скрипта и выполнение условий траты на основе начального хеша.

Как же будет организована трата средств, заблокированных P2SH-транзакцией? В разблокирующей транзакции приводится оригинальный скрипт (скрипт погашения). Затем майнеры, хешируя фактический скрипт, смогут подтвердить, что представленный скрипт соответствует хешу, включенному в заблокированный выход. Таким образом транзакция может быть подтверждена и включена в блок, а затем и в блокчейн.

Способы оплаты P2SH чаще всего представляют сценарии мульти-подписи, но они также могут представлять сценарии, реализующие другие типы транзакций.

У P2SH-транзакций существует определенный набор преимуществ перед аналогами:

Отправитель может финансировать любой произвольный сценарий погашения, не зная, каковы эти условия расходов. Это имеет смысл, если отправителю во многом все равно, как будут потрачены эти средства в будущем – это вопрос для получателя, которому небезразличны условия дальнейшего расходования средств. В случае транзакций с мультиподписью, отправитель может отправить средства, не зная необходимые открытые ключи, принадлежащие получателю, адреса получателя, которые раскрываются только тогда, когда получатель тратит средства. Подобный вид сделки повышает безопасность получателя.

Отправитель может использовать короткий 34-символьный адрес, подобный указанному выше, вместо длинного и громоздкого адреса, содержащего сведения о сценарии полного погашения. Это позволяет получателю разместить короткий адрес на платежной странице или в сообщении, что снижает вероятность человеческих ошибок в транскрипции.

Такой способ снижает операционные сборы для отправителя средств. Комиссии за транзакцию пропорциональны размеру транзакции, а дайджест фиксированной длины позволяет отправителю перечислять средства на любой произвольный скрипт погашения, не беспокоясь о выплате более высоких сборов. Это ответственность получателя, который создает скрипт погашения, чтобы определить, насколько велика будет их транзакция расходов и сколько это будет стоить. На данный момент это не является большой проблемой в связи с тем, что транзакционные издержки довольно малы. Однако они могут быть более важными в будущем, поскольку вознаграждение за блок в сети Биткоин постоянно уменьшается.

Обычно реализация функции P2SH является мульти-подписью адреса скрипта. В данном сценарии необходимо, чтобы выход транзакции разблокировался более чем одной электронной подписью, чтобы доказать право собственности и, следовательно, потратить средства.

Мульти-подписи

В настоящее время наиболее распространенной реализация функции P2SH — это сценарий мульти-подписного адреса. Как следует из названия, основной сценарий требует более одной подписи для доказательства права собственности и, следовательно, возможности распоряжения средствами.

Таким образом, мульти-подпись M-из-N, реализованная в сети Биткоин, означает, что для разблокирования выхода транзакции достаточно получить M подписей (это число также называют "порогом") из N возможных. При этом, $M \leq N$.

К такому биткоин-адресу привязано сразу несколько пар ECDSA ключей. Каждая пара состоит из закрытого и открытого ключей. Схемы комбинаций, согласно которым можно использовать эти ключи, могут быть различными. Более того, можно установить условия, при которых нужно будет предоставить несколько подписей, чтобы потратить средства, заблокированные в выходе транзакции.

Существуют различные комбинации ключей при использовании мульти-подписи. Самыми популярными вариантами являются 2-из-2, 2-из-3, а также 3-из-3. Максимально возможный вариант — 15-из-15.

Технические аспекты реализации мульти-подписи рассмотрим в разделе, посвященном транзакциям.

Краткие итоги

Право владения любыми цифровыми активами устанавливается через криптографические ключи, биткоин-адреса и электронные подписи. Закрытые криптографические ключи не

перемещаются по сети. Они генерируются и хранятся пользователями в специализированном клиенте (кошельке). Цифровые ключи в кошельке абсолютно независимы от протокола Биткоин. Благодаря такой стратегии управления ключами становятся возможными многие важнейшие свойства сети Биткоин, включая децентрализованные консенсус и контроль, подтверждение владения и модель безопасности, основанную на математическом (криптографическом) доказательстве.

В большинстве случаев биткоин-адрес формируется на основе публичного ключа. Однако протоколом Биткоин разрешаются адреса, реализуемые по иным схемам.

В сети Биткоин для создания криптопары, контролирующей доступ к счету, используется умножение на эллиптических кривых. Согласно протоколу Биткоин закрытый ключ (число длиной в 256 бит) генерируется случайным образом, открытый ключ (два 256-битных числа + 8 бит префикса типа) создается путем криптографического преобразования закрытого ключа с использованием алгоритма эллиптической криптографии secp256k1 – частного случая алгоритма ECDSA. Биткоин-адрес (число длиной в 200 бит) – результат последовательного двукратного вычисления хеш-функций (сначала SHA-256, а затем RIPEMD-160) публичного ключа и добавления к результату префикса типа (1 байт) и контрольной суммы (4 байта).

Для записи биткоин-адреса используется кодировка Base58Check, мощность алфавита которой составляет 58 символов.

Использование сжатых открытых ключей позволяет существенно сократить нагрузку на блокчейн за счет уменьшения размера транзакции.

Сжатые открытые ключи лишь подчеркивают факт использования программным обеспечением новой схемы вычисления открытых ключей исключительно в сжатом виде.

По способу хранения ключей кошельки принято делить на "горячие" и "холодные". Холодный способ хранения реализуется в бумажных кошельках (имеются даже зашифрованные бумажные кошельки), системах с дополнительным физическим носителем и специальных аппаратных кошельках (например, Trezor, Keepkey или Ledger Nano S).

Большинство нестандартных платежей, например, мульти-подписи или проверка условия, что определенный выход транзакции не может быть потрачен до определенного момента времени в будущем (CheckLockTimeVerify), используют схемы с общим названием P2SH. Такие биткоин-адреса, начинаются с цифры "3".

Лекция 3: Транзакция – это запись в распределенном реестре

Все представляют, что такое Википедия? А теперь вообразите эту электронную энциклопедию без информационных статей. Вообще без единой. Будет ли от такого ресурса хоть какая-нибудь польза? Конечно нет. Подобную параллель можно провести между блокчейном и учтенными в нем транзакциями. Концептуально блокчейн – это организационная структура,

обеспечивающая удобный, прозрачный и безопасный способ реализации жизненного цикла транзакций. Правда, способ настолько оригинальный и неожиданный, что вызвал целую технологическую революцию и, вообще, обещает перевернуть весь мир. Тем не менее, основную ценность представляет контент, т.е., транзакции. Именно в них отражается движение денежных средств. Поэтому блокчейн Биткоина часто называют глобальной бухгалтерской книгой. Итак, мы выяснили, что транзакции являются наиболее важной частью системы Биткоин. Остальные элементы платформы выполняют инфраструктурные функции и задачи. Достаточный повод, чтобы серьезно во всем этом разобраться.

Транзакции - это специальные структуры данных, фиксирующие процессы передачи ценности (токенов) между участниками криптоплатформы.

Жизненный цикл транзакции представлен на рисунке 3.1.

Рис. 3.1. Жизненный цикл транзакции

Подписание транзакции валидной электронной подписью – обязательный шаг, снимающий блокировку средств, замороженных где-то в глубинах блокчейна. Можно сказать, что механизм распределенных реестров является вдвойне прозрачным. По распределенному реестру каждый пользователь сети всегда может отследить любую цепочку транзакций, фиксирующих движение конкретных электронных монет. Но кроме записей, уже внесенных в блокчейн, клиенты платформы имеют возможность анализировать транзакции (сделки), которые еще даже не включены в блоки. Практически любой узел в сети через какое-то время получает все подписанные, но еще не оприходованные транзакции. Ноды проверяют полученные транзакции и передают их дальше. Примерно так работает пиринговая сеть Биткоина. На плечи майнеров ложится очень нелегкая задача реализации принятого в платформе Биткоин механизма поддержания консенсуса. Настолько нелегкая, что в настоящее время майнеров-одиночек практически не осталось – хешрейт не позволяет. Итогом майнинга является включение очередного блока, содержащего среди прочих интересующую нас транзакцию, в блокчейн. На этом активная фаза жизненного цикла транзакции завершается. Новый транзакционный цикл начнется в тот момент, когда владелец полученных средств решит ими воспользоваться, удлиняя тем самым цепочку владения, отраженную в реестре.

Когда речь идет о крупных сделках рекомендуется несколько отложить момент ее завершения. Следует дождаться включения в блокчейн нескольких блоков дополнительно. Добавление очередного блока называется подтверждением. В настоящее время считается вполне

достаточно шести подтверждений, т.е., ждать придется недолго – около часа. Программное обеспечение большинства кошельков настроено именно таким образом и будет отображать транзакции как неподтвержденные вплоть до момента получения шести подтверждений.

Создание транзакций

Есть пара аспектов, которые роднит транзакции в сети Биткоин и традиционные чеки. На самом деле не важно кто создает эти финансовые документы, выражающие намерение передать определенные ценности. В платежной системе они никак не проявляются до тех пор, пока не будут предъявлены для исполнения. А для этого они должны быть подписаны непосредственно владельцем средств. Так же как чек указывает некий счет в качестве источника средств, транзакция ссылается на одну или несколько транзакций, записанных в блокчейне, и черпает ценности с их выходов.

Должным образом сформированная и подписанная, транзакция содержит всю информацию, необходимую для перевода денежных средств. Такая транзакция программным обеспечением кошелька отправляется в путешествие по сети Биткоин, чтобы майнеры могли ее проверить и включить в свои блоки.

Трансляция транзакций по сети Биткоин

Среднестатистический размер транзакции составляет от 300 до 400 байт. Поскольку каждый узел в сети Биткоин связан с несколькими соседними узлами, проблема доверия между ними не возникает (пока мы не принимаем в расчет возможность атаки Сивиллы (Sybil attack) — вид атаки, характерной для одноранговой сети, в результате которой жертва ограничена коммуникациями с узлами, контролируемыми злоумышленником). Аутентификация отправителей транзакций получающими их нодами также не предусмотрена. Напомним, что пока расчеты осуществляются в рамках сети Биткоин поддерживается анонимность их участников.

Чуть позже мы убедимся в том, что транзакции не содержат секретной информации (закрытые ключи, учетные данные). Благодаря этому для их трансляции можно без опасений использовать любой удобный сетевой транспорт, чего ни в коем случае нельзя делать, например, в платежных системах на основе кредитных карт, содержащих конфиденциальную информацию и использующих для коммуникаций исключительно шифрованные каналы. Для отправки транзакции подходят самые простые и незащищенные средства, включая Wi-Fi, Bluetooth, NFC, Chirp, штрих-коды, копирование и вставку в веб-формах. Могут использоваться каналы спутниковой или коротковолновой радиосвязи и т.д.

Каждый узел сети, получив транзакцию, осуществляет ее проверку. Такой подход позволяет предотвратить распространение спама, атаки на отказ в обслуживании (DoS или DDoS-атаки), уменьшить другие угрозы системе. Если транзакция оказывается валидной, узел ретранслирует ее другим, известным ему нодам. Соответствующее сообщение получит также непосредственный отправитель транзакции. Таким образом, в течение непродолжительного времени (речь идет о

секундах) валидная транзакция распространится в виде круговой волны (подобно волне на поверхности воды от брошенного камня) со скоростью, растущей в геометрической прогрессии.

Если результат проверки окажется отрицательным, узел отклоняет транзакцию и извещает об этом отправителя.

Сеть Биткоин является пиринговой. Т.е., любой узел связан с несколькими другими узлами, обнаруженными им непосредственно после запуска при помощи особого децентрализованного протокола. Транспортная система Биткоин образует слабо связанную сеть без фиксированной топологии, или какой-либо структуры, т.е., является одноранговой.

Являясь неотъемлемым инфраструктурным компонентом платформы Биткоин, одноранговые (пиринговые, P2P - от английского "peer-to-peer") сети заслуживают определенного внимания.

Основная цель пиринговых сетей – организация эффективного обмена файлами практически любого размера. Однако, известны и другие сферы использования концепции одноранговых систем, например, в области распределенных вычислений – технологии, позволяющей задействовать удаленные вычислительные системы для решения сложных, ресурсоемких задач. Еще одно название подобных сетевых объединений -децентрализованные. Для наших целей достаточно рассмотреть основные особенности технических решений на основе пиринговых сетей для построения сетевых систем файлообмена.

Одноранговый характер сети свидетельствует о том, что все ее участники имеют одинаковые права и выполняют одни и те же функции, то есть, могут, как принимать информацию, выступая в роли сервера, так ее и отдавать, выступая в роли клиента. Здесь нет выделенных серверов, предоставляющих в общее пользование свои ресурсы или информационные сервисы. Недаром сочетание "peer-to-peer" можно перевести как "равный к равному". Еще такие сетевые объединения называют децентрализованными.

Как работают пиринговые сети?

Как и любая другая сеть, пиринговая представляет собой свободное объединение компьютеров. Отличительной особенностью является логика организации его работы, основанная исключительно на равноправии всех участников. Кстати, принято участников такой сети называть пирами. Мы все очень любим глобальную сеть Интернет, ориентированную на клиент-серверную архитектуру. Что же такого полезного есть в одноранговой сети и что не может обеспечить Интернет? Прежде всего, потрясающая работоспособность, демонстрируемая независимо от числа доступных пиров, при любом возможном их сочетании. В одноранговой системе пропускная способность серверов не является критичным местом (типичная уязвимость обычных сетей). Их там просто нет.

Когда мы используем клиент-серверную модель, качество информационного сервиса (скорость выдачи информации, время обслуживания) полностью зависит от возможностей

соответствующего сервера. Если сервер выходит из строя, информация или оказываемые им услуги становятся полностью недоступными.

Пиринговая модель передачи данных позволяет выстроить иную логику работы. Информация реплицируется на множестве узлов сети. Мало того, что пропадает зависимость от ограниченного в ресурсах единственного сервера, способного удовлетворить Ваши информационные потребности, Вы еще и получаете принципиальную возможность получать необходимую информацию одновременно из разных источников. Теперь скорость получения информации будет определяться только Вашей готовностью ее принимать (пропускной способностью Вашего канала). Надежность такой системы безусловно выше чем в централизованном решении.

Конечно мы нарисовали слегка идеальную картину пиринговой сети. Практические реализации одноранговых файлообменных систем сталкиваются с рядом трудностей, главной из которых является поиск таких пользователей, которые обладают интересующим Вас файлом и одновременно находятся в активном состоянии. На практике подобные проблемы решаются с помощью гибридных систем, допускающих присутствие выделенных серверов, координирующих работу, процессы поиска зарегистрированных в сети компьютеров, определения их текущего статуса ("активен" или "не активен").

