Атомарные свопы между блокчейнами Bitcoin и Monero

Джоэл Гуггер (Joël Gugger)

h4sh3d@protonmail.com

Аннотация Атомарные свопы уже используются в случае с блокчейнами, где возможно использование контрактов с хешированной временной блокировкой. Но когда один из блокчейнов не предусматривает такой возможности, это становится проблемой. Данный протокол описывает, как обеспечить возможность проведения атомарных свопов между блокчейнами Bitcoin и Monero в рамках двух транзакций на блокчейн без привлечения каких-либо централизованных организаций, серверов или других участников, требующих доверия. Нами предлагается вариант проведения свопа между двумя участниками, один из которых владеет Bitcoin, а другой - Monero. При этом, если оба участника последуют этому протоколу, их средства ни на каком этапе не будут подвергаться какому-либо риску. Протокол не требует ни использования временной блокировки со стороны Monero, ни реализации скриптов, но требует наличия двух доказательств знания равного дискретного логарифма в группах `edward25519` и `secp256k1`, а также одноразовой подписи VES `ECDSA`.

Ключевые слова: блокчейн, атомарный своп, проведение транзакций между блокчейнами, Bitcoin, Monero

1 Введение

Нами описывается протокол проведения атомарных свопов между блокчейнами Monero и Bitcoin, но данный протокол можно обобщить для любой криптовалюты, которая соответствует тем же требованиям, что и Monero, а также любой другой криптовалюты, которая соответствует тем же требованиям, что и Bitcoin. Подробный список предварительных условий приводится в Главе 3.

Участники отправляют средства на определенный адрес, генерируемый во время процесса (блокировка) в каждом блокчейне (обмен между блокчейнами), и каждая сторона может взять под контроль средства в другом блокчейне (своп) автоматически (то есть возможность получения средств из любого блокчейна является взаимоисключающей из-за возможности забирать средства из другого блокчейна).

В ходе реализации данного процесса участники не раскрывают своих средств, если следуют протоколу надлежащим образом, а это означает, что своп не требует доверия и не требует никакого залога, что позволяет двум незнакомым людям обмениваться криптовалютами без каких-либо рисков или посредничества третьей стороны.

2 Сценарий

Опишем участников и их мотивацию. Элис, у которой есть Monero (XMR), и Боб, у которого есть Bitcoin (BTC), хотят обменяться своими средствами. Предполагается, что они договорились о цене заранее (то есть согласовали ту сумму Bitcoin, которая будет обменена на определённую сумму Monero). Такое согласование сумм также может быть интегрировано в протокол, например, с помощью сервисов обмена, которые будут указывать соответствующую цену своим клиентам.

Оба участника желают использовать только один из двух возможных путей выполнения протокола (которые являются взаимоисключающими друг для друга): (1) протокол выполняется успешно, и Элис получает Bitcoin, а Боб получает Monero, или же (2) протокол не выполняется, и оба сохраняют свои изначальные средства за вычетом минимально возможной комиссии за проведение транзакций.

2.1 Успешный своп

Если оба участника будут следовать протоколу, то будет проведено всего четыре транзакции (три, если Monero не будут переведены сразу после завершения, что не является проблемой): две в блокчейне Bitcoin и две в блокчейне Monero. Первые транзакции в обоих блокчейнах блокируют средства и готовят их к обмену в каждом из блокчейнов. Вторые разблокируют средства только для одного участника и

дают об этом знать другому участнику, который принимает на себя контроль над выходом в другом блокчейне.

Это оптимальный вариант выполнения протокола, не требующий временной блокировки, использующий минимальное количество транзакций и только блокировку средств для минимального подтверждения в каждом из блокчейнов в зависимости от уровня безопасности, ожидаемого каждым участником, то есть сколько подтверждений транзакции потребуется каждому участнику, чтобы она считалась окончательной и можно было бы перейти к следующему шагу, предусмотренному протоколом.

2.2 Правильное прерывание свопа

При блокировке Bitcoin (после временной блокировки) Элис или Боб могут запустить процесс возврата заблокированных средств. На этот момент Monero, возможно, еще не будут заблокированы. Если ни одна Monero заблокирована не была, в процессе будут возвращены только Bitcoin. В противном случае у Элис будет достаточно информации, чтобы вернуть и свои Monero.

Когда транслируется транзакция возврата, Боб должен осуществить возврат до истечения срока некоторой временной блокировки, иначе в конечном счёте он может потерять свой Bitcoin, не получив при этом Monero. С точки зрения Боба это можно описать как интерактивный протокол, то есть Боб не может перейти в оффлайн - он должен реагировать на подобную ситуацию во время свопа. Элис, с другой стороны, может оставаться оффлайн.

2.3 Наихудший сценарий

Если своп отменяется запуском процесса возврата и Боб не воспользуется возможностью возврата до истечения срока временной блокировки, Элис сможет потребовать возврат средств, не раскрывая известной информации, необходимой Бобу, чтобы воспользоваться возвратом в другом блокчейне. Таким образом, один участник, Боб, оказывается в невыгодном положении, и уже требуется три транзакции Bitcoin вместо двух.

