К вопросу построения безопасных протоколов функционирования блокчейн-систем

Григорий Маршалко,
Антон Гуселев,
Иван Лавриков

— ТК 026 —

РусКрипто 2018

С чем будем иметь дело

Что такое блокчейн?

По сути своей, блокчейн – это на удивление простая и элегантная структура данных. Фактически это просто список связанных блоков, но с одним важным дополнением – каждый из них содержит в себе криптографический хэш предыдущего блока. Таким образом, создается цепочка блоков и их индивидуальных отпечатков, вплоть до самого первого, которую, по сути, невозможно изменить.[a]

[a] *Блокчейн: революция, к которой мы не готовы*, habrahabr.ru, 2017

По сути... можно ли изменить? И при чем тут криптография и хэш... а подпись?

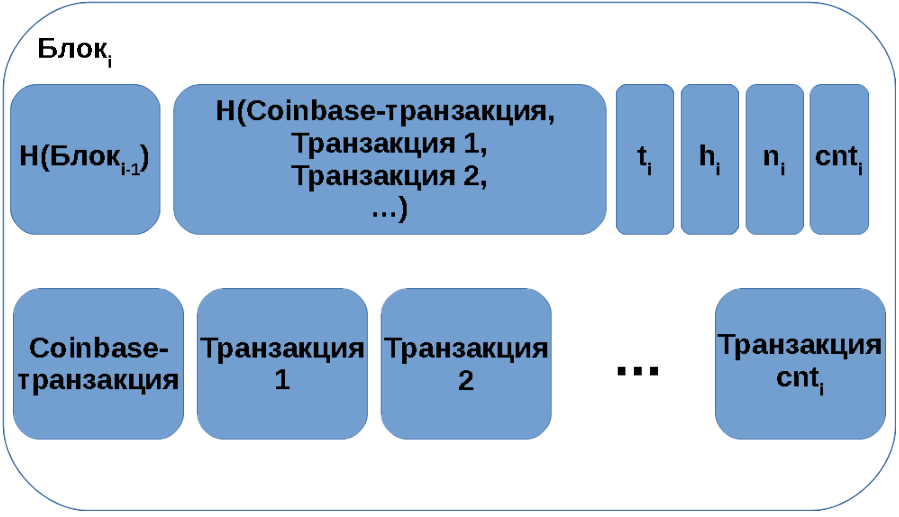
С чем будем иметь дело

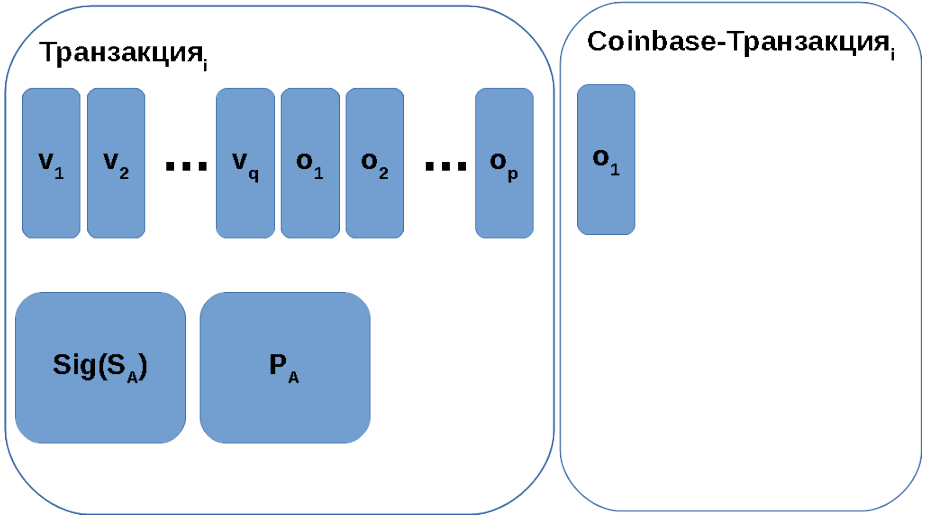
Что такое блокчейн?

