Атакуем Zcash Протокол Ради Забавы И Выгоды
Whitepaper Version 0.6

Duke Leto + Hush Разработчики [†]

7 марта 2021 г.

Краткий обзор.

Данный документ, впервые, опишет как именно работает **ITM Атака** (связывающая атака против защищённых транзакций) против Zcash Протокола и **Hush** как первую криптовалюту имеющую **Sietch** - смягчение оборонительного характера против данной атаки. Sietch уже работает и запущен, так же проходят этапы улучшения на основе отзывов экспертов. Это не научная статья о несбыточных мечтах. Документ описывает производственный код и сети.

Мы начнем с обзора упоминаний о способе атак метаданных, которые могут быть использованы против Блокчейнов Протокола Zcash. Будет включена стоимость данной атаки и модель угроз. Документ далее опишет "ITM Атаку", являющая конкретным примером нового класса атак метаданных против блокчейнов, который автор опишет как **Metaverse Metadata Attacks**.

Документ далее объясняет Sietch в деталях, что было ответом на подобные атаки. Мы надеемся новые знания и теория помогут криптовалютам увеличить их безопасность против очень хорошо финансируемых врагов, включая национальные государства и компании занимающиеся анализом блокчейнов.

Несколько других проблем конфиденциальности и атак метаданных против Протокола Zcash будут рассмотрены и опубликовы впервые. Идеи в этом документе применяются ко всем криптовалютам которые используют графики транзакций, что можно сказать про всю известную криптовалюту. В особенности, класс атак Metaverse Metadata подходит для всех форков **Bitcoin** (включая Dash, Verge, Zerocoin и их форки), криптовалюта Протокола **CryptoNote** (например Monero) и монеты Протокола **MimbleWimble** (Grin, Beam, т.д), но они здесь не рассматриваются, только детально описываются, как применять методы к их блокчейнам.

> Мы верим в приватность zdust
>
> Если пыль (dust) может атаковать нас, пыль может защитить нас.
>
> - Sietch Mottos

Keywords: анонимность, протокол zcash, криптографические протоколы, zk-SNARKs, утечка метаданных, деанонимизация, электронная коммерция и оплата, финансовая приватность, математика с нулевым разглашением, линкабельность, графики транзакций, защищённые транзакции, анализ блокчейна.

Содержание

[†] hush.is, https://keybase.io/dukeleto, F162 19F4 C23F 9111 2E9C 734A 8DFC BF8E 5A4D 8019

1 Введение

Sietch увеличивает приватность [Zcash] Протокола, где утечка метаданных становится намного сложнее для выполнения и добавляет **индетерминизм**, например [Hush] не действует аналогичным образом при тех же inputs. Это делает имитирование или "распыление" в полной ноде HUSH затруднительным.

Hush перешел на принудительную приватность на Блоке 340000 в Ноябре 2020, обеспечивая высочайший уровень конфиденциальности для пользователей в мире Zcash и напрямую конкурируя с отличными функциями конфиденциальности [Monero] и других монет Протокола [CryptoNote].

2 Анализ Метаданных Блокчейнов Протокола Zcash: Основы

2.1 Концепции и Определения

Этот документ будет связан с **графиками транзакций**, которые мы определяем в традиционном математическом смысле как комплекс нод с набором вершин соединяющих узлы. В криптовалютах это всегда работает как ориентированные графики, поскольку всегда есть неизрасходованные средства, которые становятся потраченными, то есть направление связано с каждой транзакцией. Это направление может быть опредено математически, используя временную метку транзакции. Вовремя транзакции, неизрасходаванные inputs становятся потраченными после транзакции. Outputs несуществуют перед транзакцией и являются потраченными после транзакции.

Значительное количество математической истории было уделено изучению **теории графов**, но которая не была применена в анализе блокчейнов, в основном потому, что еще несколько лет назад не существовало блокчейнов для анализа, как и финансовой выгоды от изучения этой информации. Но все очень резко изменилось.

Недавно мы обнаружили улучшенное программное обеспечение анализа блокчейна который использует "семантически обогащённые" графики транзакций с поисковой системой и расширенными кластероообразованными алгоритмами для создания диаграмм о сложных денежных потоках через большое количество адресов [OBitcoinWhereArtThou].

Этот документ, в большинстве своем, описывает **графики защищённых транзакций** которые называются **ориентированные ациклические графы (DAGs)** где нода представляет собой **транзакцию** с уникальным айди (id) называющиеся **txid**. Поступающие вершины являются inputs, которые были потрачены и уходящие вершины являются новыми outputs, которые были созданы. Полноценная **защищённая** транзакция не раскрывает адрес ни Алисы, ни Боба, ни количество обмениваемых средств, но это разглашает большое количество метаданных на уровне протокола, что не появится в эксплорерах (исследователях) блоков, и что недостаточно хорошо изучено в криптоиндустрии.

Защищённая транзакция имеет как минимум один **защищённый** адрес, который упоминается как zaddr.

Нас волнует только **Zcash Протокол**, который позволяет нам узнать слаженный язык и обозначения для описания новой **ITM Атаки *zaddr*** (связывающая атака) и меры по смягчению воздействия этой атаки. Все методы описанные здесь могут технически быть использованы против открытых (прозрачных) блокчейнов, но это не имеет смысла, поскольку утечка всех метаданных и так уже существует в этих блокчейнах. Новые атаки могут рассматриваться как "сжатие" новой утечки метаданных из zaddrs из тех мест, где никто и не подумал бы смотреть.

Атака таких монет, у которых есть только график транзакций на уровне P2P, но не хранится в блокчейне (например криптовалюта на Протоколе MimbleWimble), становится более сложной и дорогой. Поскольку национальные государства не чувствительны к уровням издержек и очевидно имеют интерес деанонимизировать любые блокчейны, MW монеты не защищены от этих новых атак, описанные в документе. График транзакций по-прежнему существует и поэтому ключевые концепции могут быть применены.

2.2 Типы защищённых транзакций

Существует много типов защищённых транзакций, отражающие сложность прозрачных (незащищённых) транзакций в [Bitcoin] Протоколе. Здесь мы представим конвенцию для описания транзакций и список часто встречаемых транзакций:

- Полностью защищённая транзакция T со сдачей $z \rightarrow (z, z)$
- Полностью защищённая транзакция T без сдачи $z \rightarrow z$
- Защищённая транзакция T с прозрачной сдачей $z \rightarrow (z, t)$
- Незащищённая транзакция T со сдачей $z \rightarrow (t, z)$
- Незащищённая транзакция T без сдачи $z \rightarrow t$
- Защищённая транзакция T без сдачи $t \rightarrow z$
- Защищённая транзакция T с прозрачным получателем и без сдачи $t \rightarrow (z, t)$
- Защищённая транзакция T с прозрачным получателем и со сдачей $t \rightarrow (z, t, t)$
- Защищённая транзакция T с защищённой сдачей (автощит, autoshield) $t \rightarrow (z, z)$

Выше приведены наиболее распространенные транзакции. Теперь предположим, что мы хотим описать транзакцию, которая отправляется на 5 ***zaddrs*** адресов и на 3 прозрачных адреса без сдачи: $z \rightarrow z, z, z, z, z, t, t, t$. Для описания очень крупных транзакций можно использовать индексы: $z \rightarrow z_{52}, t_{39}$.