Обоснование. Мы выбрали этот вариант, чтобы избежать следующего вероятного сценария: если Monero будут заблокированы, Элис сможет вернуть их тогда и только тогда, когда Боб уже вернёт свои Bitcoin. Нам нужен механизм, который бы мотивировал Боба воспользоваться возможностью возврата, чтобы избежать тупиковой ситуации в процессе возврата или чтобы Элис получила компенсацию, если Боб не станет следовать протоколу надлежащим образом.

В противном случае Боб, обладая всей полученной им информацией, может уйти в оффлайн и перевести свои Bitcoin через год после свопа, вынуждая Элис постоянно отслеживать блокчейн, пока она не увидит транзакцию Боба и не узнает последнюю часть информации, которая ей необходима, чтобы разблокировать её Monero.

3 Необходимые условия

Как говорилось ранее, чтобы своп проводился автоматически, необходима возможность условного выполнения. Bitcoin использует простой язык сценариев на базе стеков, который обеспечивает возможность условного выполнения и временных блокировок. С другой стороны, в настоящий момент протокол RingCT, ориентированный на обеспечение анонимности Monero, для разблокировки UTXO предусматривает только использование подписей. Контроль над UTXO связан только с тем, кто контролирует соответствующие приватные ключи. Тогда задача состоит в том, чтобы передать контроль над средствами только при условии знания некоторых приватных ключей.

В этой главе мы рассмотрим все необходимые части информации, необходимые для выполнения атомных свопов в обоих блокчейнах и вне блокчейна в рамках ранее раскрытого сценария.

3.1 Monero

Monero не требует использования каких-либо примитивов в блокчейне (блокировки по хешам, временной блокировки). Все структурные элементы являются примитивами, реализуемыми вне блокчейна. Поэтому нам необходимо предоставить доказательства правильной инициализации протокола, подобные тем, что описаны в подпункте 4.3. Эти доказательства гарантируют атомарность свопа для каждого участника.

Секретные части. В случае с Monero необходимы для обеспечения двустороннего выполнения. Приватный ключ траты Monero разбивается на две секретные части: k_a^s и k_b^s. Участники не используют какого-либо протокола мультиподписи. Вместо этого при инициализации процесса свопа секретные части распределяются между участниками, и один из них узнаёт полный ключ $k^s \equiv k_a^s + k_b^s \pmod{l}$ при окончании выполнения протокола либо при удачном завершении свопа, либо при его прерывании.

3.2 Bitcoin

Транзакции Bitcoin в рамках данного протокола требуют гибкости транзакций, которая была обеспечена обновлением `SegWit`. Это позволяет нам включать транзакции в блокчейн без их общей трансляции. Данный протокол совместим только с теми криптовалютами, которые используют модель `UTXO` в стиле Bitcoin и имеют эквивалентный механизм обеспечения гибкости, такими как Litecoin (то есть с Bitcoin Cash этот протокол будет несовместим).

Временная блокировка. Используется для реализации новых путей выполнения по истечении какого-либо предварительно заданного интервала времени, например, для запуска процесса возврата после блокировки средств в одном из блокчейнов без какой-либо конкуренции. Стоит отметить, что при этом в другом блокчейне не требуется никакой временной блокировки.

Блокировка по хешу. Используется для синхронизации обоих блокчейнов перед запуском свопа.

Схема мультиподписи «2 из 2». Используется для создания общего пути, доступного только для двух участников при наличии их согласия. В рамках данного протокола мы используем опцию, реализуемую в блокчейне, в контексте Bitcoin, но схемы мультиподписи, реализуемые вне блокчейна, более эффективны, и при иных вариантах следует использовать именно их.

Протокол частичного отсутствия скриптов. Используется, чтобы раскрыть секретную часть и продолжить выполнение протокола, не прибегая к сложным скриптам. Мы используем одноразовую подпись VES ECDSA, подобную описанной в подпункте 3.4. Следует отметить, что полный протокол «скриптов без скриптов» [5] должен позволить добиться тех же результатов, но с большей эффективностью.

3.3 Применение равного дискретного логарифма в групповых доказательствах знания с нулевым разглашением

Применение равного дискретного логарифма в групповых доказательствах знания с нулевым разглашением, описанное в технической записке [2], позволяет верифицировать общий дискретный логарифм α при наличии двух групп с заданными генераторами $G \in \mathbb{G}$ и $H \in \mathbb{H}$, где $\alpha \leq min(|G|, |H|)$, что возвращает в результате xG и yH: $x = y = \alpha$.

В нашем контексте мы фокусируем внимание на группах `edward25519` с $|G| = l$ и `secp256k1` с $|H| = n$.