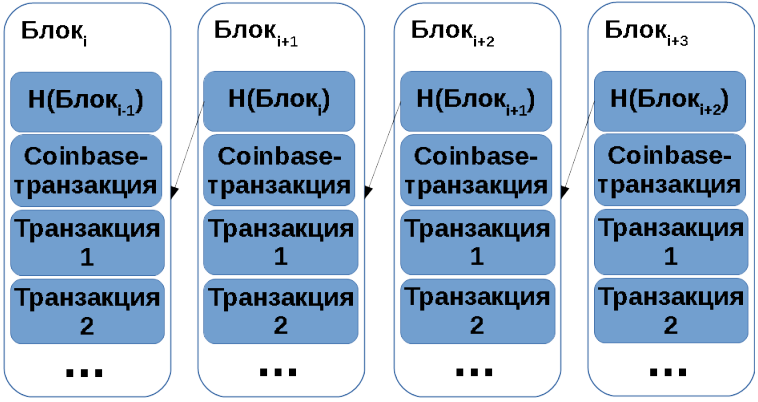
По сути своей, блокчейн – это на удивление простая и элегантная структура данных. Фактически это просто список связанных блоков, но с одним важным дополнением – каждый из них содержит в себе криптографический хэш предыдущего блока. Таким образом, создается цепочка блоков и их индивидуальных отпечатков, вплоть до самого первого, которую, по сути, невозможно изменить.[a]

[a] *Блокчейн: революция, к которой мы не готовы*, habrahabr.ru, 2017

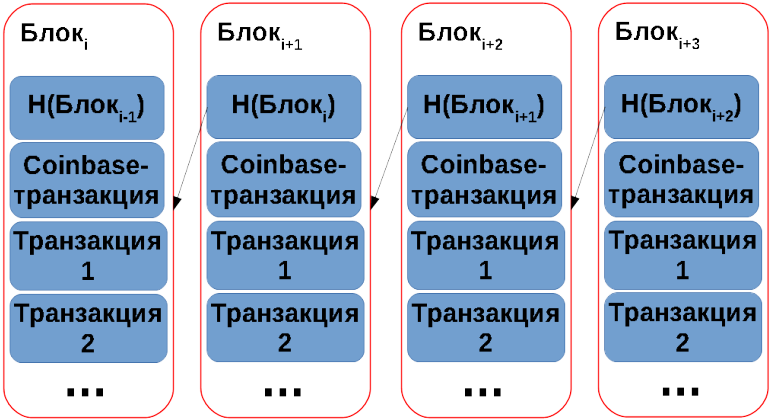
По сути... можно ли изменить? И при чем тут криптография и хэш... а подпись?


Блок$_i$
H(Блок$_{i-1}$)
H(Coinbase-транзакция, Транзакция 1, Транзакция 2, ...)
t_i
h_i
n_i
cnt_i
Coinbase-транзакция
Транзакция 1
Транзакция 2
...
Транзакция cnt_i

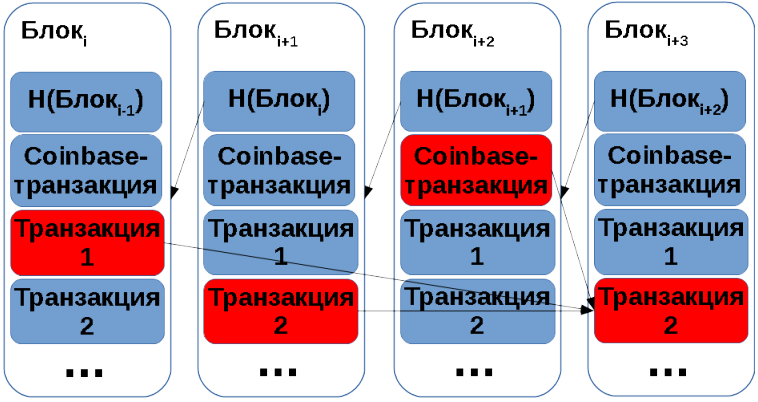

Транзакция$_i$
V_1
V_2
...
V_q
O_1
O_2
...
O_p
Sig(S_A)
P_A
Coinbase-Транзакция$_i$
O_1


Блок$_i$
H(Блок$_{i-1}$)
Coinbase-транзакция
Транзакция 1
Транзакция 2
...
Блок$_{i+1}$
H(Блок$_i$)
Coinbase-транзакция
Транзакция 1
Транзакция 2
...
Блок$_{i+2}$
H(Блок$_{i+1}$)
Coinbase-транзакция
Транзакция 1
Транзакция 2
...
Блок$_{i+3}$
H(Блок$_{i+2}$)
Coinbase-транзакция
Транзакция 1
Транзакция 2
...

