|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *[Шпаргалка](https://github.com/baro77/RctCS)*  *[Версия от](https://github.com/baro77/RctCS)*  *[20210604](https://github.com/baro77/RctCS)*  Концептуальные (неупорядоченные) компоненты **RCTTypeCLSAG (тип 5)** | | | | | | | |
| **Вход** (здесь обозначение **для каждого фактического** входящего UTXO) | | | | | **UTXO выход “ t ”** | **Комиссия за транзакцию** | **Время разблокировки выходов** |
|  | ложное UTXO  **смещение** | **фактическое** UTXO  **смещение** | ложные UTXOs  **смещения** |  | скрытый адрес **X** | **f**  простое значение, чтобы майнеры могли оценить его  в обязательствах выражено как “**a**” в **атомных единицах** (1/1012 XMR) | абсолютное время, выраженное как:   * **Блокировка отключена** (0) * **Высота блока**  (< 500.000.000) * **Время периода** (≥ 500.000.000)   Протоколом также предусмотрен относительный период блокировки на **10 блоков** |
| уровень j=1  Публичные CLSAG ключи: **X i** | **X …** | **X i = 𝜋** | **X …** | подпись  CLSAG  (**c 1** , **r i**)  Подписывается  вся транзакция, исключение -  поля подписи | обязательство **C ≜ b G + a J**  где:  **a** : кол-во Monero в UTXO (**простое** значение)  **b** ≜ H( “commitment\_mask” , H(r Vi , t) )  = H( “commitment\_mask” , H(v0 R , t) )  обычная “платёжная хитрость” отправителя/получателя при наличии  транзакции и ключей просмотра |
| уровень j=2  Публичные CLSAG ключи:  **C i - C pseudo** | (C i = 𝜋 – C pseudo) = (b - b’) G  Способность подписаться, используя эту EC пару из публичного/приватного ключа, доказывает, что обязательства C i = 𝜋 и C pseudo относятся к одному “a”  **C …** | **C i = 𝜋 = b G + a J** | **C …** | **Bulletproof** | **Дополнение** |
| **C pseudo ≜ b’G + a J** | | | Доказательства диапазона “a”, которых следует избегать с доказательствами по выходу  входы выходы  5 **J** + 6 **J** = 21 **J** + (**𝑙** -10) **J**  = (21-10) **J** + **𝑙 J**  *Продолжение следует…*  Инфляция по переполнению  циклической группы ↻ | * Публичный ключ транзакции (ий) **R** * Зашифрованный **payID** (если присутствует) * …   ПРИМЕЧАНИЕ: слабо структурированное поле, подписанное, но не являющееся частью консенсуса (⟹ задача для кошельков) |
| Образы ключей: |  | действительный **X\***  артефакт **C\***  ( ⟹ CLSAG’s  pseudo **W\*** ) |  | | **сумма**  **Зашифрованное** значение:  ≜ **a** ⊕ H( “amount” , H(r Vi , t) )  = **a** ⊕ H( “amount” , H(v0 R , t) ) |