Определение 1 (применение равного дискретного логарифма в групповой схеме). *Применение равного дискретного логарифма в групповой схеме определяется двумя алгоритмами с набором параметров* $(\mathbb{G}, \mathbb{H}, G, G', H, H')$*:*

- $\mathit{DLProve}(\alpha) \to (\phi, A, B)$*: вероятностный алгоритм доказательства, который при наличии вводного дискретного логарифма* α *выдаёт доказательство* ϕ*, точку* $A \in \mathbb{G}$ *и точку* $B \in \mathbb{H}$*.*
- $\mathit{DLVrfy}(\phi, A, B) \to \{0, 1\}$*: детерминированный алгоритм верификации доказательства, который при наличии вводного доказательства* ϕ*, точки* $A \in \mathbb{G}$ *и точки* $B \in \mathbb{H}$ *выдаёт 1, если (и только в том случае)* (A, B) *имеют общий дискретный логарифм в соответствующих группах.*

Параметры кривой. Bitcoin и Monero используют разные эллиптические кривые. Bitcoin в соответствии со *стандартами эффективной криптографии (SEC)* использует кривую `secp256k1` и алгоритм `ECDSA`. Monero, в основе которой лежит вторая версия протокола CryptoNote [10], использует кривую `curve25519`, далее также называемую `edward25519`, предложенную Дэниелом Дж. Бернштейном [6].

Мы обозначаем параметры

для кривой edward25519 как

$$\begin{aligned} &q : \text{простое число; } q = 2^{255} - 19 \\ &d : \text{элемент } \mathbb{F}_q;\ d = -121665/121666 \\ &\mathcal{E} : \text{уравнение эллиптической кривой; } -x^2 + y^2 = 1 + dx^2y^2 \\ &G : \text{базовая точка; } G = (x, -4/5) \\ &l : \text{порядок базовой точки; } l = 2^{252} + 27742317777372353535851937790883648493 \end{aligned} \tag{1}$$

для кривой secp256k1 как

$$\begin{aligned} &p : \text{простое число; } p = 2^{256} - 2^{32} - 977 \\ &a : \text{элемент } \mathbb{F}_p;\ a = 0 \\ &b : \text{элемент } \mathbb{F}_p;\ b = 7 \\ &\mathcal{E}' : \text{уравнение эллиптической кривой; } y^2 = x^3 + ax + b \\ &H : \text{базовая точка; } H = \\ &(\texttt{0x79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798}, \\ &\ \texttt{0x483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8}) \\ &n : \text{порядок базовой точки; } n = 2^{256} - 432420386565659656852420866394968145599 \end{aligned} \tag{2}$$

3.4 Одноразовая подпись VES ECDSA

Схема одноразовой подписи VES ECDSA предложена Фурнье и др. в работе [1] в качестве обобщения концепции адаптивных подписей в рамках схемы Шнорра [7] и ECDSA. Одноразовые верифицируемые зашифрованные подписи, далее именуемые нами одноразовыми подписями VES, построены таким образом, что при знании шифротекста и простого текста появляется возможность восстановления ключа шифрования.

Мы перечисляем некоторые из алгоритмов, определяемых в работе [1]:

Определение 2 (одноразовая подпись VES ECDSA). *Схема одноразовой подписи VES ECDSA содержит:*

- $\mathsf{EncSign}(sk_S, pk_E, m) \xrightarrow{\$} \hat{\sigma}$*: возможный вероятностный алгоритм создания зашифрованной подписи, который при наличии вводного секретного ключа подписи sk_S, публичного ключа шифрования pk_E и сообщения m выдаёт шифротекст $\hat{\sigma}$.*
- $\mathsf{EncVrfy}(pk_S, pk_E, m, \hat{\sigma}) \to \{0, 1\}$*: детерминированный алгоритм верификации зашифрованной подписи, который при наличии вводного публичного ключа подписи pk_S, публичного ключа шифрования pk_E, сообщения m и шифротекста $\hat{\sigma}$ выводит 1, если и только если $\hat{\sigma}$ является действительной зашифрованной версией подписи сообщения m для pk_S при pk_E.*
- $\mathsf{DecSig}(sk_E, \hat{\sigma}) \to \sigma$*: (как правило) детерминированный алгоритм расшифровки подписи, который при наличии вводного ключа расшифровки sk_E и действительного шифротекста $\hat{\sigma}$, соответствующего ключу шифрования, выдаёт действительную подпись σ.*
- $\mathsf{RecKey}(pk_E, \hat{\sigma}) \to \delta$*: детерминированный алгоритм выведения ключа восстановления, который на основе шифротекста $\hat{\sigma}$ и публичного ключа шифрования pk_E выводит ключ восстановления δ.*
- $\mathsf{Rec}(\sigma, \delta) \to sk_E$*: детерминированный алгоритм восстановления ключа расшифровки, который при наличии расшифрованной подписи σ и ключа восстановления δ, связанного с оригинальным шифротекстом, выдаёт секретный ключ расшифровки sk_E.*

3.5 Обобщение

Как было сказано в подпункте, связанном с Monero, требования к наличию примитивов в блокчейне отсутствуют. Мы только генерируем приватный ключ и используем адреса. А это значит, что данная схема может быть обобщена для пары криптовалют, одна из которых будет соответствовать предварительным условиям, связанным с Bitcoin, что делает эту схему очень агностической в отношении блокчейна, о чём говорится в работе [8] (при условии наличия вышеуказанных криптографических примитивов для базовых параметров блокчейнов). Тем не менее стоит отметить, что другие схемы будут проще и эффективней, если оба блокчейна в паре будут обладать возможностями реализации в блокчейне. Но такие схемы уже в целом были развёрнуты [3, 4, 9].