Более сложные транзакции, такие как $t, t, t \rightarrow z$ возможны, что и есть, скорее всего, защищённая транзакция, созданная z_shieldcoinbase. Необработанные транзакции могут быть настолько сложными, насколько это допустимо, а некоторые могут быть классифицированы как защищёнными и незащищёнными одновременно, например, $t, z \rightarrow t, z$ что разрешено по правилам консенсуса, но на данный момент RPC способ не создает подобную транзакцию ни в одной криптомонете Протокола Zcash, известные авторам данного документа. Даже в этом случае необработанные транзакции могут создавать их и если/когда они появятся они будут сильно выделяться как очень уникальные транзакции.

Индивидуальная сделка T является подграфом полного графа транзакции $T \subset \mathbb{T}$ с одним числом вершин.

2.3 Различия между KMD + HUSH + ZEC

Напоминаем, что проект Komodo был самым первым добытым генезис блоком Протокола Zcash. Сообщество Komodo взяло Zcash исходный код и сначала добыли собственный генезис блок, 13 сентября 2016 [KMDGenesis] перед Генезис блоком Zcash 28 октября 2016 [ZcashGenesis]. Мейннет Hush был запущен сразу после 17 ноября 2016 [HushGenesis] и был форком исходного кода Zcash 1.0.8, а не форком Komodo.

Существует понятие как **Правило Экранирования-Защиты, (Shielding Rule)** в Zcash, что стало одной из самой больших ошибок совершенным проектом способствуя отсутствию приватности. Изначально KMD и ZEC мейннет были очень похожи в том, что они оба разрешили **необязательное** использование zaddrs. Главная особенность в том, что jl777 уже разрабатывал свои дополнения конфиденциальности поверх Протокола Zcash, и он правильно понимал, что принуждение людей к немедленному **экранированию** средств просто поспособствует людей делать это неверно.

В мейннет ZEC, новые добытые coinbase средства должны сначала быть отправлены на zaddr адрес, до того как они могут быть отправлены на прозрачный адрес. По началу казалось хорошей идеей, это "дополняет набор анонимности" заставляя всех переходить на защищённые адреса, увеличивая пул таких адресов. Но не всё то золото, что блестит.

Практический эффект **Правила Экранирования** Zcash способствует заражению пула всех защищённых адресов с утечкой метаданных, особенно с ценовой и временной утечкой метаданных, делая данное Правило фактически бесполезным. В среднем, средства в мейннет ZEC проводят только 1.4 hops ("прыжков-переводов")

в защищённом пуле, что по сути, почти все средства проводят только 1 hop ("прыжок-перевод"), чтобы удовлетворить правило, и сразу же отправить обратно на прозрачный адрес. Очень часто точно такое же количество "входит" и "выходит" в следующем блоке или через один, полностью уничтожая цель zaddrs адресов.

У KMD нет **Правила Экранирования**, как и у мейннет HUSH, что означает, что новые добытые coinbase могут быть отправлена на taddr сразу же, не заставляя пользователей заражать пул защищённых адресов с утечкой метаданных по причине неверного использования. Изначальная версия мейнет HUSH была основана на исходном коде ZEC и использовало их **Shielding Rule**, но когда Hush запустил второй мейннет в апреле 2019, он был основан на исходном коде KMD и, следовательно, **Shielding Rule** было удалено.

Это означает, что история HUSH и ZEC выглядит иначе с точки зрения блокчейн аналитика. В мейннет сети ZEC, все средства которые на данный момент находятся на прозрачных адресах прошли через защищённый-экранированный пул адресов как минимум один раз, и обычно задействуя все возможные виды утечки метаданных.

Hush отключил переводы на прозрачные адреса на Блоке 340000, и поэтому сейчас единственный вариант отправить средства с прозрачного адреса - это отправить их на zaddrs адреса и никогда не покидать пул защищённых адресов. Zcash продолжает игнорировать принятие zaddr и будет позволять пользователям иметь необязательную (то есть почти нулевую) приватность в дальнейшем неопределенном будущем.

3 Анализ Метаданных Блокчейнов Протокола Zcash: Расширенная версия

3.1 Активные и Пассивные Атаки/Аналитика

Помимо простого анализа общедоступной информации, доступной для каждой полной ноды, существует **активный режим** возможный в любом анализе. То есть, чтобы заразить данные (средства) и посмотреть, как реагирует блокчейн, чтобы "следить за деньгами", если можно так выразиться. Некоторые организации должны предоставить ***zaddrs*** своим клиентам или знать ***zaddrs*** своих клиентов, такие как биржи, майнинговые пулы и провайдеры кошельков. Кроме того, многие люди предпочитают публично публиковать zaddrs и txid, которые связывают их личность в социальных сетях и реальную жизнь с уникальными идентификаторами блокчейна. Многие пользователи случайно деляться этой информацией, не понимая, что Github вопросы (issues) и посты на форумах являются ресурсами для добычи этих данных OSINT, но другие демонстративно публикуют эту информацию, например zecpages.com. Наше мнение таково, что они хотят как лучше, и каким-то образом помогают принятию zaddrs в массы, но так же они слишком упрощают работу по деанонимизации. Многие из этих пользователей публикуют скриншоты, включая их zaddr и id транзакции или ссылки на эксплорер. Это позволяет связать адрес zaddr и ShieldedInput или ShieldedOutput, что не должно быть возможным, и значительно упрощает работу блокчейн аналитика. Это позволяет программному обеспечению потенциально сказать "У этого twitter пользователя такой-то zaddr адрес, средства которого были отправлены txid еще одному Twitter пользователю zaddr адреса" и другие аналогичные примеры.

Как пример активного режима против биржи, которая поддерживает **zaddr**, злоумышленник может создать аккаунт и получить депозит ***zaddr*** на бирже. На данный момент все формы пыльных атак ("dust attacks") доступны для атакующего лица.

Точно так же для майнинговых пулов, которые поддерживают выплаты в **zaddr**, атакующее лицо присоединяется к пулу и майнит достаточное количество для одной выплаты. Теперь они будут знать один из zaddrs и точное количество средств, которое было выплачено в этой транзакции. Майнинговые пулы это огромное количество информации, которая может быть использована для деанонимизации ***zaddrs*** и пулы должны быть аккуратны и не позволять утечки важной информации.

Стоит упомянуть [LuckPool] как пример майнингового пула использующий Лучшие Практики, который поддерживает **zaddrs**, они не публикуют ***zaddrs*** публично, не позволяя искать ***zaddr*** и не показывают какие ***zaddrs*** получили выплату. Еще очень давно сообщество Hush так же связалось со всем майнинговыми пулами Pirate и они удалили общедоступные метаданные о ***zaddr*** майнерах, когда им сообщили о риске для их кон-

фиденциальности. Все майнинговые пулы, которые могут делать выплаты в ***zaddrs*** должны следовать этим примерам. Вся публичная информация о ***zaddrs*** может быть подана для ITM и **Metaverse Metadata** атак.

3.2 Временной Анализ

Данный анализ использует эвристический метод тех транзакций, которые ближе к друг другу и скорее всего связаны, или транзакции образующие подобный временной паттерн. Например, при транзакции в одно и то же время каждый день, или две транзакции, с временным интервалом в 1 час раз в неделю. В публичных (прозрачных) блокчейнах, количество всегда доступно и временной/ценовой анализ имеет большую роль для дальнейшей де-анонимизации. В Протоколе Zcash, у нас есть только тайминг, и только иногда количество. Полностью защищённые $z \rightarrow z$ не имеют какой-либо информации о количестве, в то время как $z \rightarrow t$ и $t \rightarrow z$ имеют только частичную количественную информацию.

Существуют так же недавние атаки временного анализа такие как [PING-REJECT] которая может использовать сетевой временной анализ для связывания IP адреса пользователя и их ***zaddr*** адреса.