|  |  |  |
| --- | --- | --- |
| **Обязательства Педерсена 101** | **Обзор CLSAG и текущей реализации** | **Секретный поток значений, передаваемых с UTXOs Monero** |
| Обязательства являются способом привязки к значению без его раскрытия (возможно, раскрытие при этом произойдёт позже). Обязательства Педерсена в форме EC выглядят следующим образом:  **C ≜ b G + a J**  где:  **a** : значение, по которому даётся обязательство;  **b** : случайный “ослепляющий фактор”, отвечающий за создание энтропии, чтобы затруднить вычисление “a”посредством радужной таблицы;  **G** : общая базовая точка;  **J**  : точка на EC, для которой значение “j”, находящееся в пределах J ≜ j G, неизвестно (DLP).  СВОЙСТВА  **Теоретически скрывает сумму** : множество пар (a,b) может дать  одно и то же значение обязательства C  **Обязательное вычисление**: попытка предоставления  обязательства по ложному значению эквивалентна  решению системы DLP:  C(a2 , b2) = C(a , b) ⟹ j = (b – b2)/(a2 – a) ⟹ DLP решена  **Гомоморфизм**: C(a1 , b1) + C(a2 , b2) = C(a1+a2 , b1+b2) | * CLSAG является видом схемы мультиподписи MLSAG, где каждый “сжатый” уровень имеет свой собственный фактический ключ и набор ложных ключей * В отличие от MLSAG фактический образ ключа, защищающий от двойной траты, доступен только для первого уровня, поэтому каждый входящий UTXO требует наличия отдельной CLSAG (но и MLSAG использовались подобным образом в целях обеспечения анонимности) * В рамках протокола RingCT используется двухуровневая схема CLSAG, где подпись 2-го уровня служит доказательством эквивалентности значений фактических UTXO, по которым дано обязательство, и обязательством по псевдовыходу Cpseudo : эквивалентность считается доказанной *при* наличии новой специфической пары, состоящей из публичного/приватного ключей на эллиптической кривой, поэтому подписание 2-го уровня CLSAG при помощи такой пары доказывает существование этой пары и их эквивалентность. | Значение UTXO Monero зашифровывается в двух полях: в поле обязательства **C** и поле **суммы**:   * поле **суммы** позволяет отправителю и получателю совместно использовать значение: отправитель вычисляет его на основе простого значения **a** в соответствии с приведённым определением; получатель может проверить простое значение **a** путём применения операции XOR к сумме (**amount**) с использованием той же хеш-функции, что использовалась в рамках определения (две равные XOR исключают друг друга). Чтобы хеш мог вычислить как отправитель, так и получатель, транзакция и ключи просмотра используются подобно тому, как это происходит в случае с протоколом Диффи-Хеллмана при создании скрытых адресов; * обязательства **C** позволяют каждому узлу в сети убедиться в том, что значения суммы входящих UTXO транзакции равны сумме значений выходящих UTXO транзакции с добавлением комиссии. При этом фактические значения остаются скрытыми (кроме значения комиссии, которое изначально является простым).   В рамках схемы RingCT:   * отправитель проверяет, чтобы исходящие сумма (**amount**) и обязательство **C** соответствовали одному и тому же значению **a** (это обязывает отправителя/получателя обменяться значением, соответствующим тому, что проверяется сетью). Это возможно благодаря уже задокументированному методу определения суммы (**amount**) и **b**; * для каждого входа отправителем определяется обязательство по псевдовыходу Cpseudo, связанное с тем же значением «a» фактического UTXO (при этом не раскрывается его место в кольце, поскольку оно отделяется при помощи подписи CLSAG); * таким образом, для проверки правильности потока значений узлам в сети требуется проверить только баланс и доказательство Bulletproof:   Проверка Bullet  proof  ⟹ Σ inputs a J = Σ outputs a J + f J  Σ inputs a = Σ outputs a + f  Σ inputs Cpseudo = Σ outputs C + f J  Σ inputs b’ = Σ outputs b (задаётся отправителем) |

* Множество постов на [Monero Stack Exchange](https://monero.stackexchange.com/) (например, “[Полная структура дополнительного поля...](https://monero.stackexchange.com/questions/11888/complete-extra-field-structure-standard-interpretation)”)
* [API блоков Monero](https://localmonero.co/blocks/api) (например, [данный запрос](https://localmonero.co/blocks/api/get_transaction_data/b0b3c299bd14f846bed560ac457b6fb8b84184c72629aad81910fc33526d4949) проверки транзакции TX – предлагается разбор JSON посредством [jq](https://stedolan.github.io/jq/))
* [Исходный код](https://github.com/monero-project/monero) Monero CLI (например, [/src/ringct/rctTypes.h](https://github.com/monero-project/monero/blob/master/src/ringct/rctTypes.h) с новым 5 типом подписи), будем надеяться, прошедший проверку специальным ПО (например, [Sourcetrail](https://www.sourcetrail.com/))

Перевод шпаргалки выполнил [v1docq47](https://github.com/v1docq47)

|  |  |
| --- | --- |
| **Примечания**  **и ссылки** | * Данная шпаргалка связана с предыдущими шпаргалками по [адресам Monero](https://github.com/baro77/MoneroAddressesCS) и [кольцевым подписям](https://github.com/baro77/RingsCS) * [От нуля к Monero: Вторая](https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf) редакция, главы 5, 6 и приложения A, B (даже если речь идёт о транзакциях,   подписанных при помощи MLSAG, CLSAG является простым способом, если вы понимаете  разницу между двумя видами колец).   * [От доказательств с нулевым разглашением до Bulletproofs](https://github.com/AdamISZ/from0k2bp) (первые 6 страниц, посвящённых   обязательствам). |