4 Протокол

Протокол работает следующим образом: Элис переводит Monero на определённый адрес. При этом каждый из участников владеет своей половиной приватного ключа траты (далее именуемой «частью»). Затем для раскрытия одной из половин приватного ключа траты (в зависимости от того, кто из участников забирает Bitcoin) используется язык сценариев Bitcoin и протоколы частичного отсутствия скриптов. В зависимости от того, кто раскрывает свою половину приватного ключа траты, Monero меняет владельца. Транзакции Bitcoin строятся таким образом, что в том случае, если участники будут следовать протоколу, при завершении свопа никаких потерь быть не должно.

Если сделка состоится, Элис потратит свой Bitcoin, раскрыв свою часть приватного ключа траты, что позволит Бобу потратить заблокированный Monero. Если сделка будет отменена, Боб потратит Bitcoin по истечении времени первой временной блокировки, раскрыв свою часть приватного ключа, что позволит Элис потратить Monero. В обеих случаях комиссии за проведение транзакции будут минимальными.

Обмен ключами выполняется при помощи одноразовой подписи VES `ECDSA` (а также адаптивных подписей) и применения равного дискретного логарифма в групповых доказательствах знания с нулевым разглашением. Одноразовые подписи VES строятся таким образом, что при наличии шифротекста и расшифрованной подписи ключ расшифровки легко восстанавливается. При выборе части приватного ключа (половины полного приватного ключа траты Monero) в качестве ключа расшифровки мы получаем способ автоматической продажи части приватного ключа другому участнику. Таким образом, поскольку Bitcoin и Monero используют различные кривые, нам необходимо доказать отношение между точками на `edward25519` и `secp256k1`, чтобы гарантировать отсутствие в необходимости доверия в рамках протокола.

4.1 Неинтерактивный возврат

Если Элис или Боб заблокировали свои средства, но один из них в определённый момент прерывает своп или связь, протокол не должен требовать наличия интерактивности для выполнения процедуры возврата средств обоим участникам. В противном случае Элис, не отвечая, сделает Боба заложником, которому придётся ждать истечения срока второй временной блокировки, чтобы забрать Bitcoin. При надлежащим образом прерванном свопе Боб должен раскрыть свою часть приватного ключа Monero, что позволит Элис вернуть свои Monero при помощи одноразовой подписи VES.

Одноразовая подпись VES `ECDSA` является интерактивной: один из участников должен предоставить зашифрованную подпись, и если её верификация будет успешной, другой участник в ответ предоставит действительную подпись в соответствии со схемой мультиподписи Bitcoin «2 из 2» (протокол частичного отсутствия скриптов), что позволит первому участнику расшифровать и опубликовать две действительные подписи по схеме «2 из 2», а второму — узнать (в блокчейне) расшифрованную подпись и восстановить ключ расшифровки.

Как упоминалось выше, процесс возврата не должен быть интерактивным: протокол разработан таким образом, что Элис становится известна расшифрованная подпись возврата Боба, а сама она предоставляет Бобу действительную подпись, построенную по схеме мультиподписи «2 из 2», до блокировки средств. В случае возврата Боб сможет расшифровать и опубликовать подпись без участия Элис.

4.2 Приватные ключи Monero

Приватными ключами Monero являются две скалярные величины `edward25519`: первая — это приватный ключ просмотра, а вторая — приватный ключ траты. Для обозначения приватных ключей мы используем прописные буквы, а для обозначения публичных ключей — заглавные:

$$X = xG$$

где G является генератором группы $\mathbb{G}$. Мы обозначаем

(i) приватный ключ k^v как полный приватный ключ просмотра;

(ii) K^v как полный публичный ключ просмотра;

(iii) k_a^v как часть приватного ключа просмотра, принадлежащую Элис, и k_b^v как часть, принадлежащую Бобу;

(iv) приватный ключ k^s как полный приватный ключ траты;

(v) K^s как полный публичный ключ траты;

(vi) и k_a^s как часть приватного ключа траты, принадлежащую Элис, и k_b^s как часть, принадлежащую Бобу.

Частичные ключи. Мы обозначаем части приватного ключа k_a^s и k_b^s как

$$k_a^s + k_b^s \equiv k^s \pmod{l}$$

А затем

$$\begin{aligned} k_a^s G &= K_a^s \\ k_b^s G &= K_b^s \\ K_a^s + K_b^s = (k_a^s + k_b^s)G &= k^s G = K^s \end{aligned} \tag{3}$$

То же касается k^v с k_a^v и k_b^v.