Доступ к сервисам одноранговых сетей осуществляется с помощью соответствующего клиента.

И в заключении небольшого экскурса в мир пиринговых сетей приведем список наиболее известных файлообменных P2P-сетей:

ED2K (eDonkey2000). Работает по протоколу MFTP. В качестве клиента используется приложение eMule (Edonkey – его устаревшая версия). Сеть продолжает действовать и в настоящее время, хотя разработчики прекратили его поддержку еще в 2005 году. Этот факт лишний раз подчеркивает потрясающую живучесть такого рода систем.

BitTorrent – самая популярный файлообменник, обеспечивающий высокую скорость передачи данных. Популярные клиенты: uTorrent, BitComet, BitSpirit, Azureusи т.д.

Direct Connect – представляет собой набор связанных между собой небольших хабов (серверов), предназначенных для поиска информации на компьютерах участников этих сетей. Информацию об активных хабах концентрируется на специальных серверах (хаблистах). Яркий пример частично децентрализованной сети. Может использоваться для организации файлового обмена в крупных районных или городских локальных сетях. Основной клиент - DC++.

Gnutella и Gnutella2 – одноранговые сети в чистом виде, использующие для передачи данных свой собственный протокол, разработанный фирмой Nullsoft. Основные клиенты: Shareaza, LimeWire, Phex, Morpheusи т.д.

FastTrack. Использует классическую версию протокола P2P, правда в передаче информации участвуют только те источники, которые имеют полные версии файлов. Основные клиенты – KaZaA, giFT(KCeasy) и mlDonkey.

Структура транзакции

Цель: Сформировать комплекс знаний о структуре транзакции, составе и взаимосвязи ее компонентов, процедуре фиксации факта передачи цифрового актива от одного владельца другому с помощью записи в реестре.

Разберемся каким образом в транзакции записывается инструкция передать средства из конкретного источника (определяется входом транзакции) заданному получателю (определяется выходом транзакции).

Как уже упоминалось ранее, блокчейн сети Биткоин не содержит балансовые счета или данные их владельцев. В связи с этим входы и выходы транзакций проще ассоциировать с некоторым числом токенов, заблокированных посредством криптографических операций. Система организована так, что только владелец закрытого ключа способен их разблокировать.

Наконец-то заглянем внутрь транзакции (Таблица 3.1).

Таблица 3.1. Структура транзакции

Размер	Поле	Описание
4 байта	Версия	Описывает правила управления транзакцией
1-9 байт	Число входов	Показывает число входов транзакции
Без ограничения	Входы	Структура, включающая один или несколько входов транзакции
1-9 байт	Число выходов	Показывает число выходов транзакции
Без ограничения	Выходы	Структура, включающая один или несколько выходов транзакции
4 байта	Locktime	Время (в формате ОС Unix) или номер блока

Locktime

Сначала разберемся с полем Locktime (Время связывания транзакции). Очень похоже на функционал обычного чека с отсрочкой платежа. Трактуется в платформе как временная точка, до наступления которой транзакция не может стать действительной, распространяться по сети или включаться в блоки.

В большинстве транзакций поле Locktime устанавливается в 0, что означает немедленное распространение и исполнение. Если его содержимое отлично от 0 и меньше 500 000 000, то его значение интерпретируется как высота блока, до достижения которой транзакция не считается действительной, не распространяется по сети и не включается в блокчейн. Если же оно больше 500 000 000, система интерпретирует его как Unix-время (количество секунд, прошедших с 1 января 1970-го года), до наступления которого транзакция не считается действительной.

Входы и выходы транзакции

Цель: Сформировать комплекс знаний о строении входов и выходов транзакции, составе, функциях и взаимосвязи их структурных компонентов.

Безусловно, это самые интересные и важные части транзакции. Концептуальным элементом схемы являются неизрасходованные выходы транзакций (Unspent Transaction Output) или UTXO.

Биткоин был первой валютой, реализующей модель UTXO для отслеживания состояния реестра. Каждая очередная транзакция расходует выходы предыдущих и создает новые выходы, которые в свою очередь будут израсходованы последующими транзакциями. Каждый выход может использоваться только один раз. Такая структура обладает множеством очень полезных математических свойств, включая конструктивное доказательство невозможности двойных трат при условии, что каждая транзакция доказывает факт того, что сумма ее входов больше, чем сумма ее выходов.

Основная причина успеха UTXO - это присущий ей естественный параллелизм в виду того, что каждая транзакция может обрабатываться параллельно, так как все они относятся к независимым неконфликтующим выходам (поскольку у каждого выхода ровно один владелец). С точки зрения теоретической информатики, UTXO – очень элегантная и легко доказуемая схема.

Отметим, что модель UTXO подвергается определенной критике. Оппоненты утверждают, что технические решения, основанные на подобной схеме, подходят только для тех предметных областей, где каждый выход имеет строго одного владельца. Данное обстоятельство хорошо стыкуется с моделью валюты, и поэтому работает в случае с платформой Биткоин. А вот для приложений общего назначения, допускающих притязание на один выход нескольких владельцев (конфликтующие выходы), использование UTXO может привести к неуправляемому шторму неподтвержденных транзакций, и даже к комбинаторному взрыву блокчейна.

К счастью, в этом разделе книги мы рассматриваем именно криптовалюту. Даже самые ярые критики UTXO признают, что для данной цели модель UTXO подходит наилучшим образом.

Итак, платформа Биткоин отслеживает все доступные (неизрасходованные) выходы UTXO (на февраль 2019 года общая масса выпущенной криптовалюты превысила 16 840 000 BTC). Каждый раз, когда пользователь так или иначе получает биткоин, эта сумма учитывается в платформе как нерастраченный выход. Таким образом, средства конкретного пользователя могут быть раскиданы по пулу UTXO среди тысяч транзакций и сотен блоков. Именно поэтому платформа исключает такое понятие как баланс биткоин-адреса или счета и оперирует только отдельными нерастраченными выходами, привязанными к конкретному владельцу. Впрочем, для удобства пользователей программное обеспечение кошельков эмулирует балансы счетов пользователя, которые прекрасно отражаются в интерфейсной части приложения. Для этого кошельки сканируют блокчейн и суммируют все нерастраченные выходы, принадлежащие данному пользователю.

Неизрасходованный выход транзакции может блокировать любую произвольную сумму в биткоинах (или сатоши). При этом он является неделимым (не станете же Вы отрывать кусочки от пятитысячной купюры для оплаты покупки в 500 рублей). Аналогия тут совершенно полная. Это означает, что для осуществления сделки требующей купюры меньшего номинала, чем имеющийся в Вашем распоряжении неизрасходованный выход, Вам все равно придется потратить его целиком. На самом деле все не так печально. Сдачу самому себе никто не отменял. Т.е., транзакция кроме одного выхода, отправляющего требуемую сумму по адресу получателя платежа, будет содержать еще один выход – сдачу самому себе (аналогичным образом, если средств в одном нерастраченном выходе недостаточно, следует аккумулировать несколько UTXO, т.е., транзакция может включать несколько входов). Ну и о вознаграждении майнерам не стоит забывать. Все ровно так как показано на рисунке 4.

Приложение кошелька пользователя, как правило, формирует платеж из имеющихся в распоряжении различных неизрасходованных выходов, так чтобы итоговая сумма превышала или равнялось величине перевода. Могут использоваться различные стратегии в зависимости от предпочтений владельца.

Почему модель UTXO называют симметричной? Нерастраченные выходы в блокчейне, которые транзакция потребляет являются ее входами. Одновременно с этим создаются новые нерастраченные выходы, которые становятся выходами данной транзакции. Фраза конечно звучит не очень благозвучно, но по сути здесь все верно.

Созданная транзакция тратит выход предшествующей транзакции, разблокируя его подписью текущего владельца денежных средств. В тоже самое время указанная в транзакции сумма сразу же блокируется механизмом, использующим биткоин-адрес нового обладателя.

А теперь попробуйте ответить на вопрос: что является первичным вход или выход транзакции?

Есть только одно исключение из сформулированного правила в модели UTXO. Речь идет о так называемых coinbase-транзакциях, являющихся источником эмиссии криптовалюты Биткоин. Эти токены отправляются майнеру, которому удалось сформировать очередной валидный блок, включенный в блокчейн. Первой транзакцией любого блока является такая coinbase-транзакция, фиксирующая факт перевода майнеру, сгенерировавшему этот блок, гонорара за его труды. Разумеется, майнер сам эту транзакцию и формирует в процессе создания блока. Он даже может в качестве получателя новеньких биткоинов указать любого другого участника сети (не думаю, что на практике майнеры часто поступают таким образом). Особенностью этой цифровой записи является отсутствие входов. Ее выход создается из воздуха.

Признаться, в блокчейне Биткоина есть еще одна особая, единственная транзакция, входящая в его генезис-блок. Первые 50 BTC были отправлены Сатоши Накамото по адресу 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa. Никаких входов у этой транзакции конечно же нет. Но

поскольку блок генезиса жестко зашит в программном обеспечении всех приложений (кстати это типичная ситуация для генезис-блоков практически всех криптовадют) непонятно каким образом эти монеты вообще можно потратить. Впрочем, Сатоши Накамото такой же большой оригинал в отношении использования собственных криптокапиталов как и во всем остальном.

Как мы уже знаем, передача биткоинов - это создание неизрасходованного выхода транзакции (UTXO), зарегистрированного на получателя платежа. Каждая полная нода в сети Биткоин отслеживает множество (пул) нерастраченных выходов (UTXO).

Выходы транзакции

Выход транзакции состоит из двух основных частей: суммы платежа (в сатоши) и блокирующего сценария, также называемого "обременением" - скрипта, блокирующего указанную сумму вплоть до выполнения условий, соблюдение которых снимает обременение.

Как именно создаются сценарии мы подробно рассмотрим немного позже, а пока в таблице 7 представим подробное строение выхода транзакции.

Таблица 3.2. Структура выхода транзакции

Размер	Поле	Описание
8 байт	Сумма	Отправляемая новому владельцу сумма. Неотрицательное целое число. Сумма номинирована в сатоши (1 BTC = 108 сатоши).
От 1 до 9 байт	Размер скрипта обременения	Неотрицательное число, в байтах.
Без ограничения	ScriptPubKey Обременение	Сценарий, определяющий условия, после удовлетворения которых средства можно будет потратить

Обременение

Выходы транзакций инкапсулируют определенную сумму и фиксированное обременение, определяющее условие вывода средств. В большинстве случаев, блокирующий скрипт ассоциирует обременение с определенным адресом в сети Биткоин – адресом получателя платежа. Таким образом право собственности на выделенные средства передается новому владельцу. Если мы воспользуемся ставшим классическим примером оплаты, адресованной хозяину кофейни Сергею, чашки кофе стоимостью 0,025 BTC, выпитой Мариной, то после окончания всех инфраструктурных процессов в блокчейне появится транзакция на сумму 2 500 000 сатоши с обременением, блокирующем средства на биткоин-адрес Сергея. Недешевый вышел кофе. Выход этой транзакции на сумму 2 500 000 сатоши становится частью пула неизрасходованных выходов транзакций, а кошелек Сергея отметит пополнение суммы доступных средств. Когда Сергей решит потратить эту сумму, его новая транзакция снимет блокировку с выхода (UTXO), предоставив соответствующий скрипт, содержащий открытый ключ и электронную подпись, произведенную с помощью закрытого ключа Сергея.

Входы транзакции

В самой простой интерпретации входы транзакций можно рассматривать как указатели на неизрасходованные выходы (UTXO). Они ссылаются на конкретный нерастраченный выход, используя хеш транзакции и порядковый номер (в одной транзакции может содержаться несколько выходов). Но просто так нельзя обратиться к понравившемуся Вам выходу, хранящему привлекательную сумму. Такой вход не будет принят системой, и, честно говоря, вообще не может появиться. Вы также должны доказать платформе право владения конкретным неизрасходованным выходом. Поэтому вход транзакции должен включать отпирающий сценарий, удовлетворяющий условиям снятия обременения, установленным данным выходом. Как правило это электронная подпись, доказывающая факт владения биткоин-адресом, фигурирующим в сценарии блокировки. А, если быть совсем точным, то скрипт состоит из двух компонент: электронной подписи и открытого ключа пользователя. Открытый ключ служит подтверждением того, что создатель транзакции имеет право распоряжаться суммой с указанных выходов. Второй компонент – это ECDSA-подпись хеша упрощенной версии транзакции. Совместно с открытым ключом, подпись подтверждает, что транзакция была создана истинным владельцем данного биткоин- адреса.

Разработчики программного обеспечения, если не желают связываться со всей структурой блокчейна, могут получить пул неизрасходованные выходов (pool UTXO) или набор выходов, ассоциированных с конкретным биткоин-адресом с помощью RPC-вызова кошелька Bitcoin Core или API сторонних приложений, например, blockchain.info.

Когда пользователь совершает перевод средств, его клиент собирает входы транзакции путем выбора из нерастраченных выходов. Например, чтобы провести платеж суммой 0,025 BTC, программное обеспечение кошелька может выбрать два выхода с 0,02 BTC и 0,005 BTC и сложить их балансы для получения требуемой суммы. После того, как выходы выбраны, кошелек для каждого из них генерирует разблокирующие скрипты, содержащие подписи. Поскольку условия обременения выходов требовали именно этого, все прекрасно сработает.

Формальные характеристики структуры данных для хранения входа транзакции представлены в таблице 3.3.

Таблица 3.3. Структура входа транзакции

Размер	Поле	Описание
32 байта	Хеш-дайджест более ранней транзакции	Указатель на более раннюю транзакцию, содержащую выход, который будет использован. Значение представлено в виде двойного SHA-256 хеша более ранней транзакции.
4 байта	Номер выхода	Индекс выхода более ранней транзакции, начиная с 0. Неотрицательное целое.

От 1 до 9 байт	Размер разблокирующего скрипта	Неотрицательное число, в байтах.
Без ограничения	ScriptSig блокировки	Снятие Сценарий, удовлетворяющий условиям снятия обременения, установленным данным выходом
4 байта	Номер последовательности	В настоящее время отключенная функция замены транзакции. Установлено в 0xFFFFFFFF.