3.3 Количественный Анализ

Количественный Анализ и Временной Анализ по сути одинаковые понятия в Биткойн Протоколе, но разделяется на дополнительные методы, когда мы добавляем ***zaddrs*** в анализ. В $t \rightarrow z$ транзакции, у нас есть "идеальная утечка метаданных" в том смысле, что мы знаем точное количество средств отправляемых на защищенный output. Довольно редко, но все же случается, в случае траты output, который точно равен отправленной сумме и плюс комиссия. Так же есть вариант $t, t, ..., t \rightarrow z$ транзакции, создаваемая z_shieldcoinbase RPC, что превращает прозрачные coinbase outputs в один защищённый output и дает утечку всей суммы, переданной на этот единственный защищённый output. Более встречаемая транзакция $t \rightarrow z, z$ вносит неопределенность, но все же дает полезные метаданные. Если прозрачный input был 10 HUSH, то мы знаем, что сумма значений на всех экранированных outputs должна быть 10 HUSH и что любой отдельный output не может быть больше 10 HUSH. Это дает нам максимальное значение (верхнюю границу) для значения в защищённом output, что является полезным для аналитика блокчейна.

Сейчас рассмотрим процесс потери защищённости (de-shielding) $z \rightarrow t$, что так же считается как "идеальная утечка метаданных" в том смысле, что мы точно знаем точное количество в **zaddr**, который владеет этим защищённым output и сейчас находится в прозрачном адресе. Часто встречается $z \rightarrow t, z$ адрес со сдачей добавляет неопределенность и мы не знаем ни точной суммы, поступающей на защищённый адрес для сдачи, ни общей потраченной суммы тем же **zaddr**.

Существуют улучшенные формы Количественного Анализа, такие как **Danaan-Gift Атаки**, так же известные как *дактилоскопия вредоносной ценности* ("malicious value fingerprinting") [BiryukovFeher]. Предположим, идет отправка определенного количества средств на **zaddr**, например 0.72345618 и далее наблюдаем, происходит ли транзакция $z \rightarrow t$, в которой находятся все или большинство этой суммы, возможно слегка изменена из-за стандартной транзакционной комиссии. Данная атака не имеет высокую вероятность работы в любых обстоятельствах, но может быть эффективной "при частом повторении", т.к ничего не останавливает злоумышленника атаковать вновь и вновь.

Hush обойдёт почти все Количественные Анализы, отключив прозрачные outputs в конце Ноября 2020 и станет "конфиденциальным по умолчанию" блокчейном на блоке 340,000 [HushHalving].

3.4 Анализ Комиссий

Этот анализ не самый лучший и не очень эффективный, но зато очень простой, чтобы анализировать комиссию каждой транзакции, не важно защищённой или нет, поиск закономерностей, такие как использование нестандартных комиссий, использование более низких комиссий чем обычно и так же тех, которые платят большие комиссии. Иногда это автоматизированное программное обеспечение, которое создает метаданные о комиссиях, выделяясь среди большинства других способов. В других случаях это другие пользователи выбирающие

индивидуальную комиссию в своем кошельке, пытаясь снизить затраты. Анализ не требует никаких расходов и никак не вовлекает **zaddrs**. Программное обеспечение для анализа комиссий в Биткойн можно напрямую использовать в цепочках Протокола Zcash практически без изменений.

3.5 Атаки Пыли

Пыль (Dust) - это термин, используемый в разговорной речи, а также очень специфический термин, который происходит от внутренностей (internals) исходного кода Биткойна. Мы не нуждаемся в строгом определение и мы использует Пыль подразумевая очень малую (потенциально нулевое) количество, которое стоит дёшево для злоумышленника. Атаки Пыли могут быть в форме **Отказа-в-Обслуживании** или **Утечки Метаданных** и мы ориентируемся на последнюю форму. "Активный режим" в ITM атаке это форма Атаки Пыли, где мы переводим средства известному **zaddr**, узнавая, что происходит дальше.

Подобные атаки могут быть использованы вместе с **Danaan-Gift Attacks** и так же с [BiryukovFeherVitto].

3.6 Input/Output Arity Анализ

Как бы там ни было, Sapling ***zaddr*** транзакции имеют общедоступное количество inputs и outputs. Это пожалуй единственная потерянная функция по сравнению с предыдущей Sprout имплементацией, которая использовала JoinSplits, что скрывало точное количество inputs и outputs. Количество используемых inputs в защищённой транзакции и количество защищённых выходов говорят о многом.

Один упрощенный пример активной "Input Arity Атаки" происходит примерно следующим образом: Атакующее лицо Элис находит или узнаёт о zaddr Боба и она так же знает, что на этом адресе нет средств т.к это только созданный адрес. Затем она переводит 69 (или другое уникальное количество) пыльных outputs в одной транзакции, платя транзакционный сбор. Когда Боб тратит эти средства, Элис может посмотреть на транзакцию, содержащую 69 inputs и далее идентифицировать txid содержащий ***zaddr*** на который она отправляла и затем соединить вместе ее изначальные inputs и outputs этой транзакции.

Что касается output arity анализа, если у вас очень уникальное количество outputs в вашей транзакции в сети - это плохо для вашей анонимности. Если никто, кроме вас не делает переводы с 42 защищёнными outputs каждый вторник в 13:00, все ваши транзакции могут быть анализированы с точки зрения одного единственного владельца, а не потенциально разных владельцев.

Sietch сильно препятствует и input, и output анализу потому что большинство транзакций в сети будут иметь 8 outputs, что означает для всех транзакций с отправкой между 1 и 7 получателями, все выглядят одинаково. В мейннет Zcash, все они могут быть тривиально изолированы и изучены от их output arity. **Hush** смешивает воедино все эти очень распространенные output arity транзакции в "одну корзину". Те, кто отправляют 9 и более ***zaddr*** outputs не могут быть защищены и стандартные output arity гистограммы могут быть использованы для изучения транзакций с большим количеством outputs.

3.7 Биржи и Майнинговые Пулы

Утечка огромного объёма метаданных проходит через биржи и майнинговые пулы в их повседневной работе, поэтому они должны прикладывать большие усилия для уменьшения утечки, что в их интересах, а так же для блокчейнов, на которых они полагаются.

3.8 Что не показывает эксплорер/обозреватель блоков?

Удивительно многое! Около дюжины или более уникальных id (айди) могут быть обнаружены практически в каждой защищённой транзакции и все эти идентификаторы имеют потенциальный риск выдать часть метаданных и коррелироваться друг с другом.

Новая RPC (рандомизированная частичная проверка) z_viewtransaction может быть использована для просмотра всех необработанных данных, которые обеспечивают доказательства с нулевым разглашением (zero knowledge proofs) zaddrs адресов.

4 ITM Атака: Возможность Связывания z2z Транзакций

ITM атакует конкретно транзакцию $T : z \rightarrow z, z$, то есть полностью защищённую транзакцию Протокола Zcash, которая имеет самый высокий уровень конфиденциальности. Сначала мы дадим определение успешной атаки, где любое из следующих данные могут быть установлены:

- Количество отправляемых средств в **zaddr**.
- Количество получаемых средств в любом из **zaddrs**.
- Любое количество потраченных ShieldedInputs в транзакции.
- Ряд возможных значений отправленных на любой **zaddr**, например 0.42 и 1.7 (с учетом погрешности)
- Ряд возможных значений хранящихся в отправителе **zaddr**.

Если утечка каких-либо из вышеперечисленных метаданных может "просочиться" - атака будет считаться успешной. Мы заметили что эта атака полностью пассивна по своей сути, но может быть значительно улучшена путем добавления активных компонентов "по вкусу". Вот почему атаки для утечки метаданных такие поскольку это можно рассматривать как метод анализа или прямую атаку.