4.3 Доказательства с нулевым разглашением

Доказательства с нулевым разглашением необходимы изначально, чтобы протокол не требовал доверия. Протокол использует одноразовую подпись VES, чтобы раскрыть части приватного ключа, но мы не можем проверить равенство дискретного логарифма для части публичного ключа Monero и публичного ключа расшифровки Bitcoin другого участника, пока он не попадёт в блокчейн. Следовательно, нам необходимо доказательство того, что дискретный логарифм одинаков для двух групп $\mathbb{G}$ и $\mathbb{H}$.

Равенство дискретного логарифма в группах. Элис и Боб должны доказать друг другу, используя

$$\begin{aligned} &k_i^s \leftarrow \text{скалярные величины } \texttt{edward25519} \text{ и } \texttt{secp256k1} \text{ с эквивалентным двоичным представлением} \\ &K_i^s = k_i^s G \in \mathbb{G} \\ &B_i^s = k_i^s H \in \mathbb{H} \end{aligned} \tag{4}$$

для $i \in \{a, b\}$, при K_i^s и B_i^s

$$\exists k_i^s \mid K_i^s = k_i^s G \wedge B_i^s = k_i^s H \wedge k_i^s < \mathsf{min}(l, n) \tag{5}$$

4.4 Временные параметры

На этапе инициализации задаются две временные блокировки: t_0, t_1. t_0 определяет временное окно, в рамках которого может быть совершён безопасный обмен. По истечении t_0 может быть запущен процесс возврата, что сделает обмен небезопасным, поскольку создаётся условие для конкуренции (несмотря на то, что это довольно сложно реализовать в реальности). t_1 определяет время ответа, в течение которого Боб должен отреагировать, раскрыть свою часть приватного ключа Monero и получить свой Bitcoin обратно, что также позволит Элис вернуть её Monero (если Monero были заблокированы). По истечении t_1 Элис сможет забрать Bitcoin в одностороннем порядке.

4.5 Скрипты Bitcoin

Со стороны Bitcoin необходимо два скрипта: первый используется для завершения свопа или запуска процесса возврата (известен как `SWAPLOCK`), второй — для завершения процесса возврата (известен как `REFUND`). При успешном проведении свопа второй скрипт не передаётся в блокчейн и не используется. Каждый скрипт определяет два возможных пути, и мы последовательно разъясним все четыре возможных пути (*покупка*, *возврат*, *трата* и *получение*) траты двух `UTXOs`.

SWAPLOCK является скриптом, используемым для блокировки средств, и определяет два базовых пути выполнения: (1) выполнение свопа (успешный путь) и (2) возврат (неуспешный путь). Мы определяем скрипт SWAPLOCK следующим образом:

SWAPLOCK **покупки.** Элис получает контроль над Bitcoin и раскрывает свою долю ключа Monero Бобу, используя σ_1 (одноразовую подпись VES, раскрывающую k_a^s), что даёт Бобу контроль над Monero. $\texttt{BTX}_{buy}$ выполняет SWAPLOCK при помощи:

SWAPLOCK **возврата.** SWAPLOCK, подписанный обоими участниками, возвращается, и средства перемещаются в скрипт REFUND. $\texttt{BTX}_{refund}$ выполняет SWAPLOCK при помощи:

REFUND. REFUND является скриптом, используемым в том случае, если своп уже происходит в блокчейне, но прерывается. Данный скрипт возврата используется для перемещения средств из скрипта SWAPLOCK при помощи мультиподписи, построенной по схеме «2 из 2», с временной блокировкой. Мы определяем скрипт REFUND следующим образом:

REFUND **траты.** Боб прерывает своп и раскрывает свою часть приватного ключа Monero, используя σ_1' (одноразовую подпись VES, раскрывающую k_b^s), что возвращает Элис контроль над её Monero. $\texttt{BTX}_{spend}$ выполняет REFUND следующим образом:

REFUND **получения.** Элис получает контроль над Bitcoin по истечении обеих временных блокировок, не раскрывая своей части приватного ключа Monero. В результате Боб теряет деньги, так как не следует протоколу. $\texttt{BTX}_{claim}$ выполняет REFUND следующим образом:

4.6 Транзакции

Ниже мы описываем и называем транзакции Bitcoin и Monero, необходимые для выполнения всего протокола.

$\texttt{BTX}_{\boldsymbol{lock}}$**,** транзакция Bitcoin, содержащая ≥ 1 входов от Боба и первый выход (vout: 0) для скрипта SWAPLOCK и возможных выходов сдачи.

$\texttt{BTX}_{\boldsymbol{buy}}$**,** транзакция Bitcoin, содержащая 1 вход и использующая скрипт SWAPLOCK ($\texttt{BTX}_{lock}$, vout: 0) с мультиподписью, построенной по схеме «2 из 2», в соответствии с протоколом частичного отсутствия скриптов и ≥ 1 выходов.