Поле "Номер последовательности" предназначено для переопределения транзакции до истечения момента времени, установленного в ее поле Locktime - возможность, которая в настоящее время отключена. В большинстве транзакций в это поле записано максимальное значение (0xFFFFFFFF), которое просто игнорируется платформой Биткоин. Если в транзакции поле Locktime установлено в ненулевое значение, тогда, как минимум, один из ее входов должен иметь порядковый номер меньше чем 0xFFFFFFFF.

Комиссия за транзакцию

Цель: Сформировать представление о функциях комиссии за транзакции, правилах их вычисления.

В большинстве транзакций номинальная сумма включает в себя комиссионные майнерам. На этапе становления сети Биткоин число транзакций было относительно не большим, поэтому майнеры, собирая блоки, буквально дрались за них, подбирая даже бесплатные. Сейчас ситуация коренным образом поменялась. Без достаточной мотивации Ваша транзакция рискует долго оставаться вне поля внимания майнеров.

Большинство кошельков учитывают комиссионные автоматически. Однако, если Вы предпочитаете интерфейс командной строки, или создаете транзакции программно, Вам придется рассчитывать и включать комиссионные самостоятельно.

Кроме стимулирования майнеров, комиссия выполняет важнейшую системную функцию – избавляет сеть Биткоина от спам-транзакций – за них тоже придется платить, причем столько же сколько и за нормальные. Готовы потратиться?

Величина комиссионных рассчитывается на основе размера транзакции в килобайтах и не зависит от переводимой суммы. На самой заре биткоина величина комиссии была фиксированной и постоянной по всей сети. Но постепенно в благородном деле майнинга установились вполне себе рыночные отношения. В приоритете у современных майнеров - транзакции с повышенной комиссией. Стоит отметить, что часть цены комиссии обусловлена системными причинами и регулируется на основе пропускной способности сети и числа транзакций, ожидающих обслуживания. Все, что касается пропускной способности платформы Биткоин, всегда подвергалось самой решительной критике. А решение критичных технических вопросов имеет тенденцию со временем переходить в финансовую плоскость.

Но это не значит, что бесплатная транзакция никогда не будет обработана. Логика у пользователей здесь совершенно очевидная. Если Вы очень торопитесь, то повысьте вознаграждение за свою транзакцию, и она очень быстро попадет в один из ближайших блоков. Согласно протокола Биткоин комиссионные не являются обязательными, и транзакции без вознаграждения в итоге могут быть обработаны. Но с достаточными комиссионными это произойдет наверняка намного быстрее.

Теоретически, комиссия за транзакцию может составить 0%, при соблюдении следующих условиях:

размер транзакции менее 1000 байт;

сумма на каждом выходе не менее чем 0.01 BTC;

высокий приоритет.

Программное обеспечение современных кошельков вполне адекватно рассчитывают оптимальный размер комиссии. Заодно они предложат прогноз скорости проведения транзакции. Но если Вы попытаетесь рассчитать оптимальную комиссию самостоятельно, учтите, что это многофакторная задача. А многие рекомендации, транслируемые из одного издания в другое, безнадежно устарели.

Добавление вознаграждения к транзакции

В структуре данных транзакции не предусмотрено поля для комиссии. Майнер забирает себе разность между суммой входов и суммой выходов. Любое количество сатошей, оставшееся после вычитания суммы выходов из суммы входов транзакции, признается комиссией. Если Вы создаете свои транзакции собственноручно, обязательно убедитесь в том, что случайно не установили очень большие комиссионные, например, забыв включить в транзакцию дополнительный выход со сдачей самому себе.

Итак, величина комиссии за транзакцию напрямую зависит от ее размера. А что же влияет на размеры транзакции? Расклад здесь примерно такой:

каждый адрес, с которого получены средства, добавляет 148 байт;

каждый адрес, на который отправляют средства, добавляет 34 байта;

структура транзакции занимает 10 байт (независимо от числа входящих в нее адресов).

Сложные схемы транзакций и осиротевшие транзакции

Мы уже рассмотрели механизмы, благодаря которым транзакции образуют собственные, внутренние по отношению к блокчейну цепочки, в результате чего одна транзакция тратит выходы предшествующей транзакции и создает выходы для последующей транзакции. Модель UTXO отлично справляется с регулированием таких потоков денежных средств.

А теперь представим, что все транзакции такой цепочки поступили в сеть почти одновременно. Такие сложные транзакционные схемы – отнюдь не беспочвенные фантазии.

В качестве примера упомянем CoinJoin-транзакции. CoinJoin — это протокол, призванный усилить уровень конфиденциальности транзакций в сети Биткоин. С его помощью несколько пользователей после согласования друг с другом могут создавать совместно подписанную транзакцию, смешивая личные балансы в один общий. Впервые был предложен в 2013 году.

Когда цепочка транзакций передается по сети, изначальный хронологический порядок легко может нарушаться. В этом случае некоторая нода сначала получает дочернюю транзакции и лишь через какое-то время – родительскую, выходы которой были задействованы в дочерней. Получив подобную, казалось бы, непригодную транзакцию с неправильными входами, нода вместо того, чтобы признать ее недействительной, помещает ее в специальный, временный пул, где дочерняя транзакция будет ожидать прибытия родительской. Поскольку транзакция не была признана недействительной, она также будет ретранслирована другим узлам.

Подобный временный пул называют пулом (множеством) сиротских транзакций. После поступления родительской транзакции, все сироты, ссылавшиеся на ее выходы, освобождаются из сиротского пула, еще раз проверяются, после чего переводятся на уровень рабочего пространства сети Биткоин, где они могут быть включены в очередной блок.

Таким образом, наличие пула для сирот гарантирует, что валидные транзакции не будут отброшены только потому, что их родитель задерживается в пути и что, в конечном итоге, цепочка, которой они принадлежат, будет реконструирована в правильном порядке, независимо от хронологии поступления ее составных частей.

На количество транзакций потеряшек, хранимых в памяти, имеется ограничение, что позволяет предотвратить DoS или DDoS-атаки. Этот лимит отражается как параметр MAX_ORPHAN_TRANSACTIONS в исходном коде биткоин-клиентов. Если число бесхозных транзакций в пуле превышает критическое значение, лишние будут удалены. Выбор кандидатов на ликвидацию происходит случайным образом.

Язык сценариев транзакций

Цель: Сформировать представление о языке сценариев транзакций Script в необходимом объеме.

Проверка валидности транзакции в сети Биткоин осуществляется на основе выполнения скрипта, записанного на специальном языке сценариев, называемым Script. Это Форт (Forth) – подобный стековый язык программирования с обратной польской нотацией, позволяющей избежать присутствия скобок и приоритетов в выражениях, что чрезвычайно удобно для вычисления формул в ЭВМ на основе стеков. Важная особенность таких языков - использование стека для передачи параметров между термами, что позволяет очень гибко и просто реализовывать достаточно сложные конструкции. Отличительная особенность обратной польской нотации заключается в том, что аргументы располагаются перед знаком операции. Стек - это коллекция, элементы которой доступны согласно принципа "последний вошел, первый вышел" ("Last-In-First-

Out", LIFO). Т.е., в любой момент времени мы будем иметь дело только с элементом, добавленным последним (лежащим на вершине стека). Как известно, структура данных стек поддерживает две операции: добавление (push) и удаление (pop).

Неполнота по Тьюрингу для языка Script в первую очередь означает отсутствие циклов и сложных инструментов управления ходом выполнения кода. Для сети Биткоин данное свойство означает ограничение сложности сценариев и предсказуемое время их выполнения. Это решение принято, исходя из соображений безопасности системы. Ограничения языка Script гарантируют отсутствие бесконечных циклов или других форм так называемых "логических бомб", способных привести к атакам на отказ в обслуживании. Таким образом, механизм проверки транзакций становится практически неуязвимым. Эта характеристика особенно важна, если учесть, что каждая транзакция в сети Биткоин проверяется каждой полной нодой.

Почему в данном случае речь не идет о SMART-контрактах? Хотя некоторые принципы умных контрактов впервые были заложены именно в протоколе Биткоин, его платформа не может реализовать их в клиентском программном обеспечении по соображениям безопасности. Поэтому большинство разработчиков создают цифровые контракты либо на новых платформах, таких как Ethereum, либо экспериментируют с сайдчейнами (вилками основной сети биткоина). Язык Script подходит для программирования операций, содержащих только простые условия. Он не является Тьюринг-полным, а сама платформа не имеет маркеров состояния, чтобы на их основе можно было проектировать приложения. К тому же пропускная способность сети Биткоин сильно ограничена, что также не позволяет создать систему многофункциональных контрактов.

Выполнение скрипта, как и положено в системах с постфиксной нотацией, осуществляется по одному элементу слева направо. Обрабатываемые данные добавляются в стек. Операторы добавляют или извлекают одно, или несколько значений из стека, производят над ними запланированное действие, результат которого также может добавляться в стек. Инструкции языка (всего их порядка 80) подразделяются на категории (управляющие, для работы со стеком, для работы со строковыми фрагментами, побитовые логические операции, арифметические, криптографические).

Например, оператор OP_ADD извлечет два операнда из стека, вычислит их сумму, и поместит полученный результат на вершину стека. А оператор OP_EQUAL извлекает два операнда из стека, сравнивает их и добавляет в стек результат сравнения (ИСТИНА (кодируется как цифра 1) – если операнды равны и ЛОЖЬ (кодируется как цифра 0) – в противном случае).

Рассмотрим пример скрипта для решения следующей задачи: проверить равна или нет числу 5 сумма двух чисел 2 и 3.

Сценарий решения этой простой задачи будет выглядеть следующим образом:

2 3 OP_ADD 5 OP_EQUAL

Логика выполнения сценария биткоин-клиентом демонстрируется на рисунке 3.2.

Транзакция признается действительной, если указатель стека принимает в итоге значение TRUE (OP_TRUE), т.е. равен 1, или после выполнения скрипта, стек окажется пустым, или на его вершине окажется любое другое ненулевое значение.

Транзакция признается недействительными, если после выполнения скрипта на вершине стека окажется значение FALSE (OP_FALSE), или выполнение сценария будет прервано оператором (например, OP_VERIFY, OP_RETURN), операторной скобкой (например, OP_ENDIF) и др.

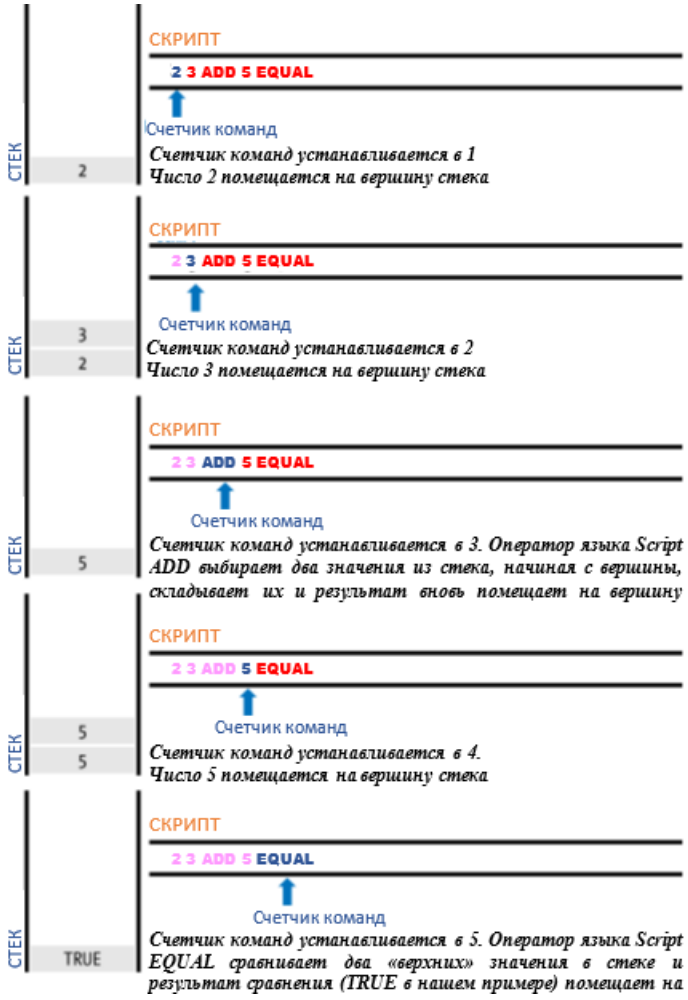

Рис. 3.2. Выполнение простейшего сценария биткоин-машиной

В большинстве случаев блокирующие сценарии ссылаются на биткоин-адреса или публичные ключи, требуя предоставления доказательства их владением для преодоления обременения. Однако, сценарий совсем не обязательно должен быть таким сложным. Подойдет любая конкатенация из блокирующего и разблокирующего сценариев, результатом которой является значение TRUE (OP_TRUE).

Возвращаясь к последнему примеру, мы может использовать в качестве блокирующего скрипта набор инструкций:

3 OP_ADD 5 OP_EQUAL

Разблокировать такой выход можно с помощью транзакции, вход которой содержит следующий скрипт разблокировки:

2

Программное обеспечение клиента объединяет блокирующий и разблокирующий сценарии, в результате чего получается уже знакомый нам скрипт:

2 3 OP_ADD 5 OP_EQUAL

Результат исполнения этого сценария равен OP_TRUE, следовательно, транзакцию следует считать действительной. Наверное, это не совсем надежная защита для заблокированных средств, поскольку выход с таким обременением может потратить любой, кто знает, что 2+3=5.

Независимость от состояния системы

Язык сценариев не требует восстановления состояния до выполнения скрипта и сохранения состояния после. Все данные, необходимые для работы сценария, содержатся в нем самом. Любой сценарий единообразным образом будет исполнен в любом клиенте. Если Ваша система выполнила сценарий с положительным результатом, можно не сомневаться в том, что любая другая система в сети Биткоин завершит сценарий с точно таким же результатом. Таким образом, проверка транзакции инвариантна относительно инфраструктуры сети и любых внешних условий.

Сценарии транзакций

Цель: Сформировать комплекс знаний о схеме выполнения блокирующего и разблокирующего сценария в рамках реализации механизма проверки действительности транзакций, а также всей системы передачи ценности от одного владельца другому в блокчейне платформы Биткоин.