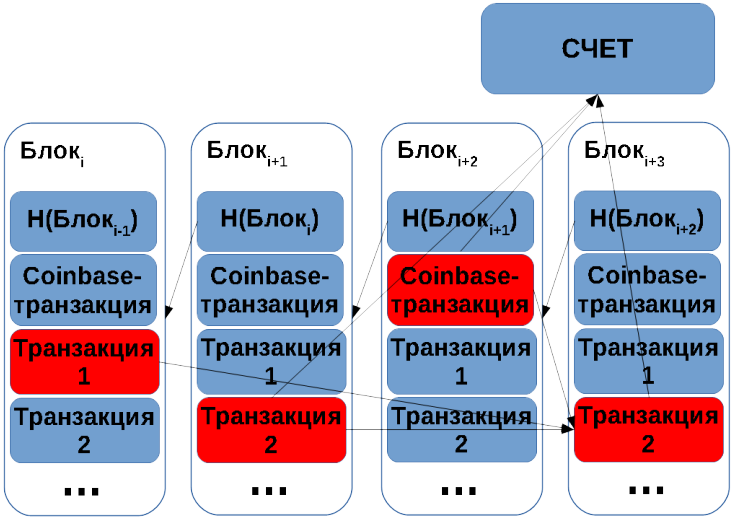
Логический реестр блоков



Логический реестр транзакций



Логический реестр счетов



Экономическая основа

- Для пользователя (если он гик или киберпанк) важен реестр счетов.
- *Желательно* увеличить баланс счета.
- Ну... и чтобы все **работало**.

Где взять коины? – В реестре!

Компания Chainalysis, занимающаяся изучением блокчейна, провела исследование и обнаружила, что за все время существования биткоина в мире пропало от 2,78 до 3,79 миллиона единиц этой криптовалюты. Получается, что исчезло от **17** до **23**% всех биткоинов[a].

[a] https://rb.ru/story/lost-bitcoins/

Как взять коины? – Сделать новый реестр!

- Он должен быть **допустимым** – не противоречить правилам системы.
- Его нужно сделать за **реальное** время.

Где взять коины? – В реестре!

Компания Chainalysis, занимающаяся изучением блокчейна, провела исследование и обнаружила, что за все время существования биткоина в мире пропало от 2,78 до 3,79 миллиона единиц этой криптовалюты. Получается, что исчезло от **17** до **23**% всех биткоинов[a].

[a] https://rb.ru/story/lost-bitcoins/

Как взять коины? – Сделать новый реестр!

- Он должен быть **допустимым** – не противоречить правилам системы.
- Его нужно сделать за **реальное** время.

Технические предпосылки

- Протокол консенсуса контролирует хэшрейт в течение только одной *эпохи* (пересчет хэшрейта осуществляется каждые 2016 блоков).
- **Никто** не контролирует записанные блоки и транзакции.

Техническая реализация предпосылок

- Можно пересчитать реестр *с меньшим* хэшрейтом и перейти на него **в момент смены эпох**.
- Если на потерянный кошелек пришла только coinbase-транзакция, то необходимо также заменить только ее.
- Если на потерянный кошелек пришли транзакции с других адресов, то необходимо также заменить транзакции, связаные с этими адресами.