$\texttt{BTX}_{\boldsymbol{refund}}$**,** транзакция Bitcoin, содержащая 1 вход и использующая скрипт SWAPLOCK script ($\texttt{BTX}_{lock}$, vout: 0) с мультиподписью, построенной по схеме «2 из 2», с временной блокировкой и ровно одним выходом для скрипта REFUND.

$\texttt{BTX}_{\boldsymbol{spend}}$**,** транзакция Bitcoin, содержащая 1 вход и использующая скрипт REFUND ($\texttt{BTX}_{refund}$, vout: 0) с мультиподписью, построенной по схеме «2 из 2», в соответствии с протоколом частичного отсутствия скриптов и ≥ 1 выходов.

$\texttt{BTX}_{\boldsymbol{claim}}$**,** транзакция Bitcoin, содержащая 1 вход и использующая скрипт REFUND ($\texttt{BTX}_{refund}$, vout: 0) с подписью Элис и ≥ 1 выходов.

$\texttt{XTX}_{\boldsymbol{lock}}$**,** транзакция Monero, в которой средства переводятся на определённый адрес (K^v, K^s).

$\texttt{XTX}_{\boldsymbol{buy}}$**,** транзакция Monero, в которой тратятся средства, находящиеся по адресу (K^v, K^s).

4.7 Полная последовательность выполнения протокола

Ниже мы описываем полную последовательность выполнения протокола, позволяющую успешно завершить своп с вычислением и взаимным использованием всей необходимой информации в случае, если в какой-то момент один из участников перестанет отвечать или же запустит процесс возврата.

Во время первого раунда обмена данными, поскольку обе стороны используют общие параметры, необходимые для запуска протокола, благодаря применению равного дискретного логарифма в доказательстве знания с нулевым разглашением мы сможем избежать таких схем, как схема обязательства с раскрытием, даже с $K^s = K_a^s + K_b^s$. Никто не сможет произвольным образом выбрать K^s и вычислить действительное доказательство z_i. Тем не менее, чтобы при каждом выполнении гарантировано выбирался случайный ключ просмотра, необходимо добавление схемы обязательства с раскрытием по k_i^v. Таким образом, не добавляя ещё один раунд обмена данными, мы также можем добавить схему обязательства с раскрытием по k_i^s.

Мы определяем некоторые вспомогательные алгоритмы, позволяющие инициализировать, подписать и верифицировать транзакции Bitcoin и Monero.

- InitTx(): Общий и детерминированный алгоритм, который (при наличии ряда вводных параметров) создаёт действительную инициализированную транзакцию.
- Sign(): Общий и вероятностный алгоритм для подписания транзакции, который (при наличии приватного ключа и инициализированной транзакции) выдаёт действительную подпись транзакции.
- VrfyTx(): Общий и детерминированный алгоритм, который (при наличии ряда транзакций и параметров) выдаёт 1, если и только если транзакции являются действительными в соответствии с правилами протокола и алгоритмом консенсуса блокчейна.
- Vrfy(): Общий и детерминированный алгоритм, который (при наличии публичного ключа, транзакций и подписи) выдаёт 1, если и только если подпись является действительной транзакций, соответствующей публичному ключу.
- PubTx(): Общий алгоритм для публикации транзакций в сети.
- WatchTx(): Общий алгоритм ожидания подтверждения транзакций.
- RecSig(): Общий алгоритм выделения подписей транзакций.