ITM Атака берет айди (id's) транзакций и ***zaddrs*** как input, или другие OSINT, которые легко найти в Github, Twitter, Discord, Slack, публичных формах, рассылочном списке, IRC и многих других местах. С этой публичной информацией, **ITM Атака** может перейти с теоретически интересной атаки к действительной деанонимизации ***zaddr*** адреса, который соответствует аккаунтам в социальных сетям, емейл адресам, IP адресам, информации о местоположении и не только.

Данная атака не для "воскресных войнов" или людей с небольшим бюджетом, и не предназначена быть рентабельной для атак на один **zaddr**. Это лучше всего подходит для серьёзных игроков и больших целей, такие как NSA, GCHQ и им подобным. Вероятно, они уже используют анализ атак описанных в этом документе.

Только наиболее финансируемые частные компании, занимающиеся анализом блокчейнов, смогут позволить себе инфраструктуру для такого типа атак, но как только данные были "добыты" - это становится продаваемым товаром для тех, у кого меньше ресурсов.

ITM Атака является дополнительным "слоем" анализа, который может быть наложен поверх всех остальных типов анализа, и таким образом имея потенциальную возможность "закончить" большинство "частичных де-анонимизаций", например места где блокчейн аналитика предоставляет некоторую информацию, но не достаточно для полного де-анона. При добавлении временной, количественной и комиссионной аналитики, можно идентифицировать, что определенные ***zaddrs*** были вовлечены во многих транзакциях и их примерные input и outputs значения. Эта информация не доступна любым иным способом и точные значения не так важны.

Если блокчейн аналитик сможет найти транзакцию, которая вовлекает не менее 1M USD против нескольких пенни - это наложит курс для дальнейшего расследования и анализа. Идеальная де-анонимизация не требуется и на практике не так важна. Программное обеспечение с использованием информации из ITM анализа даст возможность идентифицировать как outputs транзакции, так и определённые промежутки значений и потенциальные вовлечённые zaddrs из OSINT данных.

4.1 ITM Атака: Предположения

В духе радикальной прозрачности и открытости, полностью рабочий пример кода был сделал в качестве упражнения для заинтересованных компаний блокчейн анализа. Мы опишем атаку достаточно детально для

экспертов, чтобы доказать наши доводы, так же для разработчиков, которые смогут использовать данный пример для атаки или защиты.

Мы предполагаем, что у атакующего лица есть хотя бы 100,000 USD средств вовлеченные в изучение операций одного определенного Zcash блокчейна. Основные затраты пойдут на покупку GPU/FPGA ферм для обработки данных. Блокчейны с бóльшей историей и защищённым пулом адресов повлияют на цену соответственно.

Стоит сказать, что данная атака не является разовой с точки зрения финансовых затрат, это методология для изучения всего блокчейна и дальнейшего возможного индексирования и поиска потенциально важной информации. Компании занимающиеся блокчейн анализом и IC готовы стратегически использовать эту информацию за наименьшие расходы, поскольку у них уже есть налаженная инфраструктура для поддержки нового набора данных.

4.2 ITM Атака: Победа над *zk-SNARKs*

Мы можем думать об этой атаке как о "победе" над математикой с нулевым разглашением только на практике, но не в теории. Требуется серьёзная квалификация. Мы ни в какой степени не "сломали" математику ***zk-SNARKs***, мы используем преимущество того как **zk-SNARKs** применен на более высоких уровнях протокола, то есть Zcash Transaction Format Protocol и ему подобные правила консенсуса.

Поэтому **zk-SNARKs** логичны и мы не утеряли **"знания"** напрямую из **zero-knowledge proof** - это математически невозможно. Мы теряем "знания" от того как эти доказательства использованы в крупных системах как Zcash Протокол, который сам по себе является добавлением к Биткойн Протоколу с печально известной утечкой метаданных.

4.3 ITM Атака: Инфраструктура

Данная атака предполагает хранение значительного количества промежуточных данных в дополнение к необработанным блокчейн данным на диске. После компьютерных мощностей, хранение данных занимает вторую позицию по расходам. Возможно, аренда компьютерной мощности поможет снизить расходы, но это не повлияет на расходы связанные с хранением данных. При анализировании блокчейна в B байт, примерная оценка будет $100 * B$ байт на промежуточные данные для анализа информации и далее очень сжатая версия финальной полезной информации может храниться в $B \div 100$ байт или меньше. Поэтому, финальный размер данных будет меньше, чем input данные, но промежуточные данные скорей всего будут на два порядка больше.

Предположим, у нас есть смоделированный искусственный блокчейн на блоке N, находящийся в состоянии покоя, аналитик располагает собственным майнинговым хешрейтом для "рывка" цепи вперед с помощью собственно-определённых правил консенсуса. Этого можно добиться блокируя все посторонние ноды и подключаясь только к локальному хешрейту.

Мы так же допускаем, что аналитик сможет без проблем "запустить" блокчейн на определенной высоте блока и попытаться внести изменения для получения новых данных. Это тривиально возможно с образами виртуальных машин, docker контейнерами и/или Git, и остается в качестве упражнения для мотивированных блокчейн аналитиков.

Возможно существует намного более производительные способы запустить **"ITM Атаку"**, но на данный момент этот известный метод очень дорогой. Это только доступно для компании или организации, которая желает де-анонимизировать целый блокчейн, но это и есть как раз то, от кого/чего мы хотим защититься.

4.4 ITM Атака: Оракул Консенсус

Сейчас рассмотрим определенную $T : z \rightarrow z, z$ на заданной высоте блока H, которая определяет специальный **защищённый пул** включающий непотраченные защищённые outputs и имеющие к ним отношение метаданные, такие как **Merkle Tree** данные.

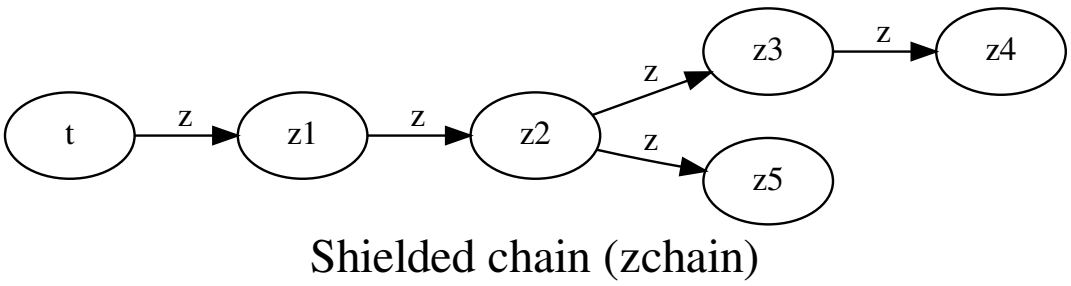
Если более подробно, то симуляция будет использовать **SaplingMerkleTree** внутреннюю структуру данных Протокола Zcash, определенная в src/zcash/IncrementalMerkleTree.hpp. ITM Атака фокусируется на этой структуре данных, но остальные могут и следует быть обнаружены как metadata oracles, например **SaplingWitness** данные.

На полученной высоте блока H защищённая "заметка" или **zUTXO** является либо потраченной, либо нет.

Как и прозрачная **UTXOs**, **zUTXO** может быть создана из *mempool* (ряд неподтвержденных транзакций), например output транзакции в этом блоке может быть потрачена другой транзакцией, как $t \rightarrow z$ тратя UTXO из mempool и создавая zUTXO. ITM Атака не зависит то того, был ли zutxo потрачен из mempool или нет.