IT-специалисты давно занимались проблемой хранения связанных данных. Разрабатывались разные модели: реляционные; иерархические; сетевые. Однако, идея, реализованная в платформе Биткоин, была поистине революционной. Семантическая связь внутри цепочек транзакций реализуется на основе программного кода. Элегантная, прозрачная и абсолютно надежная схема. В 2009 году в свет вышли не просто электронные монеты. Биткоин – это программируемые деньги.

Создание сценария

Механизм проверки действительности транзакций, а заодно, и вся система передачи ценности от одного владельца другому в блокчейне платформы Биткоин опирается на два типа сценариев: блокирующий сценарий и разблокирующий. Блокирующий скрипт воплощает налагаемое на выход транзакции обременение. Здесь определяются условия, выполнение которых откроет в будущем доступ к средствам, замороженным на выходе транзакции. Исторически сложилось так, что сценарий блокировки назывался scriptPubKey (сценарий открытого ключа). В большинстве случаев этот скрипт действительно хранил открытый ключ или биткоин-адрес. Однако, мы уже не раз упоминали, что возможный спектр блокирующих сценариев гораздо шире. При этом традициям (особенно в том, что касается терминологии) тоже необходимо следовать.

Аналогично, за разблокирующими скриптами закрепилось название scriptSig, поскольку, являясь частью каждого входа транзакции, они, как правило, содержат электронную подпись пользователя, сделанную с помощью его закрытого ключа.

Программное обеспечение любого клиента платформы Биткоин проверяет правомочность и соответствие стандартам каждой транзакции, выполняя ее блокирующий и разблокирующий сценарии вместе. При этом проверяются все входы транзакции, для чего сначала из пула извлекаются все нерастраченные выходы, на которые эти входы ссылаются. Каждый анализируемый выход содержит соответствующий блокирующий сценарий, определяющий условия доступа к средствам. После этого, для каждого входа оба сценария выполняются вместе. Схематично данная процедура отображена на рисунке 3.3.

Рис. 3.3. Проверка одного входа транзакции

А на рисунке 3.4 представлен пример записи объединенного сценария для самого популярного вида платежа на хеш открытого ключа (Pay-to-Public-Key-Hash).

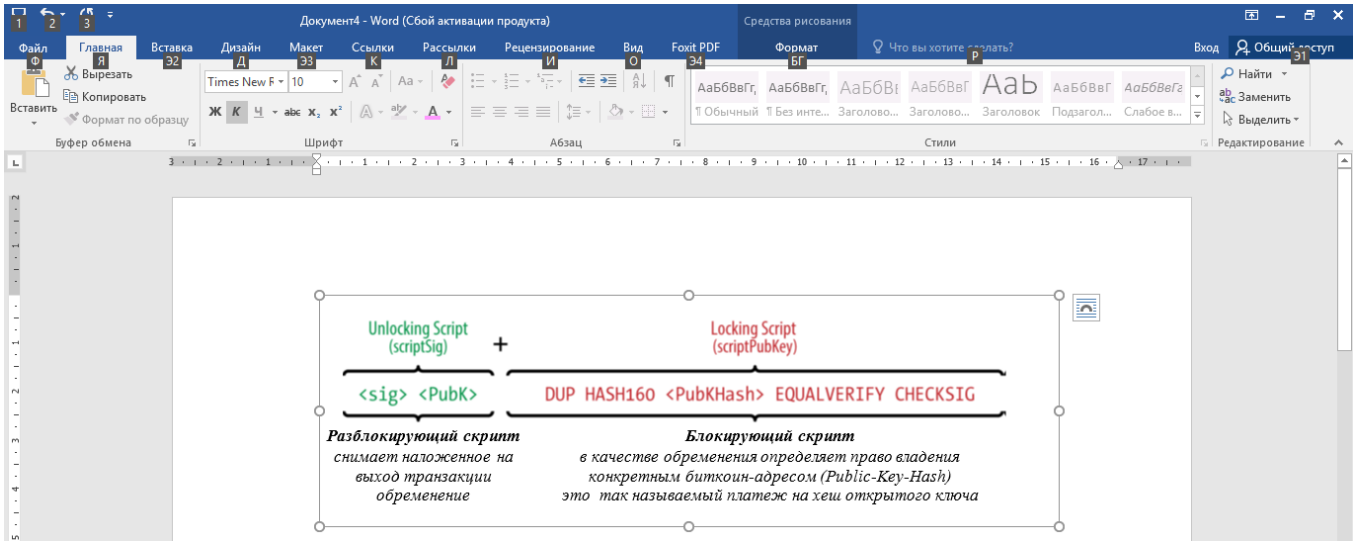

Рис. 3.4. Комбинированный сценарий для проверки транзакции в формате платеж на хеш открытого ключа

Отметим, что вплоть до 2010 года платформа Биткоин работала в полном соответствии с изложенным алгоритмом. Скрипты сначала извлекались, потом склеивались и на исполнение поступал уже единый, объединенный сценарий. Однако соображение безопасности (при определенных условиях разблокирующий сценарий мог повредить блокирующий) заставили внести в эту процедуру определенные поправки. В настоящее время схема немного изменилась. Каждый

из скриптов выполняется отдельно, по очереди, а полученный после выполнения первого скрипта результат передается через стек. Рассмотрим подробнее каким образом это реализуется.

Сначала выполняется разблокирующий скрипт. Если он выполнился без ошибок (например, не осталось лишних операторов), основной (не альтернативный) стек копируется. Затем запускается на исполнение блокирующий скрипт. Если результат выполнения блокирующего скрипта с ранее скопированным из стека результатом разблокирующего скрипта дает значение "TRUE", то данный вход транзакции признается действительным (в случае получения такого же результата проверок для остальных входов, транзакция считается полностью готовой (валидной) к включению в блокчейн). Т.е., разблокирующий скрипт смог полностью удовлетворить условиям обременения, закодированным в блокирующем скрипте и, следовательно, данный вход успешно подтвердил полномочия на трату конкретного неизрасходованного выхода. Если же в результате выполнения объединенного сценария получен иной результат (не "TRUE"), тогда вход признается недействительным, как не удовлетворившим условия доступа к средствам, закодированные в соответствующем нерастраченном выходе. Поскольку нерастраченный выход вместе с остальными частями транзакции уже записан в блокчейн (т.е., никогда не меняется), никакие недействительные (неудачные, или злонамеренные) попытки израсходовать его по ссылке из новой транзакции никак не могут на него повлиять. Только действительная транзакция, удовлетворяющая условиям ограничения, изменит состояние нерастраченного выхода на "растраченный" и вызовет его удаление из пула доступных (нерастраченных) выходов.

Да, конечно, Script язык с очень скромными возможностями, которому ох как далеко до современных инструментов разработчиков умных контрактов (да ему эти возможности и не нужны, если честно). Однако сама идея хранить код обработки непосредственно в самих структурах данных (прообраз умных контарктов) безусловно впервые была успешно реализована именно в сети криптовалюты Биткоин. К тому же история развития этой платформы демонстрирует иную тенденцию – сужение возможностей и вариабельности платежных конструкций и схем в целях безопасности. Об этом мы поговорим в следующем параграфе.

Разрешенные в платформе Биткоин типы транзакций

Цель: Сформировать комплекс знаний о типах разрешенных в платформе Биткоин транзакций и реализуемых с их помощью различных схемах платежей в сети Биткоин.

Следует сразу предупредить читателя, что мы ступаем на довольно скользкую почву, в том плане, что никаких догм, жестко заданных правил в блокчейн-платформах обычно не существует (кроме содержимого самого блокчейна – он неизменен). Технология настолько молодая, что изменения стандартов идет перманентно. Поэтому содержимое данного параграфа актуально в момент написания и вполне возможно уже не полностью будет отвечать действительности при прочтении. Хотя тот набор, который мы сейчас разберем не менялся в течении нескольких последних лет. И для реализации многих приложений его не вполне достаточно. Поэтому работа в

плане увеличения гибкости транзакционной модели сети Биткоин никогда не прекращалась. Правда, чаще всего подобные истории заканчивались с противоположным результатом – вводились новые ограничения и запреты.

Итак, что мы сегодня имеем в плане разнообразия типов биткоин-транзакций?

Почти сразу же разработчики внесли весьма серьезные ограничения на типы сценариев, которые должны поддерживаться эталонным клиентом. Всего было определено ровно пять возможных вариантов. Транзакции, выполненные только по таким лекалам, будут восприниматься программным обеспечением клиентов, и, самое главное, майнеров – без их участия транзакция никогда не попадет в блокчейн. Вы конечно можете создать нестандартную транзакцию, не относящуюся ни к одному задекларированных типов, но тогда Вам придется искать майнера, который признает ее валидной и включит в блок.

Разрешенными являются следующие типы сценариев транзакций:

pay-to-public-key-hash (P2PKH);

public-key;

multisignature (с ограничением на 15 ключей);

pay-to-script-hash (P2SH);

выход данных (OP_RETURN).

Поговорим о них подробнее.

Pay-to-Public-Key-Hash сценарии

Большинство транзакций, наполняющих сегодня сеть Биткоин, используют один и тот же сценарий проверки валидности - Pay-to-Public-Key-Hash (P2PKH, платеж на хеш открытого ключа), проецирующий на криптовалютную платформу простейшую операцию перевода денежных средств от одного лица другому, но только в обезличенной форме.

Такие транзакции включают блокирующие сценарии, обременяющие выходы хешами открытых ключей, известных также как биткоин-адрес. Транзакции, реализуемые по такой схеме, содержат скрипты P2PKH. Нерастраченный выход, блокированный с помощью сценария P2PKH, можно открыть, предоставив открытый ключ и электронную подпись, сделанную с помощью соответствующего закрытого ключа.

Вернемся к рассмотренному ранее примеру с Мариной, оплатившей биткоинами чашку кофе, выпитую в кофейне Сергея. Выход соответствующей транзакции будет заблокирован скриптом scriptPubKey следующей конструкции:

OP_DUP OP_HASH160 <Public-Key-Hash Сергея> OP_EQUALVERIFY OP_CHECKSIG

"Public-Key-Hash Сергея" представляет собой биткоин-адрес кошелька владельца кофейни в шестнадцатеричном формате (т.е., без перевода в кодировку Base58Check с характерной для этого случая "1" вначале). Это вполне естественно, если вспомнить, что формат Base58Check придуман

для лучшего восприятия больших чисел людьми. А компьютеры легко разберутся и с шестнадцатеричным представлением данных.

Когда Сергею понадобятся эти средства, он легко сможет снять с них обременение с помощью следующего разблокирующего скрипта (scriptSig), входящего в состав новой транзакции (в качестве одного из ее входов):

<Signature Сергея> <Public-Key Сергея>

Здесь под термом Signature понимается электронная подпись владельца криптопары.

Если объединить оба скрипта в один сценарий, мы получим уже отчасти знакомый нам программный код на языке Script:

<Signature Сергея> <Public-Key Сергея> OP_DUP OP_HASH160 <Public-Key-Hash Сергея> OP_EQUALVERIFY OP_CHECKSIG

После выполнения этого комбинированного сценария, мы получим на вершине стека единственное значение "ИСТИНА" тогда и только тогда, когда разблокирующий скрипт подойдет условиям, закодированным в блокирующем скрипте. Воспользоваться средствами может только обладатель закрытого ключа Сергея, представив для снятия обременения свою электронную подпись. В штатной ситуации это и есть сам Сергей.

Ход выполнения сценария в случае P2PKH-транзакции проиллюстрирован рисунком 3.5 (в целях улучшения восприятия у операторов языка Script на рисунке опущены префиксы OP_). Причину разделения процесса на две части Вы уже знаете. Скрипты объединенного сценария выполняются последовательно, а результат первого передается во второй посредством стека.

Выполнение сценария начинается с разблокирующего скрипта scriptSig, включающего всего два шага. Сначала на вершину стека добавляется электронная подпись владельца биткоин-адреса, на хеш которого был совершен платеж. Затем – на вершину стека добавляется соответствующий открытый ключ.

В самом начале второй части комбинированного сценария - скрипта обременения scriptPubKey выполняется дублирование указателя стека, т.е., самого верхнего значения - открытого ключа. Инструкция OP_HASH160 на самом деле вычисляет двойной хеш, применяя последовательно хеш-функции SHA-256 и RIPEMD-160 к указателю стека (значению на вершине), а полученный результат вновь добавляется в стек. Т.е., на вершине стека окажется величина, равная RIPEMD-160(SHA-256(<отрытый ключ>)). Мы с Вами помним, что это формула вычисления биткоин-адреса. Аналогичная по смыслу величина у нас вшита в самом скрипте обременения – биткоин-адрес, на который собственно и были в свое время отправлены средства. Это следующий операнд скрипта, появление которого в нем означает, что биткоин-адрес получателя платежа размещается на вершине стека. Осталось их сравнить. Что и делает инструкция OP_EQUALVERIFY. С ее помощью из стека выбираются два верхних значения (биткоин-адрес действительного получателя средств и биткоин-адрес претендента на эти средства)

и осуществляется их сравнение. Результат сравнения добавляется к стеку. На самом деле команда OP_EQUALVERIFY немного мощнее, чем простой оператор сравнения, который также в синтаксисе языка Script присутствует - OP_EQUAL. Инструкция OP_EQUALVERIFY подразумевает также в конце исполнения вызов команды OP_VERIFY, отмечающую транзакцию как недействительную, если на вершине стека оказывается значение не равное TRUE. Если сравнение подтвердило совпадение биткоин-адресов, осталось убедиться в реальности прав пользователя на фигурирующий в сценарии биткоин-адрес.