Технические предпосылки

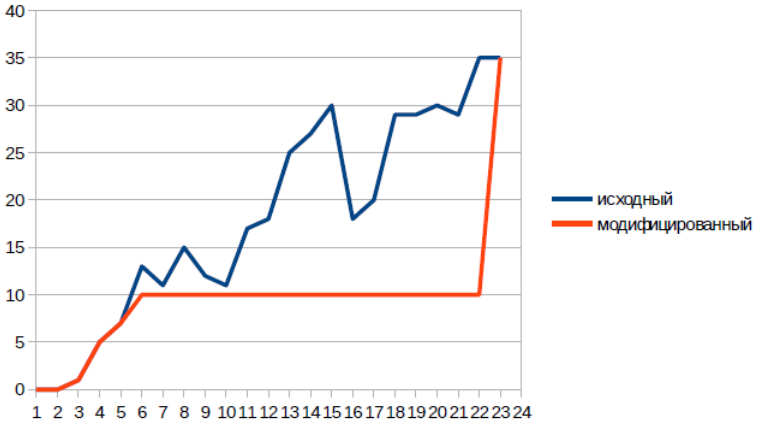
- Протокол консенсуса контролирует хэшрейт в течение только одной *эпохи* (пересчет хэшрейта осуществляется каждые 2016 блоков).
- **Никто** не контролирует записанные блоки и транзакции.

Техническая реализация предпосылок

- Можно пересчитать реестр *с меньшим* хэшрейтом и перейти на него **в момент смены эпох**.
- Если на потерянный кошелек пришла только coinbase-транзакция, то необходимо также заменить только ее.
- Если на потерянный кошелек пришли транзакции с других адресов, то необходимо также заменить транзакции, связаные с этими адресами.

Почему такой реестр может быть принят?

- Он **не противоречит** правилам системы
- Он **оставляет нетронутыми** или увеличивает балансы кошельков пользователей


40
35
30
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
исходный
модифицированный

Насколько это сложно?

Пусть

- 2^r операций в единицу времени – вычислительная мощность “майнера”,
- m блоков – до смены эпох,
- k блоков – всего в эпохе,
- t_s единиц времени – между выработкой блоков (по правилам системы),
- h – целевое значение хэш-рейта,
- N блоков – выработано к текущему моменту времени.

Тогда до смены эпох мы можем получить реестр с хэшрейтом

$$h' = r \cdot \log_2 \left(\frac{m}{N+m} \right).$$

Не будем себя ограничивать...

Если предположить возможность пересчета реестра в течение нескольких последующих (x) эпох, то последнее выражение при $k = m$ примет вид

$$h' = r \cdot \log_2 \left(\frac{xk}{N + xk} \right),$$

устремляя x в бесконечность, т.е. не ограничивая время перестройки реестра, имеем

$$h' = \lim_{x \to \infty} \left[r \cdot \log_2 \left(\frac{xk}{N + xk} \right) \right] = 0$$

Таким образом, требуемая вычислительная мощность для модификации реестра может быть сколь угодно мала.

А в попугаях?

Если пересчитать одну эпоху, как она будет выглядеть? Если хэшрейт в сети Bitcoin порядка $h = 71$, тогда при реализации вычисления хэш-функции

- **с использованием четырёхядерного процессора** (~ 400000 операций вычисления хэш-функции в секунду), $h' = 27$;
- **с использованием GPU** ($\sim 622 \cdot 16^6$ операций вычисления хэш-функции в секунду), $h' = 38$;
- **с использованием майнера EbitE10** ($\sim 18 \cdot 10^{12}$ операций вычисления хэш-функции в секунду), $h' = 53$.

В чем причины?

- **неполный контроль** за состоянием системы и, как следствие, существование трёх слабо связанных между собой логических реестров (учёта условных единиц, учёта транзакций и учётв блоков);
- **отсутствие** “**памяти**” о полном состоянии системы в реестре (фактически, контролируются только текущие балансы и связи между ними).

Выводы

- **Нет** проблем с модификацией реестра.
- Неизменность реестра гарантируется не столько криптографическими механизмами, сколько **доброй волей пользователей**.
- **Существует** возможность стимулировать пользователей перейти на новый реестр.
- Интерес представляет **построение протоколов систем распределенных реестров, позволяющих контролировать все критические** (с точки зрения безопасности системы) **параметры**.

Необходимо **корректно** использовать криптографические механизмы и не требовать от них слишком многого.

СПАСИБО ЗА ВНИМАНИЕ!

Вопросы?