Известные Sapling обязательства/якоря являются "свапнутыми" в SaplingMerkleTree по одному, в попытке определить, были ли они израсходованы. Если новое "дерево решений" недействительно, то добавленные данные привели к тому, что оно стало недопустимым деревом по определенной причине и эта конкретная причина удобно приводится, когда ошибки консенсусного уровня генерируются в Протоколах Bitcoin и Zcash. У этих ошибок есть свои ошибочные коды, и они могут стать источником утечки информации для начинающего аналитика. Утечка данных происходит путем проверки различных известных битов данных и анализа точных согласованных кодов ошибок.

Ниже есть изображение канонической ситуации, в которой работает **ITM Атака** , то, что мы называем **zchain** :



Сначала отметим, что удаление zutxo/notes из SaplingMerkleTree не делает транзакции недействительными и расходование транзакционного output зависит от того, действителен и присутствует ли zutxos.

Более простая **zchain** только $t \rightarrow z$ или $t \rightarrow z \rightarrow z$ не имеет достаточной структуры для утечки метаданных. Нужна структура, в которой мы можем удалить "внутренний zutxo", от которого зависят другие вещи.

ITM Атака помечает $z3$ как недействительный с помощью HaveShieldedRequirements() или GetSaplingAnchorAt() возвращая ложный аргумент, когда условия на самом деле действительны. Операция не будет завершена при попытке выполнить $z4$ транзакцию, поскольку доказательство zk-snark обнаружит зависимость в $z2$. ITM называет это "обратным доказательством". Существует так же возможность "прямого доказательства", когда z4 позволяет потратить z2, но не z3. В этом случае мы с большой вероятностью можем сказать $t \rightarrow z1 \rightarrow z2 \rightarrow z3$.

Эти **zchains** являются основными объектами для атак и изучения в ITM Атаке, в итеративном процессе, где изучаются цепочки размера N и иногда может быть определена связь, но часто это невозможно. Когда **ITM Атака** находит действительное обратное доказательство, ITM может попытаться расширить свои знания попробовав подцепи (subchains), чтобы получить больше метаданных. Это форма добычи метаданных. Каждый раз, когда добывается новый блок, могут быть потрачены новые средства, и процесс можно повторить.

Подводя итог, **ITM Атака** требует, чтобы zutxo был потрачен для попытки отследить его связность с другими предыдущими ztuxos. Неизрасходованный zutxo не может быть проанализирован. Кроме того, $t \rightarrow z$ и $z \rightarrow t$ в настоящее время не кажутся уязвимыми. Только $z \rightarrow z$ транзакции могут быть проанализированы, и только

"внутри" zchain может происходить утечка метаданных, самые новые неизрасходованные zutxos "не выдадут" метаданные.

5 Атаки Метавселенная Метаданных/Metaverse Metadata Attacks

Атака ITM является частным случаем того, что мы называем **Атаки Метавселенная Метаданных**, применяемые на защищенные графики транзакций в Протоколе Zcash.

Термин **Метавселенная** уместен, потому что можно смоделировать альтернативные возможные истории блокчейна, чтобы увидеть, какие правила консенсуса были бы произведены. Тщательно изменяя по одному фрагменту данных за раз, аналитик может использовать правила консенсуса в такой момент истории блокчейна как **oracle**. В этом смысле, **Metaverse** атаки можно классифицировать как **консенсусные oracle атаки**, и аналогичные атаки: **oracle сжатие** и **oracle дополнение**, такие как [BREACH], [CRIME] и [HEIST] против SSL/TLS.

В то время как вышеупомянутые атаки являются **атаками по сторонним каналам** с использованием времени ответа на запросы, Атаки Метавселенная Метаданных - это сторонние каналы изучающие данные публичной цепи и ошибки консенсус-уровня в симуляциях.

Это новый метод, насколько известно авторам, который публично не описывался. Консенсусные правила блокчейна можно моделировать в вакууме, а научный метод изменения одной переменной за раз можно использовать для извлечения метаданных из общедоступных данных конфиденциальной криптомонеты. Существует неисчислимое количество метаданных, которые можно "добыть" из общедоступных данных блокчейна, соединенных с источниками данных OSINT.

6 Sietch: Теория

6.1 Sietch: Основы

ITM Атака основана на том факте, что наиболее распространенная защищённая транзакция в большинстве существующих в настоящее время блокчейнов Zcash Протокола имеет только 2 outputs $T : z \to z, z$ и основной факт, что если какое-то количество "утраченных" метаданных об одном output, если это **израсходованное** или **неизрасходованное**, или диапазон данных возможных значений - в этих случаях много метаданных будет получено и для другого output.

Если бы было 3 outputs, тогда возникла бы неопределенность, а не более прямое алгебраическое соотношение, такое как "если один output имел сумму=5, то другой output имел сумму $total - 5$". Когда задействованы 3 ***zaddr*** outputs, зная значения одного ***zaddr*** output не дает столько информации о значении любого другого конкретного **zaddr**.

Очевидно, что этот принцип усиливается, поскольку увеличивается количество outputs, утечка любого количества ***zaddr*** input делают метаданные значительно менее ценными и дорогими для использования.

Хорошим решением было сделать Sietch обязательным для использования по своему замыслу, и все пользователи используют Sietch по умолчанию не зная об этом. Sietch делает каждую отдельную защищённую транзакцию более сложной, что затрудняет анализ граф транзакций, помогающий даже пользователям со специальным программным обеспечением, не использующее Sietch.

"Коллективный иммунитет" против деанонимизации - это совместное воздействие того, что почти все пользователи Hush, не подозревая об этом, постоянно используют Sietch. Каждая транзакция нуждается в нескольких дополнительных секундах, и сообщество Hush считает, что оно того стоит.

Даже если некоторые outputs транзакции полностью деанонимизированы, существует множество других outputs, где точные отправляемые значения не могут быть установлены, что имитирует случай когда инфицированный человек не может легко заразить другого человека вирусом, так как люди рядом с ним уже выздоравливают или имеют иммунитет.

6.2 Sietch: Индетерминизм

В дополнение к минимальному количеству ***zaddr*** outputs, Sietch вводит **индетерминизм** в Zcash Протокол. Хорошая идея была Zcash'у унаследовать детерминизм от Биткойна. Но в приватных монетах оказывается, что детерминизм может снизить конфиденциальность в некоторых ситуациях, и это на самом деле не является требованием для работы криптовалюты.

Sietch использует 3 вида индетерминизма:

- 1 Порядок автоматически добавляемых ***zaddr*** outputs случайны
- 2 Точное количество автоматически добавленных outputs случайны
- 3 Отправленные ***zaddrs*** случайны

Разработчики Hush считают, что недетерминизм является мощным средством против **Metaverse Атак**, потому что при попытке смоделировать блокчейн и искать oracles или "добыть" полезные биты метаданных, результат "теста" больше не детерминирован, и поэтому некоторые атаки станут непрактичными или невозможно.

7 Sietch: Код в Производстве

Sietch использует по умолчанию правило минимального количества outputs, в виде 7 ***zaddr*** outputs в транзакции, потому что средняя защищённая транзакция не расходует точные input значения и есть output сдача, на практике средняя транзакция Hush имеет 8 ***zaddr*** outputs.

На данный момент это не правило консенсуса и применяется только на уровне RPC. В настоящее время есть различные реализации Sietch в нашем full node (SD, SilentDragon) и lite (более простая версия, SilentDragonLite) кошельке.

Каждый раз, когда транзакция выполняется с количеством менее 7 ***zaddr*** outputs, уровень RPC автоматически добавляет их, что означает, что все программные обеспечения, использующие уровень RPC защищены без какого-либо изменения кода. Программное обеспечение, использующее необработанные транзакции должны позаботиться об этом сами.