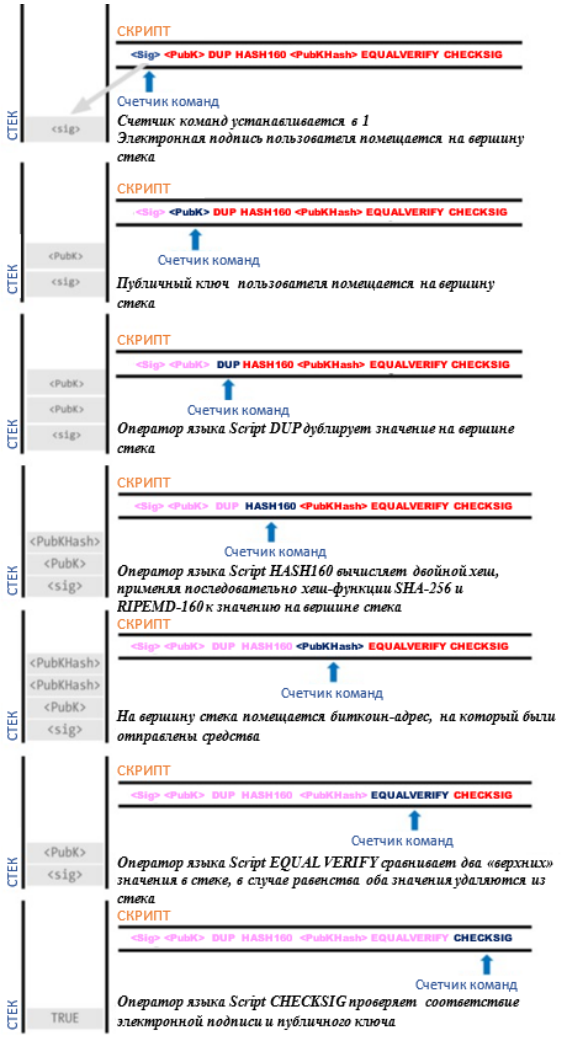

Рис. 3.5. Выполнение комбинированного сценария для P2PKH-транзакции

Поскольку последний рассчитывался через открытый ключ, достаточно проверить действительно ли предъявитель данного открытого ключа является его владельцем. А эту задачу легко можно решить достаточно традиционным способом. Имея пару электронная подпись и открытый ключ, мы всегда можем проверить подходят ли они друг к другу. Именно эту функцию и выполняет последняя инструкция сценария OP_CHECKSIG. Если проверка пройдет успешно, на вершине опустевшего к моменту завершения сценария стека окажется значение 1 (TRUE). В противном случае указатель стека будет возвращать значение 0 – транзакция.

Хотя относительно реализации последнего шага и был употреблен эпитет "традиционно", это утверждение не является справедливым. Традиционной можно назвать процедуру использования открытого ключа для проверки электронной подписи. А вот идея проверки принадлежности публичного ключа на основе электронной подписи является весьма оригинальной. Впрочем, так можно охарактеризовать практически любой компонент платформы Биткоин.

Осталось рассмотреть всего один вопрос. Подпись всегда ставится под документом. Электронная подпись не исключение. На основании какого электронного документа создается электронная подпись в скрипте scriptSig? Электронная подпись генерируется на основе дайджеста (результата хеширования) некоторых полей данной транзакции. Стоит отметить, что подпись никогда не формируется на основе целиком всей транзакции, и пользователи могут указывать, какая часть транзакции подписывается (конечно, кроме поля scriptSig).

Pay-to-Public-Key сценарии

Эта модель транзакций появилась раньше остальных и использовалась еще на заре рождения платформы Биткоин, когда в скрипте в качестве адреса поначалу указывался IP-адрес получателя, а в последствии более универсальный идентификатор - открытый ключ. Минусами такого похода являются раздутые транзакции (из-за длинных адресов) и передача по сети открытых ключей в незашифрованном виде, что может с появлением квантовых компьютеров обернуться очень серьезными неприятностями для сети Биткоин, поскольку вероятно появится возможность получить закрытые ключи по открытым.

Pay-to-public-key — является слегка упрощенной формой платежа по сравнению с моделью pay-to-public-key-hash. Сегодня этот тип транзакций чаще всего используется в coinbase-транзакциях, создаваемых старым программным обеспечением ортодоксальных майнеров.

Блокирующий скрипт scriptPubkey записывается следующим образом:

<Public Key A> OP_CHECKSIG

А в качестве разблокирующего скрипта scriptSig используется только подпись получателя:

< Signature A >

Догадались откуда пошли названия блокирующего и разблокирующего скриптов?

Комбинированный сценарий, проверяемый программным обеспечением систем платформы Биткоин имеет очень вид:

<Signature A> <Public Key A> OP_CHECKSIG

Вся проверка ограничивается добавлением в стек двух значений (подписи и открытого ключа) и вызовом для их сравнения единственного хорошо нам знакомого оператора OP_CHECKSIG.

Сценарии с мульти-подписью

Для чего используется мульти-подпись? Основные цели — разделение ответственности или повышение безопасности.

Принцип разделения ответственности призывает к тому, чтобы несколько человек коллегиально принимали решения по любому критически важному вопросу. Решение каждого из менеджеров фиксируется с помощью применения персональной электронной подписи. Примером может послужить ситуация, когда компания имеет двух и более владельцев. Например, чтобы потратить существенную сумму, владельцы должны прийти к консенсусу (знакомая для нас тема), при этом территориально они могут быть удалены друг от друга.

Безопасность с помощью мульти-подписи можно повысить, используя стратегию раздельного хранения ключей, необходимых для подписи. Как вариант - на разных устройствах. Например, один ключ записан на персональном компьютере, а другой — на мобильном устройстве. В такой ситуации задача злоумышленника, стремящегося похитить криптовалюту пользователя, существенно усложняется.

Еще один вариант использования мульти-подписи — это безопасность торговых сделок. В случае, когда продавец не доверяет покупателю (и наоборот), сделка происходит на условиях, когда оба участника операции должны подтвердить то, что она состоялась. Обычно для этого используют депозитный счет. Он может принадлежать третьему лицу, выполняющему роль арбитра, или обоим участникам сделки. В случае, когда стороны не привлекают третью сторону, процесс может быть организован созданием общего счета с необходимостью подписать сделку всеми владельцами счета. На этот счет каждая сторона заранее отправляет залог (желательно превышающий стоимость сделки). После успешного завершения сделки залоги возвращаются. Если сделка сорвана, то залоги теряются.

В общем, имеются самые разные схемы использования мульти-подписи и механизмы ее реализации. Например, популярнейшие блокчейны Биткоин и Ethereum используют разные технологии. У Биткоина имеется решение на основе P2SH - транзакции. Ethereum использует дополнительные адреса с одной подписью, контролируемые умными контрактами. Вторая технология сложнее, но при этом достигается большая гибкость приложений.

Возвращаемся к рассмотрению особенностей скриптов различных типов транзакций сети Биткоин.

Скрипт в транзакции с мульти-подписью вводит обременение, согласно которому из N публичных ключей, упомянутых в сценарии, по меньшей мере M должны быть использованы при создании подписи. Только после этого обременение будет преодолено.

Ранее уже упоминалось, что эта схема называется M-из-N, где N - общее количество ключей, а M - пороговое число подписей, необходимых для признания транзакции действительной.

Блокирующий скрипт scriptPubkey записывается следующим образом:

M<pubKey A> <pubKey B> … <pubKey N> N OP_CHECKMULTISIG

А в качестве разблокирующего скрипта scriptSig используется только подпись получателя:

OP_0 < Signature A > < Signature B > … < Signature M >

Префикс OP_0 необходим для того, чтобы обойти ошибку в оригинальной реализации OP_CHECKMULTISIG.

Комбинированный сценарий мульти-подписи 2-из-3 выглядит следующим образом:

OP_0 <Signature A> <Signature C> 2 <pubKey A> <pubKey B> <pubKey C> 3 OP_CHECKMULTISIG

При выполнении этот комбинированный сценарий закончится успешно (в стеке окажется единственное значение TRUE) тогда и только тогда, когда разблокирующий скрипт будет содержать действительные электронные подписи на основе двух любых закрытых ключей, соответствующих двум из трех открытых ключей.

Сценарии с выводом данных (OP_RETURN)

Безусловно, сеть Биткоин является криптовалютной платформой, предназначенной для проведения платежей и транзакций. Т.е., ее основная функция – это реализация покупной способности. Однако, на самом деле ее возможности далеко выходят за рамки исключительно платежной системы. Многие разработчики пробовали использовать язык сценариев транзакций, чтобы привнести такие преимуществами децентрализованных приложений как безопасность и устойчивость в системы, автоматизирующие самые разные области профессиональной деятельности. В первую очередь подобные технологии проникли в сферы цифровых нотариальных услуг, сертификации, государственных кадастров и реестров, умных контрактов.

Для подобных технических решений на начальном этапе характерной особенностью являлось создание выходов транзакций, записывающих данные непосредственно в блокчейн. Например, распределенный реестр может хранить цифровые отпечатки файлов (финансовых документов или произведений искусства), что позволит в любой момент времени очень оперативно доказать их существование, подтвердить право собственности, или получить необходимую информацию.

Откровенно говоря, вопрос применения блокчейна платформы Биткоин для хранения посторонних (не связанных с платежами) данных всегда вызывал бурную дискуссию. Копий на этот счет было сломано немало. Часть разработчиков не могла даже мысли допустить о таком варварском использовании поистине бесценных ресурсов блокчейна. Остальные считают своим долгом продвигать технологии распределенных реестров, демонстрируя уникальные возможности платформы Биткоин.

Отметим, что обе точки зрения вполне обоснованы. Дополнительные "непрофильные" данные неизбежно приведут к "раздуванию" блокчейна, увеличению нагрузки на полные ноды, повышению стоимости хранения данных. Негативным образом такой механизм скажется на состоянии пула нерастраченных выходов, поскольку для записи данных в блокчейн понадобятся транзакции, которые никогда нельзя будет потратить (выходы транзакций будут заблокированы несуществующими адресами, для которых просто не существуют секретные ключи). Т.е., база данных UTXO со временем тоже будет перегружена. В отношении последней угрозы со временем было найдено решение.

Наконец, в версии 0.9 эталонного клиента Bitcoin Core между противоборствующими был достигнут определенный компромисс. Разработчики ввели оператор OP_RETURN. Инструкция OP_RETURN позволяет добавить 80 байт данных к выходу транзакции. Тем не менее, в отличие от недействительных выходов оператор OP_RETURN явно создает выход, который доказуемо нельзя потратить. Но при этом такой выход не помещается в пул нерастраченных выходов (UTXO). Собственно, в этом и заключался компромисс. Выходы OP_RETURN записываются в блокчейн, провоцируя увеличение его размера, но не

хранятся в пуле нерастраченных выходов, экономя оперативную память ЭВМ, реализующих функционал полных нод.

Блокирующие скрипты с оператором OP_RETURN выглядят следующим образом:

OP_RETURN <data>

Максимальный размер данных ограничен 80-ю байтами (установлен в феврале 2015 года в Bitcoin Core версии 0.10, первоначально предел составлял 40 байт). Чаще всего это результаты хеширования, например, с помощью алгоритма SHA-256 (длина = 32 байта). Многие приложения помечают данные специальными префиксами. Например, служба цифровых нотариусов (Proof of Existence) применяет 8-ми байтный префикс DOCPROOF (0x44f4350524f4f46 в шестнадцатеричном формате).

Разблокирующих сценариев для выходов, содержащих инструкцию OP_RETURN, не существует, т.е., такие выходы доказуемо не могут быть растрачены. Поэтому их не нужно держать в пуле нерастраченных выходов. Балансы транзакций с OP_RETURN-выходами обычно нулевые. Вряд ли кому-то придет в голову навсегда и безвозвратно загнать в блокчейн некоторую сумму биткоинов.

Также запрещено указывать в качестве входа транзакции выход, содержащий инструкцию OP_RETURN. Такая транзакция сразу же будет помечена как недействительная любым программным обеспечением в платформе Биткоин.

Согласно стандарта сети Биткоин транзакция может иметь только один выход с инструкцией OP_RETURN. При этом остальные выходы транзакции могут быть любых других типов.

Pay-to-Script-Hash (P2SH) сценарии

Модель Pay-to-script-hash (P2SH) была введена в 2012 году в качестве нового типа транзакций, существенно расширяющего спектр поддерживаемых платформой финансовых схем. Появилась возможность использовать сложные сценарий транзакций. Эту модель часто путают с мульти-подписью, однако сфера применения P2SH-скриптов гораздо шире. Если отталкиваться только от скриптов обременения, так вариабельность условий обременения вообще становится практически безграничной. Достичь это позволило весьма элегантное решение. Средства отправляются на адрес скрипта. При этом что из себя представляет скрипт заранее никак не оговаривается – он может быть любым. Еще одна особенность этой модели – в поле транзакции scriptPubKey (где у нас обычно размещаются условия обременения) ограничения в виде скрипта больше не фигурируют. Акцент смещается в сторону пользователя, который впоследствии осуществит трату средств. Именно ему придется представить скрипт в качестве доказательства своего права на заблокированные в выходе P2SH-транзакции средства. Согласитесь, логика проверки по сравнению с предыдущими моделями иная. Очень мощное с точки зрения масштабируемости функционала платформы решение.

Итак, в P2SH-транзакции биткоины традиционно блокируются в скрипте, но сам скрипт не добавляется в выход транзакции (в поле scriptPubKey). Вместо этого, скрипт обременения хешируется. Полученный дайджест никак не может быть использован для восстановления исходного скрипта. Однако, имея на руках исходный скрипт, можно легко получить точно такой же дайджест путем повторного хеширования скрипта. Хеш сценария — это и есть та информация, которая включается в выход транзакции (в поле scriptPubKey). Для того, чтобы такой выход в следующей транзакции, недостаточно просто выполнить условия, определенные в сценарии, так как ноды сети Биткоин видят только дайджест хеша скрипта и поэтому о самом скрипте им ничего не известно. Следовательно, ноды не могут удостовериться в том, что условия, определенные в комбинированном сценарии выполнены. То есть, они не могут подтвердить транзакцию. Поэтому, чтобы разблокировать и растратить средства, транзакция должна включать в себя, вместе с условиями, определенными в скрипте, и сам скрипт. В этом случае ноды сети Биткоин, вычислив хеш приложенного сценарий, могут сравнить его с хешем, размещенным в выходе предыдущей транзакции. В случае совпадения нодам остается только проверить правильность выполнения условий, определенных в сценарии, после чего транзакция признается действительной. Рисунок 3.6 иллюстрирует изложенный процесс.