Элис (XMR→BTC)		Боб (BTC→XMR)
$k_a^v, k_a^s \xleftarrow{R} [1, l-1]$		$k_b^v, k_b^s \xleftarrow{R} [1, l-1]$
$b_a, b_a^r \xleftarrow{R} [1, n-1]$		$b_b, b_b^r \xleftarrow{R} [1, n-1]$
$B_a \leftarrow b_a H$		$B_b \leftarrow b_b H$
$B_a^r \leftarrow b_a^r H$		$B_b^r \leftarrow b_b^r H$
$(z_a, K_a^s, B_a^s) \leftarrow \mathsf{DLProve}(k_a^s)$		$(z_b, K_b^s, B_b^s) \leftarrow \mathsf{DLProve}(k_b^s)$
		$s \xleftarrow{R} [0, 2^{256}]$
		$h_s \leftarrow \mathtt{SHA256}(s)$
	$\xleftrightarrow{\langle k_i^v, K_i^s, B_i, B_i^s, B_i^r, z_i, h_s \rangle \ \forall i \in \{a,b\}}$	
	$k^v \equiv k_a^v + k_b^v \pmod{l}$	
	$K^v = k^v G, \ \ K^s = K_a^s + K_b^s$	
$\mathsf{DLVrfy}(K_b^s, B_b^s, z_b) \stackrel{?}{=} 1$		$\mathsf{DLVrfy}(K_a^s, B_a^s, z_a) \stackrel{?}{=} 1$
		$(\mathtt{BTX}_{lock}, \mathtt{BTX}_{refund}) \leftarrow \mathsf{InitTx}(B_a, B_b, B_a^r, B_b^r)$
		$\sigma_r' \leftarrow \mathsf{Sign}(b_b^r, \mathtt{BTX}_{refund})$
	$\xleftarrow{\langle \mathtt{BTX}_{lock}, \mathtt{BTX}_{refund}, \mathtt{BTX}_{spend}, \sigma_r' \rangle}$	
$\mathsf{VrfyTx}(\mathtt{BTX}_{lock}, \mathtt{BTX}_{refund}, B_a, B_b, B_a^r, B_b^r, h_s) \stackrel{?}{=} 1$		
$\mathsf{Vrfy}(B_b^r, \mathtt{BTX}_{refund}, \sigma_r') \stackrel{?}{=} 1$		
$\hat{\sigma}_1' \leftarrow \mathsf{EncSign}(b_a^r, B_b^s, \mathtt{BTX}_{spend})$		
$\delta' \leftarrow \mathsf{RecKey}(B_b^s, \hat{\sigma}_1')$		
$\sigma_r'' \leftarrow \mathsf{Sign}(b_a^r, \mathtt{BTX}_{refund})$		
	$\xrightarrow{\langle \sigma_r'', \hat{\sigma}_1' \rangle}$	
		$\mathsf{EncVrfy}(B_a^r, B_b^s, \mathtt{BTX}_{spend}, \hat{\sigma}_1') \stackrel{?}{=} 1$
		$\mathsf{Vrfy}(B_a^r, \mathtt{BTX}_{refund}, \sigma_r'') \stackrel{?}{=} 1$
		$\mathtt{BTX}_{buy} \leftarrow \mathsf{InitTx}(\mathtt{BTX}_{lock})$
		$\hat{\sigma}_1 \leftarrow \mathsf{EncSign}(b_b, B_a^s, \mathtt{BTX}_{buy})$
		$\delta \leftarrow \mathsf{RecKey}(B_a^s, \hat{\sigma}_1)$
		$\mathsf{PubTx}(\mathtt{BTX}_{lock})$
	$\xleftarrow{\langle \mathtt{BTX}_{buy}, \hat{\sigma}_1 \rangle}$	
$\mathsf{EncVrfy}(B_b, B_a^s, \mathtt{BTX}_{buy}, \hat{\sigma}_1) \stackrel{?}{=} 1$		
$\mathsf{WatchTx}(\mathtt{BTX}_{lock}) \stackrel{?}{=} 1$		
	$\cdots$	
$\mathtt{XTX}_{lock} \leftarrow \mathsf{InitTx}(K^v, K^s)$		
$\mathsf{PubTx}(\mathtt{XTX}_{lock})$		
	$\cdots$	
		$\mathsf{WatchTx}(K^v, K^s)$ w/ $(k^v, K^s) \stackrel{?}{=} 1$
	$\xleftarrow{\langle s \rangle}$	
$\sigma_1 \leftarrow \mathsf{DecSig}(k_a^s, \hat{\sigma}_1)$		
$\sigma_2 \leftarrow \mathsf{Sign}(b_a, \mathtt{BTX}_{buy})$		
$\sigma := (\sigma_1, \sigma_2)$		
$\mathsf{PubTx}(\mathtt{BTX}_{buy}, \sigma, s)$		
		$\mathsf{WatchTx}(\mathtt{BTX}_{buy}) \stackrel{?}{=} 1$
	$\cdots$	
		$(\sigma_1, \sigma_2) \leftarrow \mathsf{RecSig}(\mathtt{BTX}_{buy})$
		$k_a^s \leftarrow \mathsf{Rec}(\sigma_1, \delta)$
		$k^s \equiv k_a^s + k_b^s \pmod{l}$
		$\langle k^v, k^s \rangle$

Рис. 1. Выполнение протокола при успешном выполнении свопа между Бобом и Элис

Точками показано время синхронизации, за которое один или несколько участников должны дождаться подтверждения транзакций перед тем, как продолжится выполнение протокола. Уровень безопасности (то есть количество подтверждений, необходимое каждому участнику) является локальным параметром, но должен устанавливаться в соответствии с параметрами синхронизации (которые являются глобальными для обоих участников) во избежание ситуации, когда время синхронизации будет меньшим или равным параметру безопасности.

5 Дальнейшие исследования

Настоящий протокол может быть реализован с существующими версиями Bitcoin и Monero, но для использования возможностей алгоритма `Schnorr` и упрощения в целях выработки более эффективных вариантов протокола с меньшей опорой на блокчейн требуются дополнительные исследования. Это позволит повысить анонимность атомных свопов в плане анализа блокчейна.