На практике это позволяет скрыть количество получателей до средних транзакции в сети Hush. Когда вы видите $z \rightarrow z, z, z$ транзакцию в основной сети ZEC вы можете быть почти уверены, что это один ***zaddr*** отправляет 2 другим ***zaddrs*** и output сдачи. Он так же может быть отправлен на три outputs без сдачи, с резко меньшей вероятностью. Этот тип транзакции "обновлен" как минимум до $z \rightarrow z_7$ и поэтому вы не знаете, скольким получателям были отправлены средства, за исключением случаев, когда это большое количество. На практике это скрывает большинство транзакций в сети, и в основном это выплаты из майнинг пула, которые обычно используют множество ***zaddr*** outputs или другое автоматизированное программное обеспечение.

Некоторые транзакции выглядят как $t \rightarrow t, t, z, t$, что является прозрачным адресом, отправляемый на два других прозрачных адреса, один защищённый адрес и output сдачи. Когда Sietch включен, эта транзакция "улучшается" до $t \rightarrow t, t, z, t, z_6$, чтобы удовлетворить минимум 7 ***zaddr*** output правило. Первоначально была известна точная сумма, передаваемая в **zaddr**, потому что все другие значения в транзакции прозрачны и отображаются в общедоступной цепочке блоков. Но в "улучшенной" транзакции мы можем только убедиться, что некоторая сумма A была отправлена и распространена через 7 outputs, некоторые из которых могут иметь нулевое значение.

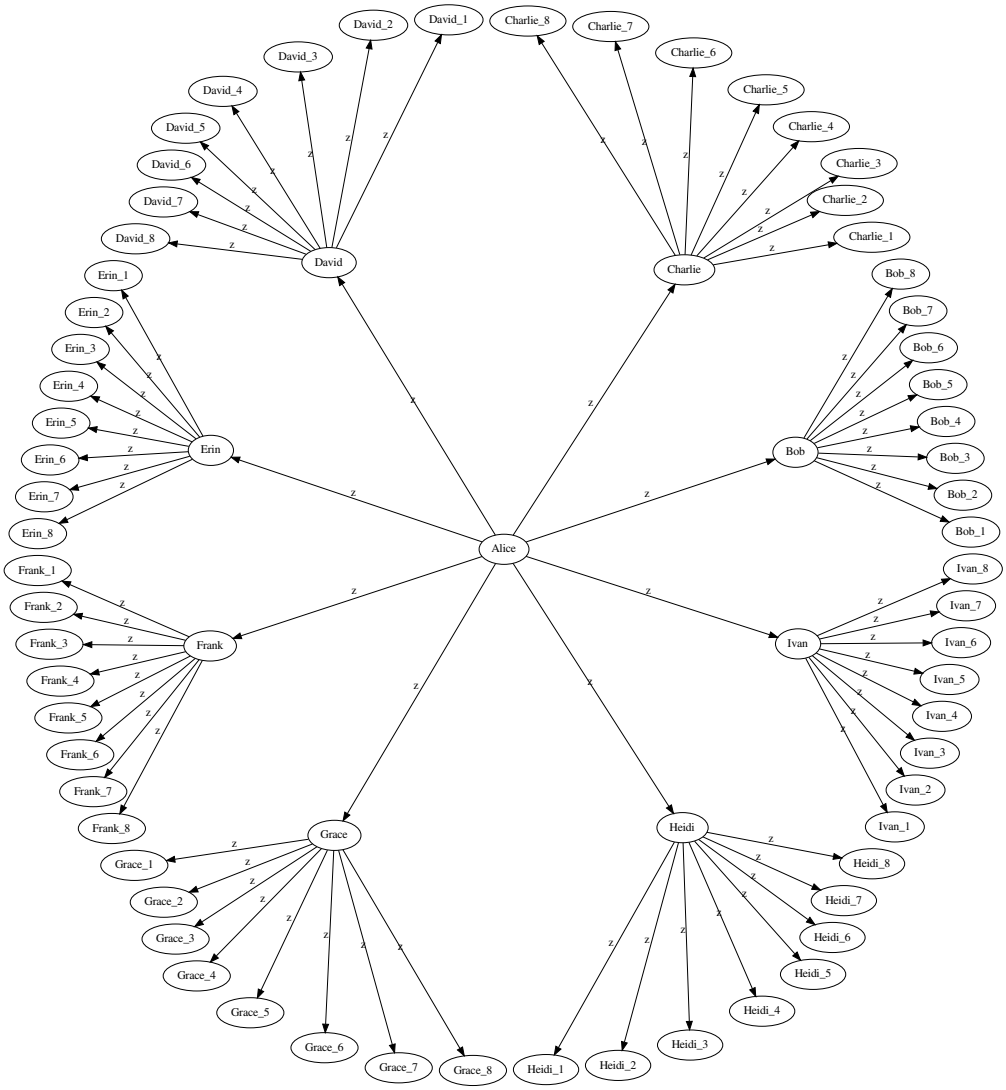
Как правило, транзакции Sietch значительно усложняют деанонимизацию цепочки на индивидуально транзакционном уровне, которые затем выстраиваются в очень надежный и сложный защищённый граф транзакций. Средняя защищённая ZEC транзакция в основной сети имеет два outputs, поэтому этот защищённый график транзакций выглядит как двоичное дерево, а блокчейн Hush с Sietch выглядит как дерево, которое разбивается на 8 частей на каждом узле. Попытка проследить за потоком средств становится комбинаторно непрактичной и дорогой даже для самых крупных игроков.

Здесь мы сравниваем, как выглядит граф защищённых транзакций в основной сети Zcash (ZEC) по сравнению с графом защищённых транзакций, который мы увидим с **Sietch** в основной сети HUSH. Эти два графика

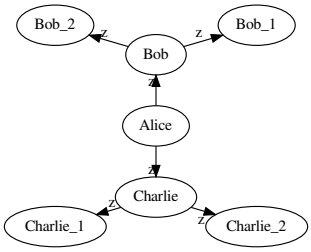
показывают два **hops, прыжка-перехода**, где мы определяем один переход (hop) как $z \rightarrow z$ и два перехода (hops) как $z \rightarrow z \rightarrow z$ и так далее. После нескольких переходов (hops) легко увидеть, что защищённый граф транзакций блокчейна с поддержкой **Sietch** блокчейна превращается в "звезду" потенциальных направлений поступления средств. Традиционная цепочка Zcash Протокола представляет собой двоичное дерево, и это означает, что если в любой момент вы можете взять под контроль этот ***zaddr*** output, вы знаете метаданные о большом подграфе графа транзакций, например, захват незащищенного wallet.dat файла с мобильного телефона, ноутбука или персонального компьютера. Вместе с **Sietch** , если у кого-то из друзей Алисы конфискуют телефон, там по-прежнему остаются 7 из 8 мест, куда могли бы уйти средства, что могло быть 1-7 фактических outputs или некоторое количество outputs автоматически добавленные **Sietch**'ем. Невозможно точно узнать **сколько** люди получили средств, за исключением того, что получили не более 8, и мы не знаем, все ли средства перевелись одному ***zaddr*** output и остальное было нулевым, или некоторая комбинация средств с несколькими ***zaddrs*** .

Наша цель, чтобы атакующие лицо вынуждено было изучать гораздо более сложный набор данных из-за **Sietch** - это превращает каждую транзакцию Hush в отдельную крепость, а затем, когда мы соединяем много таких транзакций, весь защищённый граф транзакций становится очень устойчивым к деанонимизации в любых условиях. В среднем **Sietch** силен во всех областях этого большого набора "нод"(узлов) и "ребер".