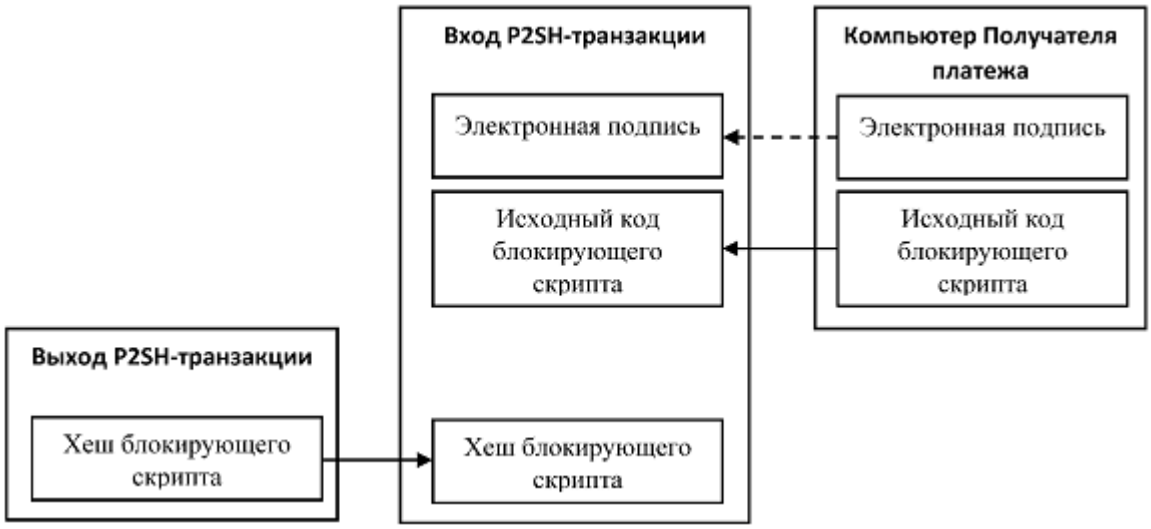

Рис. 3.6. Трата средств, заблокированных на выходе P2SH-транзакции

Таким образом, P2SH-модель по сути реализует следующее финансовое предписание "заплатить при предъявлении скрипта, соответствующего данному хешу".

Чем же модель P2SH отличается от мульти-подписи? Рассмотрим простой пример схемы с корпоративным управлением доступа к средствам, обеспечивающим высокую степень безопасности. Пусть какая-то компания использует механизм мульти-подписи для всех платежей, поступивших от клиентов. Для конкретики примем, что разблокировать средства можно при наличии подписей минимум двух владельцев компании. Всего компанией владеют пять человек.

Комбинированный сценарий траты средств в этом случае выглядит следующим образом:

OP_0 <Signature A> <Signature B> 2 < pubKey A> < pubKey B> < pubKey C> < pubKey D> < pubKey E> 5 OP_CHECKMULTISIG

Недостаток такого сценария очевиден – он слишком длинный. Мы помним, что в распределенных системах следует экономить на каждом байте блокчейна. К тому же вторая часть сценария (скрипт scriptPubKey) должен предварительно быть доведен до каждого клиента, чтобы они смогли его использовать в своих платежных транзакциях. Организационно это не самое удобное решение. Клиенты должны пользоваться специальными кошельками, быть в курсе процедуры создания мульти-подписных транзакций и платить повышенную комиссию майнерам за обработку огромных транзакций (из-за большой длины открытых ключей, которых в блокирующем скрипте стало намного больше, чем в традиционном P2PKH-выходе). Даже не будем вспоминать о дополнительной нагрузке на оперативную память каждой полной ноды, в которой должны храниться все подобные нерастраченные выходы (в пуле UTXO). Все перечисленные проблемы серьезно осложняют применение сложных скриптов обременения на практике.

Pay-to-script-hash модель как раз и была разработана для решения указанных проблем.

В P2SH-транзакциях, блокирующий скрипт заменяется процедурой проверки хеша скрипта, называемого погашающим, поскольку его код будет представлен системе в момент траты выхода (погашения), а отнюдь не в качестве предварительно записанных условий обременения. В таблице 3.4 представлены форматы скриптов для сценария с мульти-подписью и P2SH-схемы.

Структура входа транзакции	
Сценарий с мульти-подписью	P2SH-сценарий
Блокирующий скрипт:	Погашающий скрипт:
2 PubKeyA PubKeyB PubKeyC PubKeyD PubKeyE 5 OP_CHECKMULTISIG	2 PubKey A PubKey B PubKey C PubKey D PubKey E 5 OP_CHECKMULTISIG
Разблокирующий скрипт:	Блокирующий скрипт:
Signature A Signature B	OP_HASH160 <20-байтный хеш погашающего скрипта> OP_EQUAL
	Разблокирующий скрипт:
	Signature A Signature B Погашающий скрипт

Биктоин-адрес формата P2SH всегда начинается с 3, а не 1, как в P2PKH-адресах. Это связано с тем, что P2SH-адреса перед кодированием по стандарту base58check дополняются префиксом байта версии, равным 0x05 (в шестнадцатеричном формате), а не 0x00 (в шестнадцатеричном формате), характерным для P2PKH-адресов.

А теперь вместо мнемонических обозначений в блокирующий скрипт подставим реальные открытые ключи (520-разрядные двоичные числа, начинающиеся с 0x04):

2

04C16B8698A9ABF84250A7C3EA7EEDEF9897D1C8C6ADF47F06CF73370D74DCCA01CDCA79D

CC5C395D7EEC6984D83F1F50C900A24DD47F569FD4193AF5DE762C58704A2192968D8655D6A9
35BEAF2CA23E3FB87A3495E7AF308EDF08DAC3C1FCBFC2C75B4B0F4D0B1B70CD2423657738
C0C2B1D5CE65C97D78D0E34224858008E8B49047E63248B75DB7379BE9CDA8CE5751D16485F4
31E46117B9D0C1837C9D5737812F393DA7D4420D7E1A9162F0279CFC10F1E8E8F3020DECDBC3
C0DD389D99779650421D65CBD7149B255382ED7F78E946580657EE6FDA162A187543A9D85BAA
A93A4AB3A8F044DADA618D087227440645ABE8A35DA8C5B73997AD343BE5C2AFD94A504375
2580AFA1ECED3C68D446BCAB69AC0BA7DF50D56231BE0AABF1FDEEC78A6A45E394BA29A1
EDF518C022DD618DA774D207D137AAB59E0B000EB7ED238F4D800 5 OP_CHECKMULTISIG

Пугает, правда? Зато в P2SH-сценарии весь этот ужас представляется 20-ти байтным дайджестом - результатом хеширования блокирующего скрипта с помощью последовательного применения алгоритмов SHA-256 и RIPEMD-160, т.е., RIPEMD-160(SHA-256(<код блокирующего скрипта>)). Для нашего конкретного случая это всего-то:

54c557e07dde5bb6cb791c7a540e0a4796f5e97e

Выход P2SH-транзакция будет обременен с помощью следующего блокирующего скрипта:

OP_HASH160 54c557e07dde5bb6cb791c7a540e0a4796f5e97e\ OP_EQUAL

Это все, что нужно знать клиенту для проведения платежа в адрес рассматриваемой в примере компании. Размер такой транзакции не будет превышать размеров обычных транзакций (по крайней мере по причине раздутого скрипта scriptPubKey) и проведение платежа не потребует от клиента затрат на повышенную комиссию майнерам.

Как же можно снять обременение с такого нерастраченного выхода? Ведь биткоин-адреса получателя средств, в привычном для нас с Вами понимании, в P2SH-сценарии больше нет. Необходимо создать вход, в котором представлен оригинальный код погашающего скрипта (того самого, хеш которого использовался при обременении средств) и подписи двух или более владельцев компании - в качестве удовлетворения условий обременения выхода - погашающего скрипта. Например:

< Signature A> < Signature B> < 2 <PubKey A> <PubKey B> <PubKey C> <PubKey D> <PubKey E> 5 OP_CHECKMULTISIG >

Комбинированный сценарий проверки притязаний на заблокированные средства, выраженных приведенным выше входом транзакции, выполняется в два этапа. Во-первых, погашающий сценарий проверяется разблокирующим скриптом на предмет совпадения хеша:

2 <PubKey A> <PubKey B> <PubKey C> <PubKey D> <PubKey E> 5 OP_CHECKMULTISIG OP_HASH160 54c557e07dde5bb6cb791c7a540e0a4796f5e97e OP_EQUAL

Если проверочный сценарий заканчивается успешно (в стеке окажется единственное значение ИСТИНА), переходим ко второму этапу, на котором мы фактически имеем дело с обычной процедурой проверки комбинированного сценария, включающего модифицированный разблокирующий скрипт (<Условия снятия обременения> + <Погашающий скрипт>):

OP_0 < Signature A> < Signature B> 2 <PubKey A> <PubKey B> <PubKey C> <PubKey D> <PubKey E> 5 OP_CHECKMULTISIG

Далее все происходит точно также как и в случае проверки обычной транзакции.

Pay-to-script-hash адреса

Последним аспектом P2SH-модели, на котором мы остановимся, является возможность использовать хеш скрипта в качестве адреса (впервые такая возможность была определена в BIP0013 - Bitcoin Improvement Proposal "Address Format for pay-to-script-hash"). Адресом в P2SH выходе транзакции является 20-ти байтный хеш сценария в кодировке Base58Check, точно так же, как традиционный биткоин-адрес - это 20-ти байтный хеш открытого ключа в кодировке Base58Check. Шестнадцатеричные P2SH-адреса используют префикс версии "5". Это значит, что после кодирования в формате Base58Check такие адреса будут начинаться с 3. Именно по этой тройке вначале все участниками сети Биткоин легко смогут распознать P2SH-адреса. Например, упомянутый выше шестнадцатеричный P2SH-адрес 54c557e07dde5bb6cb791c7a540e0a4796f5e97e в формате Base58Check будет записан так:

39RF6JqABiHdYHkfChV6USGMe6Nsr66Gzw

P2SH-адреса могут распространяться традиционными способами аналогично биткоин-адресам и аналогично последним содержат всю необходимую информацию для создания транзакций. Клиенты нашей компании из примера могут воспользоваться практически любым Биткоин-кошельком для проведения платежей по описанной выше, достаточно сложной схеме. Единственное, что им для этого нужно 20-байтный шестнадцатеричный хеш скрипта, воспринимаемого как некоторый абстрактный адрес, или его 34-символьная запись в формате Base58Check. Ну и без некоторого числа нерастраченных биткоинов они конечно тоже заплатить не смогут.

В модели pay-to-script-hash в качестве погашающего скрипта разрешено использовать любой не содержащий ошибок сценарий, включая стандартные типы: P2PK, P2PKH, мульти-подпись. Запрещается применять сценарии типа OP_RETURN (как OP_RETURN-скрипт не может быть погашен по определению) и рекурсивные обращения, т.е., те же скрипты P2SH.

Поскольку погашающий скрипт изначально в сети не представлен (блокирующий скрипт содержит только его хеш), надо быть особенно внимательным с P2SH-транзакциями. Любая ошибка в сценарии обременения, во-первых, будет обнаружена только при попытке разблокировать денежные средства, а во-вторых, уже никак не сможет быть исправлена. И биткоины с таким обременением для владельца окажутся навсегда потерянными.

Обновление Segregated Witness

Цель: Сформировать представление о сути и задачах, выполняемых обновлением Segregated Witness (SegWit).

В 2017 году был активирован протокол Segregated Witness (SegWit) – самый крупный на сегодняшний день софтварный апгрейд в сети Биткоин, целью которого было дальнейшее масштабирование, снижение транзакционных комиссий и увеличение лимита блоков.

Для понимания проблемы, которую решает SegWit, необходимо внимательнее рассмотреть, как устроены биткоин-транзакции. Состоят они из двух главных частей: основных данных о транзакции, к которым, например, относится информация о том, какие именно перемещаются монеты и в каком направлении, и, так называемого, "свидетеля" (witness). Речь идет о части кода (которую называют scriptSig, witness или же разблокирующий скрипт) с данными электронной подписи, служащей доказательством того, что владелец монет действительно хочет их потратить. Именно в этих данных заключена небольшая проблема, получившая название пластичности транзакции. Ее суть состоит в том, что подписи могут быть изменены даже после их создания и при этом они остаются действительными. Это означает, что те, кто транслируют транзакцию, или майнеры, которые включают ее в блок, могут изменить внешний вид этой транзакции, или, если говорить более предметно, ее идентификатор. Проблема на первый взгляд кажется не столь значимой: транзакции остаются действительными, а монеты перемещаются на нужные адреса. Сложность возникает при создании новых транзакций, которым необходимо знать идентификатор прежней транзакции, на который они опираются. Это в свою очередь значительно затрудняет создание поверх сети Биткоин определенных протоколов второго слоя, например, двунаправленных платежных каналов.

Разберемся в этих проблемах более основательно. Напомним, что идентификатор транзакции (он же хеш транзакции - txid) – это 64-разрядное шестнадцатеричное число (256 бит), которое представляет собой идентификационный номер, уникальный для каждой транзакции в блокчейне Биткоин. Такие же идентификаторы используются и в других платформах.

Например, идентификатор транзакции в сети Биткоин выглядит так:

a1075db55d416d3ca199f55b6084e2115b9345e16c5cf302fc80e9d5fbf5d48d

Идентификатор транзакции в платформе Ethereum выглядит практически также:

b4bc263278d3f77a652a8d73a6bfd8ec0ba1a63923bbb4f38147fb8a943da26d

Дискуссии о решении проблемы пластичности транзакций посредством "отделения" данных о подписи от остальных данных транзакции начались еще в январе 2012 года. Среди прочих в них принимали участие разработчики Bitcoin Core Расселл О'Коннор, Мэтт Коралло, Люк Дэш-младший и Грегори Максвелл, а также модератор форума Bitcointalk Theymos. Однако подходящего решения тогда так найдено и не было.

Segregated Witness (сокращенно SegWit) — обновление платформы Биткоин, призванное решить две основных проблемы данной криптовалюты. Речь идет об обеспечении пластичности транзакций блокчейна Биткоин, а также об увеличении пропускной способности системы. Аналогичное по сути техническое решение реализовано и для целого ряда альткоинов, таких как

Litecoin, DigiByte, Groestlcoin и Vertcoin. Обновление позволяет уменьшить размер транзакций, делая блоки более вместительными, а также снимает проблему пластичности ("изменчивости" transaction malleability), что очень важно для технологий наподобие платежных каналов или лайтнинга, полагающихся на строение транзакции сети Биткоин.

SegWit является "софт форком" (т.е. изменения сделаны на уровне программного обеспечения клиентов и не затронули сам блокчейн) и позволяет сети функционировать в прежнем режиме. При этом, он предусматривает изменение структуры, используемой для хранения информации в блоке и механизма валидации транзакций полными нодами. Для достижения цели данные, инкапсулирующие электронные подписи и блокирующие скрипты выделены в обособленную структуру под названием "отдельный свидетель" (англ. segregated witness). Поскольку размеры каждой транзакции стали существенно меньше, основные блоки способны вместить большее число транзакций. Одновременно такой механизм обеспечивает инвариантность идентификатора одной и той же транзакции в рамках всего жизненного цикла.