Как уже говорилось, данный протокол может быть адаптирован для применения с другими криптовалютами. Некоторые из них, возможно, пока не предполагают возможности проведения атомных свопов, и расширение протокола для других пар может способствовать децентрализации. Такое расширение только для криптовалют, ориентированных на обеспечение анонимности (таких как Monero с Mimblewimble), является следующей задачей с точки зрения развития атомных свопов.

Интеграция протокола в сервисы или децентрализованные биржи будет также способствовать демократизации торговли за счёт отсутствия необходимости в привлечении доверенных сторон и может повысить ликвидность рынка. Так как предлагаемое решение является асимметричным, построение соответствующих сервисов будет не простым.

5.1 Известные ограничения

Чтобы доказать жизнеспособность (если по крайней мере один участник по-прежнему находится онлайн), мы рассматриваем наихудший сценарий, при котором участник может потерять средства (будучи неспособным забрать средства из другого блокчейна). Это может произойти в случае, если участники не будут следовать протоколу, например, оставаться онлайн во время ожидания завершения свопа или при своевременном получении средств. Обоснование решения приводится в подпункте 2.3.

Комиссии в разных блокчейнах будут отличаться из-за внутренних параметров этих блокчейнов и сложности проводимых транзакций, а также из-за внешних факторов, таких как требование к занимаемому блоками месту. Следует отметить, что в рамках настоящего протокола блокчейн Bitcoin используется как механизм принятия решений, когда мы используем возможности скриптов (несмотря на то что мы пытаемся реализовать максимум логических решений вне блокчейна), что является причиной увеличения размера транзакций со стороны Bitcoin. В совокупности эти два фактора делают транзакции Bitcoin в целом более затратными, чем транзакции в блокчейне Monero.

Трудно достичь немедленной реакции пользователей при совершении атомных свопов между блокчейнами. Низкая скорость работы блокчейна и количество подтверждений, необходимых для завершения транзакций, определяют скорость выполнения протокола, в некоторых случаях приводя к опережению. Тем не менее во избежание подобных ситуаций протокол также может быть расширен определённым образом. Стоит отметить, что один участник не может вызвать опережения выполнения другим участником, что делает возможным наихудший сценарий с потерей комиссий за проведение транзакций в соответствующих блокчейнах участников.

6 Благодарность

Мы выражаем благодарность Исследовательской лаборатории Monero и Сарангу Ноезеру за их полезные комментарии, внесённые при окончательной редакции данной работы. Эти исследования поддерживались и частично финансировались Сообществом Monero, и мы отдельно благодарим тех, кто вносил свои пожертвования. Наконец мы благодарим сотрудников TrueLevel SA за начальное финансирование, их вклад и комментарии.

Список литературы

[1] Lloyd Fournier. *One-Time Verifiably Encrypted Signatures, A.K.A. Adaptor Signatures (Одноразовые верифицируемые зашифрованные подписи, также известные как адаптивные подписи)*. 2019. URL: https://github.com/LLFourn/one-time-VES/blob/master/main.pdf.

[2] Sarang Noether. *Discrete logarithm equality across groups (Равенство дискретных логарифмов в группах)*. 2018. URL: https://web.getmonero.org/resources/research-lab/pubs/MRL-0010.pdf.

[3] Tier Nolan. *Alt chains and atomic transfers (Альтернативные блокчейны и атомные переводы)*. URL: https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949.

[4] Andrew Poelstra. *Adaptor Signatures and Atomic Swaps from Scriptless Scripts (Адаптивные подписи и атомные свопы в рамках протокола Scriptless Scripts)*. 2017. URL: https://github.com/ElementsProject/scriptless-scripts/blob/master/md/atomic-swap.md.

[5] Andrew Poelstra. *Scriptless Scripts (Протокол Scriptless Scripts)*. 2017. URL: https://download.wpsoftware.net/bitcoin/wizardry/mw-slides/2017-05-milan-meetup/slides.pdf.

[6] Certicom Research. *SEC 2: Recommended Elliptic Curve Domain Parameters (Исследования Certicom. РАЗДЕЛ 2: Рекомендуемые доменные параметры эллиптических кривых)*. 2010. URL: http://www.secg.org/sec2-v2.pdf.

[7] Claus-Peter Schnorr. "Efficient Identification and Signatures for Smart Cards (Эффективная идентификация и подписи для смарт-контрактов)". в: *Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*. CRYPTO '89. Berlin, Heidelberg: Springer-Verlag, 1990, с. 239—252. ISBN: 3-540-97317-6. URL: http://dl.acm.org/citation.cfm?id=646754.705037.

[8] Ruben Somsen. *SAS: Succinct Atomic Swap (SAS: сжатый атомный своп)*. URL: https://gist.github.com/RubenSomsen/8853a66a64825716f51b409be528355f.

[9] Lucas Sorianos del Pino и Lloyd Fournier. *Grin-Bitcoin Atomic Swap (Атомные свопы Grin-Bitcoin)*. 2019. URL: https://github.com/comit-network/grin-btc-poc/blob/master/spec.pdf.

[10] Nicolas Van Saberhagen. *CryptoNote v 2.0*. 2013.