После 10 переходов Sietch распределит ***zaddr*** средства на потенциально 8^{10} = 1073741824 защищённых outputs в среднем, в то время как "простой" протокол Zcash дает график транзакций размером примерно 2^{10} = 1024.



Sietch-Enabled Shielded Transaction Graph



Zcash Mainnet Shielded Transaction Graph

8 Детали Реализации

В настоящее время у нас есть четыре реализации Sietch, две из которых находятся в производстве, одна устаревшая и еще одна в стадии тестирования. Первоначальные отзывы разработчиков приватных монет указали

на некоторые проблемы в изначальных реализациях, поднимая модели угроз, о которых мы вначале не думали.

Ранее все реализации Sietch имели фиксированный список zaddrs адресов, встроенных в исходный код, и эти zaddrs были добавлены случайным образом в качестве outputs для ***zaddr*** транзакций. Это не лучший вариант, потому что если приватные ключи из этих Sietch адресов скомпрометированы, становится возможным включить эти данные в цепочку программного обеспечения для анализа и потенциально убрать преимущества конфиденциальности Sietch. Отмечаем, что в худший вариант - это вернуться к приватности существовавшая до Sietch.

В ответ на это, разработчик Hush реализовал рандомизированные Sietch ***zaddrs*** во время исполнения, которые никогда не сохраняются в исходном коде или на диске. Случайная seed фраза генерируется, а затем из этой seed фразы создается случайный **zaddr**, а затем приватный ключ и seed фраза немедленно удаляются из памяти. Практически невозможно деанонимизировать людей массово, поскольку каждый пользователь теперь генерирует Sietch ***zaddrs*** в памяти. Для восстановления этих приватных ключей или seed фраз требуется чтение памяти с отдельных узлов (нод). В настоящее время SilentDragonLite (SDL) использует этот метод. Полный узел **hushd** изначально использовал фиксированный набор из 200 случайно выбранных ***zaddrs*** [SietchRPC] [SietchHeader], но теперь имитирует поведение SDL. В то время как SDL генерирует случайную seed фразу, а затем использует полученные из нее защищённые адреса, **hushd** просто генерирует случайные публичные ключи, для которых он не владеет приватным ключом, а затем извлекает защищённые адреса из этого открытого ключа.

Мы также отмечаем, что все Sietch outputs являются действительными и расходуемыми, они не являются "фальшивыми" и не являются недействительными outputs, которые нельзя израсходовать, поскольку мы полагаем, что они могут быть обнаружены и повлечь за собой утечку метаданных. Во всех смыслах, от используемых криптографических примитивов до энтропии данных, Sietch zdust - это легитимные и достоверные защищенные данные транзакций, что делает его таким мощным инструментом.

9 Мысли об Изъятии Оборудования

Допустим, Алиса отправила Бобу и Чарли криптовалюту в полностью защищённой транзакции с защищённой сдачей: $z_A \rightarrow z_B, z_C, z_A$.

Теперь предположим, что устройства Алисы и Чарли были конфискованы, файл wallet.dat "освобожден" и загружен в программное обеспечение для анализа цепочек, которое понимает Zcash Протокол и Атаки в стиле ITM. Теперь Боб находится в положении, где его ***zaddr*** известен аналитику/злоумышленнику, точная сумма, отправленная ему в определенных транзакциях, и, возможно, другие метаданные в мемо-поле. Все эти данные являются ценными input, которые улучшают работу ITM-атаки и часто могут помочь "завершить" частичную деанонимизацию, которая не смогла полностью "разрешить" данные.

Даже без каких-либо новых атак, захват устройства и загрузка содержимого wallet.dat в программное обеспечение для анализа блокчейна представляет огромную угрозу для приватных/анонимных монет, поэтому им следует разрабатывать системы, которые предполагают, что это произойдет, и изолировать, и сопоставить возможный ущерб. Sietch предоставляет один из таких способов обеспечить защиту и конфиденциальность от реальных случаев.

10 Рекомендации для монет на Протоколе Zcash

Малое количество ***zaddr*** outputs плохо для конфиденциальности, особенно 1 или 2. Применение хотя бы **4**, вероятно, сделает атаку ITM непрактичной, поскольку существует очень много потенциальных способов поменять местами оставшиеся inputs. Hush выбрал **7** в качестве буфера безопасности, так как замедление, связанное с 7 outputs, составляет около 5 секунд или меньше на современном оборудовании при использовании небольшого количества inputs. Это казалось разумным промежутком времени для совершения транзакции, учитывая, что исходные Sprout ***zaddrs*** занимали более минуты, чтобы осуществить простейшие транзакции.

Транзакции будут выделяться, если позволить пользователям тратить огромное количество inputs одновременно. GUI Кошельки (с графическим интерфейсом) и необходимость обучения пользователей очень важна для уменьшения потери конфиденциальности.

Не призывайте пользователей размещать **zaddrs**, ссылки на txid и explorer, в которых они участвуют! Обучайте их хранить эти метаданные в личных сообщениях, прямых переписках и других закрытых местах. Чем меньше людей знают ваш **zaddr**, тем больше вы сохраняете свою приватность!

10.1 Sapling Консолидация

Sapling Консолидация рекомендуется для среднего пользователя и обеспечивает защиту от атак на метаданные, а также от **Denial-of-Service** атак в дополнение к своей основной функции - уменьшения размера файлов wallet.dat и, следовательно, ускорения их использования. **Hush** добавил **Sietch** в нашу реализацию Sapling Консолидации, а также снизил утечку метаданных за счет уменьшения количества inputs, которые пользователь когда-либо будет тратить за один раз, что составляет 8, дабы соответствовать среднему количеству outputs в **Sietch**.

Это означает, что когда эта функция включена, и узел "нода" получает атаку пыли из множества мелких inputs, узел волшебным образом очистится после атаки в фоновом режиме с лучшими Практиками для каждой транзакции. Эти транзакции гарантированно оставят размер нашего **набора анонимности** прежним или увеличат его на 1 (если нет output сдачи).

Первоначальная реализация Sapling Консолидации, написанная для монеты ZER, потребляла бы до 45 inputs одновременно и всегда отправляла бы на 1 output с комиссией=0, что тривиально выделялось в сети. В сети **Hush** эти транзакции консолидации выглядят в точности как очень распространенные $z \rightarrow z, z, .., z$ между 1-8 inputs и 7 или 8 outputs, смешиваясь с большим количеством транзакций, которые имеют те же свойства.

11 Дальнейшие Обсуждения

В этом разделе рассматриваются различные новые технологии, появляющиеся на рынке, и то, как они взаимодействуют с существующими и новыми методами анализа метаданных.

11.1 Защищённая coinbase ZIP-213

Защищённая coinbase интересна, но в ней происходит утечка большого количества метаданных, привязанных к zaddress майнера, которые могут использоваться в этом анализе. Мы рекомендуем Pirate, Arrow и другие монеты, реализующие принудительное использование **zaddr**, избегать внедрения новой [ZIP-213] "Защищённой coinbase". Сообщество Hush не соглашается с окончательным выводом ZIP-213 о том, что можно сделать ***zaddr*** output майнера общедоступным и что только пользователи, озабоченные "постквантовой" конфиденциальностью, должны беспокоиться об утечке метаданных. Sietch можно рассматривать как надежную защиту от квантовых компьютеров, но требуются дальнейшие исследования, чтобы увидеть, какие показатели квантовые компьютеры могут иметь для алгоритмов теории графов, составляющих основную часть атаки.