Форк (от англ. Fork - вилка) - процесс, в результета которого создается альтернативная успешная версия блокчейна. Подобное может произойти умышленно - в случае сосредоточения под контролем группы майнеров вычислительных мощностей, достаточных для влияния на консенсус сети (атака 51%), случайно (когда нескольким майнерам одновременно удается записать новые блоки, что приведет к ветвлению блокчейна, или в результате ошибки системы), а также целенаправленно при решении команды разработчиков расширить функционал системы (или исправить ранее допущенные ошибки) в новых версиях клиентского программного обеспечения.

Форк признается успешным, если его ветвь становится цепочкой блоков с максимальной суммарной сложностью доказательств консенсуса. В этом случае альтернативная ветка блокчейна отвергается и в дальнейшем в системе не развивается.

Также форком называют изменения протокола криптовалюты, приводящие к созданию двух различных версий блокчейна с единой историей.

Часто форком называют новую криптовалюту, основанную на протоколе действующей платежной платформы. Например, большинство известных криптовалют являются форками сети Биткоин.

Проблема масштабируемости

В 2010 Сатоши Накамото (по крайней мере, так себя называл создатель (группа разработчиков) Биткоина) ограничил размер блока одним мегабайтом. Такая мера благотворно повлияла на совместимость узлов сети и позволила существенно уменьшить эффективность DDoS-атак. Однако, одновременно с этим снизилась максимальная пропускная способность платформы, в среднем до 3-7 транзакций в секунду. В дальнейшем это ограничение оказало очень негативное влияние на возможность масштабирования сети. Рост числа участников и количества переводов привел к заметному увеличению времени задержки платежа — некоторые транзакции "зависали" в

подвешенном состоянии на несколько дней. Синхронно с задержками выросла комиссия за обработку транзакций, взимаемая майнерами. Данное обстоятельство резко снизило возможности применения платформы Биткоин для проведения мелких платежей – ниши, которая изначально предполагалась идеальной для использования криптовалюты.

Преодолеть негативные последствия разработчики пытались не один раз. Самым очевидным и известным из предложенных решений является увеличение размера блока. Сразу несколько форков Биткойна, такие как Bitcoin XT, Bitcoin Classic, Bitcoin Unlimited, Lightning Bitcoin и самый успешный Bitcoin Cash, пошли именно по этому пути.

Пластичность транзакции

Еще одной проблемной точкой сети Биткоин является, так называемая, пластичность транзакции. Обычная платежная транзакция содержит электронную подпись, которая позволяет другим участникам сети проверить правомочность намерений создателя транзакции. Подпись формируется на основе закрытого ключа для каждой транзакции, что позволяет впоследствии легко обнаружить малейшее изменение контента транзакции. Любое изменение критичных данных транзакции вызывает изменение ее идентификатора. На самом деле, можно изменить транзакцию, сохранив при этом ее статус как действительной (например, можно без проблем добавить дополнительные служебные константы в подпись, не меняющие смысла сценария проверки транзакции). Однако, подобные незначительные правки приводят к кардинальному изменению ее идентификатора. В результате, модифицированная транзакция будет считаться совершенно новой, но при этом, сможет пройти проверку другими узлами сети.

Критичной является ситуация, когда модифицированная транзакция оказывается в блокчейне раньше исходной или в дальнейшем окажется в его более длинной ветви. В этом случае оригинальная транзакция будет признана недействительной, поскольку ее вход ссылается на уже использованный выход (так же будут признаны недействительными все транзакции, ссылающиеся на нее). Это создает множество проблем, в виду того, что большое число систем проверяет состоятельность финансовых сделок по их идентификатору. Также это обстоятельство существенно затрудняет внедрение технических решений более высокого уровня, основанных на блокчейне платформы Биткоин.

Атака "дней рождения"

Сценарии с мульти-подписью теоретически могут быть атакованы. Если злоумышленник владеет хотя бы одним ключом из представленного в транзакции списка, то с учетом коллизии хеша он может уменьшить число комбинаторных вариантов до 280, перебор которых представляется вполне осуществимой задачей для современных вычислительных систем.

Указанные выше проблемы обострились в последнее время в связи с ростом числа пользователей и развертыванием поверх сети Биткоин большого числа распределенных приложений. Разработчики и раньше пытались с ними бороться, но, пожалуй, только обновление

Segregated Witness реально предоставляет механизм выхода из тупиковой ситуации. Хотя бы на какое-то время.

Подписи

Подписи – это криптографический прием, в котором для вычисления уникальной последовательности чисел используется закрытый ключ в сочетании с любыми другими данными. Соответствующий открытый ключ может использоваться для верификации того, что данная подпись была создана с использованием данного приватного ключа. Таким образом, подписи доказывают, как владение закрытым ключом, так и подтверждение определенной части данных владельцем закрытого ключа – и всё это без его разглашения. В случае платформы Биткоин закрытые ключи обычно используются для подписи данных транзакции (за исключением входов, scriptPubKeys, и некоторых других данных). В результате подпись и открытый ключ, по которому расходуются биткоины добавляются к транзакции в поле входа. Это доказывает то, что владелец ключа действительно желал совершить транзакцию, и гарантирует, что ее нельзя было подделать.

Далее все данные транзакции, включая на этот раз и входы, совместно хешируются, полученный результат впоследствии служит её идентификатором (txid транзакции). Если транзакция в конце концов попадает в блок, майнер хеширует ее txid вместе с txid другой транзакции, получая новый хеш. Этот хеш также хешируется, на этот раз с хешем двух других txid транзакций. Процесс продолжается, пока не останется всего один хеш. Эта система хешей называется деревом Меркла, а оставшийся хеш – корнем Меркла. Корень объединяется с дополнительными данными, которые используются для идентификации конкретного блока, формируя его заголовок. Хеш заголовка блока в итоге включается в заголовок следующего блока, образуя цепочку связанных блоков. Блокчейн криптовалюты является неизменяемым, поскольку редактирование части любой транзакции задним числом поменяет txid транзакции, что приведёт к изменению заголовка блока – однако такой блок уже не будет соответствовать установленным требованиям. И поскольку заголовок блока влияет на структуру последующих заголовков блока, они также не будут им соответствовать.

SegWit основан на концепции сайдчейнов, разработанной компанией Blockstream, и дополняет идею разработчика ядра платформы Биткоин Люка Дэша. Общая концепция была разработана спустя несколько месяцев в сотрудничестве с разработчиками ядра Грегори Максвеллом и Эриком Ломброзо.

С точки зрения нод, которые не используют SegWit (условно назовём их ортодоксальными) некоторые вновь созданные выводы могут начать использовать странный тип scriptPubKeys. Странность заключается в том, что их едва ли можно считать блокирующими или запирающими. Именуемые как (Anyone Can Spend - "тратят все"), эти выходы в целом заявляют, что подписи им не нужны. Кроме того, в них еще и присутствует совершенно бессмысленный (но не запрещенный протоколом) текст.

Старые ноды сочтут эти транзакции бессмысленными. Они будут уверены в том, что любой пользователь может создать новый scriptSig, высвобождая эти выводы – а это значит, что они практически полностью незащищены. В то же время, старые ноды технически не смогут не принимать новые транзакции. Текст ScriptPubKeys покажется им не имеющим смысла, но технически вполне допустимым. Поэтому ортодоксальные ноды определят в итоге транзакции как действительные, и ретранслируют их дальше по сети.

А вот ноды с SegWit (назовем их новыми) поведут себя немного иначе. Текст ScriptPubKeys приобретет для них вполне конкретный смысл и будет идентифицирован как весьма специфический тип выхода.

Подобно "доперестроечным" выходам, эти новые выходы будут нуждаться в валидных подписях для высвобождения биткоина – однако в отличие от них, для этого им не нужна будет подпись, включённая в scriptSig следующей транзакции. Вместо этого подпись должна содержаться в ранее отсутствующей части транзакции – SegWit.

SegWit по сути является параллельной структурой данных, содержащей подписи и некоторые другие данные. Главное здесь то, что SegWit полностью игнорируется старыми нодами, но признается новыми. К тому же, теперь подписи не хешируются совместно с другими частями транзакции для создания txid.

Таким образом, и ортодоксальные, и новые ноды будут считать транзакции с SegWit валидными. Ортодоксальные ноды признают их действительными, поскольку с их точки зрения им вовсе не нужны подписи, а новые – потому, что нужная подпись находится в SegWit. Поскольку и те, и другие ноды хешируют данные транзакции, получая один и тот же идентификатор, консенсус по компоновке блоков будет достигнут, а, следовательно, к состоянию блокчейна также не будет возникать никаких вопросов.

Есть, однако, небольшая проблема: если подписи не влияют на компоновку блокчейна, он уже не может являться доказательством того, что в транзакции включены корректные подписи.

Чтобы подписи всё равно включались в блокчейн, майнер с SegWit выполняет дополнительное действие – создаёт дерево Меркла не только из транзакций, но и из SegWit, причём последнее полностью соответствует дереву транзакций. Корень дерева SegWit включается в поле ввода транзакции coinbase. Таким образом корень дерева SegWit меняет данные транзакции coinbase, её идентификатор, а значит и заголовок – в результате меняется вся компоновка блокчейна. Обновление Segregated Witness позволяет удалить подписи из транзакций биткоина, сохранив его неизменяемость и не нарушая ни одного из принятых правил консенсуса.

Обновление Segregated Witness

Суть обновления Segregated Witness состоит в образовании одноименной динамической структуры вне основного блока и переносе в нее подписей транзакций. Такой механизм позволяет значительно уменьшить нагрузку на блок, оказываемую каждой транзакцией, поскольку на

электронную подпись приходится более половины ее размера. Автоматически решается и проблема пластичности транзакций, поскольку подписи больше не влияют на идентификатор транзакции.

При этом, безусловно, меняется процесс проверки транзакции. Теперь ноде нужно загрузить расширенный блок (основной блок + данные "отделенного свидетеля"). О своей готовности обработать расширенную структуру блока узел специально извещает соседей. Узлы, не поддерживающие SegWit, продолжают принимать стандартные блоки размером в 1МБ, полагая, что транзакции не требуют доказательства владения в виде подписи.

Контейнер подписей для организации связи с основной цепочкой использует дерево Меркла, корень которого хранится в заголовке блока. Для всех электронных подписей и транзакций рассчитываются значения хеш-функции, которые заносятся в дерево Меркла. Суммарный хеш подписей присоединяется к хешу первой транзакции (coinbase-транзакции) в дереве Меркла.

Расширенный блок теоретически ограничен 4 мегабайтами, но фактически максимальный размер блока составляет чуть менее 2 мегабайт.

В SegWit для защиты кошельков с мульти-подписью вместо модели P2SH используются сценарии P2WSH, защищенные хеш-функцией SHA-256. Данное обстоятельство усложняет атаку "дней рождения".

На самом деле, Segregated Witness меняет не только структуру транзакции, но и строение ее выходов. Это, однако, не значит, что в одной и той же транзакции не могут быть потрачены как традиционные неизрасходованные выходы (UTXO), так и новые SegWit типа — просто первые будут искать "доказательство" внутри входа (поле scriptSig), а вторые — снаружи. Также вводится отдельный идентификатор wtxid — он хеширует не только транзакцию, но и всю witness часть, так что если транзакция передается без witness данных, то ID равен wtxid.

Так как Segregated Witness все-таки является софт-форком, его обновления могут быть проигнорированы, а значит более старые системы должны как-то обрабатывать SegWit выходы. Дело в том, что для старых нод или кошельков эти выходы выглядят как доступные всем, то есть они могут быть потрачены с пустой подписью, что все еще допустимо. Принявшие обновление ноды и кошельки конечно же будут искать все подписи вне пространства входов в специальном поле witness.

Pay-to-Witness-Public-Key-Hash

Теперь давайте взглянем на примеры транзакций и на то, как они изменятся с обновлением Segregated Witness. Начнем со стандартной Pay-to-Public-Key-Hash (P2PKH) транзакции. Нас интересуют выходы, а именно их поля "scriptPubKey". Типичный блокирующий скрипт выглядит следующим образом:

OP_DUP OP_HASH160 <Public-Key-Hash> OP_EQUALVERIFY OP_CHECKSIG

Модель Segregated Witness низводит его до уровня "Anyone Can Spend":

0 <Public-Key-Hash>

SegWit-выход существенно проще традиционного — он состоит из двух значений, которые будут помещены в стек сценария. Как уже упоминалось, для старых версий клиентов платформы Биткоин такой выход будет виден как доступный любому, поскольку он не требует подписи. А вот для обновленного клиента первое число интерпретируется как номер версии, а второе как аналог запирающего скрипта (witness program). На самом деле, здесь должен использоваться хеш сжатого публичного ключа, об этом мы расскажем немного позже.

Теперь давайте сравним транзакции, в которых выход будет потрачен. В традиционном случае это выглядело бы так:

```
[...]
"Vin" : [
 {
"txid": "8adbca5e652c68f8f3c30ac658115bc4af395d0cc7e6beaea18168295c29d011",
"vout": 0,
"scriptSig": "<Signature> <Public-Key>"
 }
]
[...]
```

Для траты SegWit-выхода, транзакция должна иметь пустое поле sriptSig и содержать все подписи в отдельном месте:

```
[...]
"Vin" : [
 {
"txid": "8adbca5e652c68f8f3c30ac658115bc4af395d0cc7e6beaea18168295c29d011",
"vout": 0,
"scriptSig": ""
 }
]
[...]
"witness": "<Signature> <Public-Key>"
[...]
```

Несмотря на то, что ортодоксальные клиенты могут обрабатывать SegWit-транзакции, они не могут тратить их выходы, так как они просто не знают, как это сделать: кошелек старого типа попытается потратить SegWit-выход с пустой подписью, однако эта транзакция на самом деле не будет считаться действительной (ноды, поддерживающие Segregated Witness, ее не пропустят). В частности, из этого следует, что отправитель, как минимум, должен убедиться в том, что кошелек получателя реализует SegWit.

Согласно BIP1432 (Transaction Signature Verification for Version 0 Witness Program) в настоящее время выходы должны создаваться с помощью хеша сжатого открытого ключа. Если выход будет создан на основе обычного адреса или несжатого открытого ключа, его нельзя будет потратить.