Защищённая Coinbase резко снизит конфиденциальность ***zaddr*** майнеров, потому что они будут повторно использовать один и тот же ***zaddr*** для каждого блока, и это приведет к утечке ***zaddr*** на котором происходит добыча монет. "Нормальное" поведение майнинга сначала происходит на taddr, а затем переводит на **zaddr**, который изолирует утечку метаданных на taddr, ***zaddr*** майнера никогда не разглашается публично. ZIP-213 говорит о том, что майнеры должны создавать новый адрес для каждого блока, но это просто не произойдет - никто этого делать не будет, так как это необязательное условие пользования, так же это делает файлы wallet.dat очень большими, медленными, более сложными для резервного копирования и, что наиболее важно, простои, необходимые для остановки zcashd и перезапуск с новым zaddr напрямую приводят к потере

денег для майнера. Все исследования анонимных монет указывают на тот факт, что большинство пользователей делают только то, что является обязательным, они не делают дополнительной работы для обеспечения конфиденциальности. Майнеры не исключение.

Используя Временной Анализ и Количественный Анализ с Защищённой Coinbase, аналитик может получить гораздо лучшую оценку минимального значения, которое, вероятно, имеет **zaddr**, и сколько средств проходит через него за интервал времени, а также txid, которые коррелируют с **zaddr**. Все они также могут использоваться в качестве inputs для ITM Атаки. Кроме того, ***zaddr*** майнеров открываются для атак с помощью атак пыли, потому что их ***zaddr*** навсегда останется всем известным в публичном блокчейне.

ZIP-213 - это увлекательное теоретизирование, которое можно реализовать с лучшими свойствами конфиденциальности, но меньшей возможности проведения аудита, то есть с точным знанием того, сколько новых средств добывается в каждом блоке. Принимая во внимание ITM Атаку в частности и атаки Метавселенная Метаданных (Metaverse Metadata) в целом, ZIP-213 не повысит конфиденциальность цепочки блоков, а уменьшит ее, заразив защищённый пул слишком большой утечкой метаданных. По этим многим причинам мир Hush и Komodo игнорирует ZIP-213 и, действительно, игнорирует всё Обновление Сети Heartwood, поскольку оно не имеет функций анонимности.

Таким образом, Защищённая Coinbase была реализована компанией Electric Coin с небольшим практическим учетом повышения конфиденциальности в их блокчейне, хотя это интересная техническая часть работы, поскольку увеличение использования ***zaddr*** не приводит к увеличению прибыли, маловероятно, что у них когда-либо будет значимая конфиденциальность в основной сети Zcash. Только блокчейны протокола Zcash, которые обеспечивают использование **zaddr**, имеют шанс на значительную конфиденциальность.

12 Особая Благодарность

Особая благодарность всем членам сообщества HUSH и людям, которые заботятся о конфиденциальности, которые дали конструктивный отзыв об этой статье, включают Daira Hopwood, jl777, zawy, denioD, Biz и, конечно же, ITM, которые сообщили об атаке сообществу HUSH и убедил нас, что это правда, хотя мы ему долго не верили.

13 Признательность

Это независимо финансируемая исследовательская работа без сторонних источников финансирования. Финансирование от компании Electric Coin, Zcash Foundation или любой другой коммерческой или некоммерческой организации получено не было.

14 Справочные Материалы

[BiryukovFeher] Daniel Feher Alex Biryukov. *Deanonymization of Hidden Transactions in Zcash.* URL: https://cryptolux.org/images/d/d9/Zcash.pdf (дата обр. 2020-05-08) (цит. на с. 6).

[BiryukovFeherVitto] Giuseppe Vitto Alex Biryukov Daniel Feher. *Privacy Aspects and Subliminal Channels in Zcash.* URL: https://orbilu.uni.lu/bitstream/10993/41278/1/Post_sapling_ZC_paper.pdf (дата обр. 2020-05-08) (цит. на с. 7).

[Bitcoin] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System.* 8 мая 2020. URL: https://bitcoin.org/bitcoin.pdf (дата обр. 2020-05-08) (цит. на с. 4).

[BREACH] Wikipedia. URL: https://en.wikipedia.org/wiki/BREACH (дата обр. 2020-05-08) (цит. на с. 11).

[CRIME] Wikipedia. URL: https://en.wikipedia.org/wiki/CRIME (дата обр. 2020-05-08) (цит. на с. 11).

[CryptoNote] N. van Saberhagen. *CryptoNote*. URL: https://cryptonote.org/whitepaper.pdf (дата обр. 2020-05-08) (цит. на с. 3).

[HEIST] Tom Van Goethem M. Vanhoef. URL: https://tom.vg/papers/heist_blackhat2016.pdf (дата обр. 2020-05-08) (цит. на с. 11).

[Hush] The Hush Developers. *Hush*. URL: https://myhush.org (дата обр. 2020-05-08) (цит. на с. 3).

[HushGenesis] Hush. *Old Hush Block Explorer*. URL: https://hushold.explorer.dexstats.info/block/0003a67bc26fe564b75daf11186d360652eb435a35ba3d9d3e7e5d5f8e62dc17 (цит. на с. 4).

[HushHalving] Hush Developers. *Hush Halving Countdown*. URL: https://myhush.org/halving (дата обр. 2020-05-24) (цит. на с. 6).

[KMDGenesis] Komodo. *Komodo Block Explorer*. URL: https://kmd.explorer.dexstats.info/block/0a47c1323f393650f7221c217d19d149d002d35444f47fde61be2dd90fbde8e6 (дата обр. 2020-05-24) (цит. на с. 4).

[LuckPool] hellcatz. *LuckPool*. URL: https://luckpool.net/hush/ (дата обр. 2020-05-08) (цит. на с. 5).

[Monero] Monero Developers. *Monero - Secure, Private, Untraceable*. URL: https://getmonero.org (дата обр. 2020-05-08) (цит. на с. 3).

[OBitcoinWhereArtThou] Erwin Filtz Bernhard Haslhofer Roman Karl. *O Bitcoin Where Art Thou? Insight into Large-ScaleTransaction Graphs*. SEMANTICS 2016: Posters and Demos Track September 13-14, 2016, Leipzig, Germany. URL: http://ceur-ws.org/Vol-1695/paper20.pdf (дата обр. 2020-05-08) (цит. на с. 3).

[PING-REJECT] K. Paterson F. Tramèr D. Boneh. *PING and REJECT:The Impact of Side-Channels on Zcash Privacy*. URL: https://crypto.stanford.edu/timings/pingreject.pdf (дата обр. 2020-05-09) (цит. на с. 6).

[SietchHeader] The Hush Developers. *hushd src/sietch.h*. URL: https://github.com/MyHush/hush3/blob/c271fb8cbde9b7e575a3759598750f1c79e374d7/src/sietch.h (дата обр. 2020-05-08) (цит. на с. 15).

[SietchRPC] The Hush Developers. *hushd src/wallet/rpcwallet.cpp*. URL: https://git.hush.is/hush/hush3/src/branch/master/src/wallet/rpcwallet.cpp#L4674 (дата обр. 2020-05-08) (цит. на с. 15).

[Zcash] Daira Hopwood. *Zcash Protocol Specification*. URL: https://github.com/zcash/zips/blob/master/protocol/protocol.pdf (дата обр. 2020-05-08) (цит. на с. 3).

[ZcashGenesis] Zcash. *Zcash Block Explorer*. URL: https://blockchair.com/zcash/block/0 (дата обр. 2020-05-24) (цит. на с. 4).

[ZIP-213] Jack Grigg. *Shielded Coinbase*. URL: https://zips.z.cash/zip-0213 (дата обр. 2020-05-09) (цит. на с. 16).

Высказывайся И Совершай Транзакции Свободно - hush.is