Pay-to-Witness-Script-Hash

Следующим важнейшим типом транзакции является P2SH-модель. Чтобы потратить выход P2SH-транзакции нужно предоставить погашающий скрипт и условия траты, определенные этим скриптом. Используя такой подход можно отправлять биткоины на адрес, защищенный способом, о котором нам вообще ничего не известно, а также сильно экономить место — в случае, например, кошелька с мульти-подписью блокирующий скрипт был бы действительно большим, если бы мы хранили в нем все "замки" полностью, а не только хеш скрипта.

Если вспомнить пример, ранее нами уже разобранный, то традиционный блокирующий скрипт для случая с мульти-подписью, требующей наличия 2-ух подписей из 5-ти выглядит так:

OP_HASH160 54c557e07dde5bb6cb791c7a540e0a4796f5e97e OP_EQUAL

Для траты средств, нужно предоставить погашающий скрипт, который собственно определяет необходимость предоставления 2-х подписей из 5-ти возможных, а также любые 2 подписи из списка и все это должно содержаться во входе транзакции:

[...]

"Vin" : [

"txid": "8adbca5e652c68f8f3c30ac658115bc4af395d0cc7e6beaea18168295c29d055",

"vout": 0,

"scriptSig": "<SigA> <SigB> <2 PubA PubB PubC PubD PubE 5 CHECKMULTISIG>",

]

Рассмотрим технику реализации этой модели в системах с поддержкой Segregated Witness.

Начнем с блокирующего скрипта выхода:

0 9592d601848d04b172905e0ddb0adde59f1590f1e553ffc81ddc4b0ed927dd73

Как и в предыдущем случае блокирующий скрипт намного проще. Здесь первое значение является номером версии, а второе — это 32-х байтный SHA-256-дайджест погашающего скрипта (witness program).

Обратим внимание на то, что здесь хеш 32-х байтный. Сделано это для того, чтобы можно было по длине хеша легко отличить witness-программы для P2WPKH /20 байт RIPEMD-160(SHA-256(script))/ и P2WSH /32 байта SHA-256(script)/.

Транзакция, способная потратить этот выход, выглядит следующим образом:

[...]

"Vin" : [

"txid": "8adbca5e652c68f8f3c30ac658115bc4af395d0cc7e6beaea18168295c29d077",

"vout": 0,

"scriptSig": "",

]

[...]

"witness": "<SigA> <SigB> <2 PubA PubB PubC PubD PubE 5 CHECKMULTISIG>"

[...]

Можно ли использовать Segregated Witness в случае, когда один из партнеров не имеет обновленного кошелька? Оказывается, можно. Скажем, восприимчивый к инновациям владелец кофейни Сергей уже обзавелся кошельком с поддержкой Segregated Witness. В то время как у его постоянной клиентки Марины старый, ортодоксальный кошелек. Разумеется, можно прибегнуть к транзакции стандартного типа. Однако, Сергей любит все новое, и оба очень хотят сэкономить на комиссионных майнерам.

В такой ситуации Сергей может создать P2SH сценарий, содержащий SegWit-скрипт. Программное обеспечение кошелька Марины опознает хеш этого сценария как самый обычный P2SH-адрес, что позволит ей без каких-либо проблем отправить туда средства. В свою очередь Сергей так же легко сможет потратить выход транзакции Марины, используя SegWit- транзакцию.

Таким образом, оба типа SegWit-транзакций и P2WSH, и P2WPKH могут быть реализованы в рамках P2SH-модели.

P2SH(P2WPKH) сценарии

Точно также любой P2WSH скрипт может быть реализован внутри P2SH. Возьмем multisig скрипт 2 из 5, рассмотренный ранее. Все шаги будут практически идентичны случаю P2SH(P2WPKH):

Начинаем снова с создания witness- программы:

0 9592d601848d04b172905e0ddb0adde59f1590f1e553ffc81ddc4b0ed927dd73

Первое число — версия, второе число — 32-х байтный SHA-256 хеш нашего скрипта мульти-подписи. Далее шаги повторяются — находим RIPEMD-160 хеш от witness-программы и преобразуем в обычный P2SH-адрес. Для использования выхода, отправленного на этот адрес, в scriptSig нужно записать п-программу, а все подписи и полный скрипт мультиподписи в поле witness.

Снижение размера комиссионных

За счет скидки на хранение witness-данных SegWit-транзакции обходятся дешевле, по сравнению с традиционными. Фактически было изменено само понятие "размера" для SegWit-транзакций. Вместо традиционного подхода для них используется концепция "виртуального размера" (virtual size) — все данные, хранящиеся в witness, учитываются с коэффициентом 0.25, что также позволяет разместить в блоке больше транзакций. Рассмотрим на примере. Пусть у нас есть традиционная транзакция размером в 200 байт. В блок размера 1 MB поместится 5000 таких

структур данных. Теперь возьмем эквивалентный ей SegWit вариант, где примерно 120 байт это witness-данные. Тогда ее виртуальный размер = 80 + 0.25 * 120 = 110 и теперь уже 9090 таких транзакций поместятся в тот же самый блок. Также при комиссии, скажем, в 40 сатоши/байт для первой транзакции мы получим комиссию в 8000 сатоши, а для SegWit структуры - 4400 сатоши, что практически в два раза меньше.

Версия скрипта

Каждый блокирующий скрипт содержит байт, отвечающий за версию скрипта. Использование механизма версий позволяет вносить дополнения и правки (изменения в синтаксисе, новые операторы и т.д.) в виде софт-форков.

Оптимизация процессов проверки электронных подписей

В обновлении Segregated Witness были оптимизированы алгоритмы обработки электронных подписей (OP_CHECKSIG, OP_CHECKMULTISIG и т.д.). До реализации обновления число хеш-вычислений увеличивалось квадратично относительно количества подписей. В системе, использующей Segregated Witness, сложность алгоритмов валидации понижена до O(n).

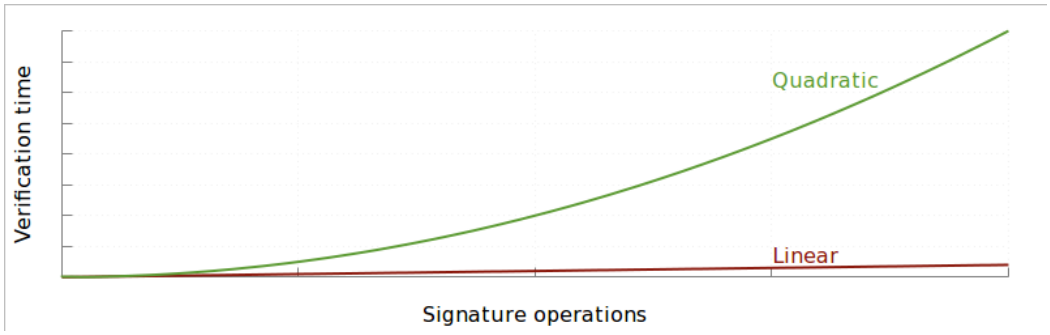

Рис. 3.7. Сравнение времени проверки электронных подписей до (зеленая линия) и после (красная линия) реализации Segregated Witness

Активация

Segregated Witness был предложен Питером Вюлле (Pieter Wuille) в конце 2015 года. Выпуск состоялся в октябре 2016 года — на 6 месяцев раньше запланированного срока. Активация должна была произойти после преодоления 95%-порога участников, сигнализирующих о поддержке обновления. Но некоторые участники сети заявили, что поддержат обновление, только если в него будет добавлено увеличение размера основного блока (китайские пулы могли заблокировать введение SegWit). 23 мая 2017 года майнеры и разработчики подписали Нью-Йоркское соглашение, предполагавшее увеличение размера основного блока до 2 МБ в течение 6 месяцев (это обновление назвали SegWit2x). SegWit был активирован 24 августа 2017 года.

Некоторые альткойны тоже решили реализовать SegWit. Так как многие альткойны основаны на коде платформы Биткоин, это не составило для разработчиков особых трудностей. Первым из них активировал обновление Groestlcoin в январе 2017 года.

Преимущества Segregated Witness:

Совместим с предыдущими версиями программного обеспечения.

Увеличивает количество транзакций в блоке.

Снижает комиссионные сборы.

За счет количества транзакций в блоке общие сборы майнеров могут увеличиться.

Уменьшает время ожидания в очереди.

Способствует масштабируемости платформы Биткоин.

Устраняет пластичность транзакций.

Облегчает разработку и увеличивает эффективность и безопасность дополнительных надстроек (смарт-контракты, Lightning Network и т. д.).

Устраняет проблему квадратичного роста времени проверки транзакций.

Повышает надежность кошельков с мульти-подписью.

Недостатки:

Увеличивает нагрузку на узлы сети.

Несколько усложняется проверка транзакций.

Сборы майнеров могут сократится.

Некоторые участники считают SegWit лишь временной мерой и настаивают на увеличении размера основного блока.

Дополнительная цепочка тоже требует обслуживания, в чем майнеры не особо заинтересованы. В сети Биткоин не предусмотрено вознаграждение за проверку транзакций (в отличие от Dash). Майнеров сдерживает только высокая вероятность майнинга ошибочных блоков при высокой концентрации облегченных клиентов (SPV-нод).

Так как SegWit является софт-форком, обновлены будут далеко не все клиенты, а значит в сети будут находится одновременно два вида нерастраченных выходов, и такие важные изменения как устранение уязвимости идентификаторов транзакций и хеширование за линейное время не будут применены к ортодоксальным выходам, а значит сеть все еще будет уязвима к атакам, основанным на изменяемости идентификаторов транзакций, а также к проблеме квадратичного времени хеширования.

SegWit может уменьшить безопасность сети. Количество нод, проводящих полную валидацию, сильно уменьшится, так как только принявшие SegWit смогут проверять witness часть транзакций.

SegWit не может быть отменен. Если его отменить и откатить все изменения обратно, все SegWit-выходы станут доступными каждому.

SegWit пытается решить все проблемы сразу и, как следствие, огромное количество кода изменено. Это усложняет дальнейшую работу и увеличивает вероятность появления багов, которые будет сложнее устранить.

Перспективы развития

8 ноября 2018 "хард-форк" SegWit2x был отложен на неопределенное время из-за отсутствия консенсуса.

Благодаря обновлению SegWit существенно облегчается разработка и внедрение надстроек, а также увеличивается их безопасность и эффективность. В ближайшее время планируется запуск Lightning Network. Разрабатывается решение для увеличения гибкости смарт-контрактов Merklized Abstract Syntax Tree (MAST), которое также улучшает масштабируемость и повышает конфиденциальность.

Начиная с версии 0.16.0 эталонного клиента Bitcoin Core, опубликованной 15-го февраля 2018 года, программное обеспечение этого кошелька обеспечивает полную поддержку технологии Segregated Witness. Базовая поддержка SegWit появилась еще в версии 0.13, но содержала много недоработок. Начиная с Bitcoin Core 0.16.0, адреса и транзакции SegWit поддерживаются полностью и используются по умолчанию.

В октябре 2018 года число SegWit-транзакций в сети Биткоин превысило 50%. Как следует из рисунка 25, рост SegWit-транзакций хотя и не был стремительным, но проявил высокую стабильность. Спустя всего семь месяцев после реализации обновления (т.е., начиная с марта 2018 года) процент SegWit-транзакций не опускался ниже отметки 30%. На начало 2019 года их доля составляет половину общего пула.

Рис. 3.8. Динамика роста числа транзакций с поддержкой технологии Segregated Witness

Краткие итоги

Транзакции - это специальные структуры данных, фиксирующие процессы передачи ценности (токенов) между участниками криптоплатформы. Ноды проверяют полученные от других узлов по пиринговой сети транзакции и передают их дальше. Жизненный цикл транзакции включает следующие этапы: создание; подписание обычной или мульти-подписью; распространение по сети; проверка майнером и включение в блок; включение блока с транзакцией в блокчейн.

Пиринговая (одноранговая) сеть представляет собой свободное объединение абсолютно равноправных компьютеров. Такая сеть характеризуется высокой степенью отказоустойчивости и

принципиальной возможностью получения необходимой информацию одновременно из разных источников.

Транзакция согласно протокола Биткоин содержит следующие поля: версия; число входов; входы; число выходов; выходы; Locktime.

Концептуальным элементом протокола Биткоин являются неизрасходованные выходы транзакций (Unspent Transaction Output), позволяющие отследить состояния реестра. Каждая очередная транзакция расходует выходы предыдущих и создает новые выходы, которые в свою очередь будут израсходованы последующими транзакциями. Причем каждый выход может использоваться только один раз. Выходы транзакций инкапсулируют определенную сумму и фиксированное обременение, определяющее условие вывода средств. В большинстве случаев, блокирующий скрипт (ScriptPubKey) ассоциирует обременение с адресом получателя платежа. Поэтому вход транзакции должен включать отпирающий сценарий (ScriptSig), удовлетворяющий условиям снятия обременения, установленным данным выходом. Как правило это электронная подпись, доказывающая факт владения биткоин-адресом, фигурирующим в сценарии блокировки.

Особенностью coinbase-транзакций является отсутствие входов.

Кроме стимулирования майнеров, комиссия за транзакцию выполняет важнейшую системную функцию – избавляет сеть Биткоина от спам-транзакций. Стандартная величина комиссионных рассчитывается на основе размера транзакции в килобайтах и не зависит от переводимой суммы.

Проверка валидности транзакции в сети Биткоин осуществляется на основе выполнения скрипта, записанного на специальном языке сценариев, называемым Script. Это Forth – подобный стековый язык программирования с обратной польской нотацией, неполный по Тьюрингу. Важная особенность таких языков - использование стека для передачи параметров между термами, что позволяет очень гибко и просто реализовывать достаточно сложные конструкции.

Разрешенными в платформе Биткоин являются следующие типы сценариев транзакций:

pay-to-public-key-hash (P2PKH);

public-key;

multisignature (с ограничением на 15 ключей);

pay-to-script-hash (P2SH);

выход данных (OP_RETURN).

Суть обновления Segregated Witness состоит в образовании одноименной динамической структуры вне основного блока и переносе в нее подписей транзакций. Такой механизм позволяет значительно уменьшить нагрузку на блок, оказываемую каждой транзакцией, поскольку на электронную подпись приходится более половины ее размера. Автоматически решается и проблема пластичности транзакций, поскольку подписи больше не влияют на идентификатор транзакции.