Алгоритм криптографического консенсуса для корпоративных сетей "блокчейн".

Введение

[Абзацы 1, 2] – из исходного текста (pdf-файл)

Для устранения недостатков алгоритма PoW (Proof of Work) были созданы новые алгоритмы DPoS, LPoS, PoE, PoTT. [ссылка]

В этой статье авторы постарались сделать новый шаг в развитии алгоритмов консенсуса. Необходимость разработки нового алгоритма консенсуса была продиктована следующими требованиями к сети блокчейн

1. Время транзакции не более 1 мин.
2. Корпоративный тип сети "блокчейн".
3. Количество узлов, принимающих участие в выработке консенсуса может меняться от 10^3 до 10^4
4. Низкая энергоемкость реализации алгоритма

Ранее предложенные алгоритмы консенсуса признаны авторами неудовлетворяющими перечисленным выше требованиям по следующим причинам

[Вставка 1.] – краткое, на несколько строк перечисление причин.

Авторы предлагают другой подход к достижению консенсуса, реализованный в алгоритме криптографического консенсуса (АКК).

Основные положения и допущения АКК.

Мы исходим из того, что задача достижения консенсуса в первом приближении сводится к классической задаче "византийских генералов" [1]

Эта задача сводится к следующей схеме: n генералов, каждый из которых возглавляет одну армию, готовятся атаковать противника. Эти генералы стремятся достичь консенсуса, который состоит в том, что при наличии абсолютно надёжных каналов связи между всеми генералами и присутствии среди генералов m предателей все $n-m$ "лояльных" генералов должны после взаимного обмена информацией о численности своих армий иметь одинаковое представление о численности $n-m$ "лояльных" армий и о том, какими армиями командуют предатели. Известно, что задача имеет решение при следующем соотношении между n и m:

$$n > 3m+1. \qquad (1).$$

Решение этой задачи с помощью алгоритмов paxos BFT [ ] или practical BFT [ ] возможно при $n \sim 10^2$ [,], поскольку время достижения консенсуса есть $O(n^2)$ для этих алгоритмов.

Если вместо n генералов рассмотреть n узлов блокчейна, участвующих в выработке консенсуса, то очевидно, что эта задача подобна задаче византийских генералов. Для решения проблемы роста времени достижения консенсуса предлагается из конечного множества B_n узлов ($\overline{B_n} = n$) выделять подмножество $B_{n'}$ ($\overline{B_{n'}} = n'$) и исходя из равномерности распределения свойств узлов во всей сети блокчейн решать задачу не на n, а на $B_{n'}$ при условии, что $n >> n'$. Общее количество узлов во всей сети блокчейн положим равным N. Обозначим множество этих узлов A_N.

Пусть задана функция $s: X' \mapsto Y_{B_n}$

$$s = s(t), \quad t \in X' \qquad (2)$$

где t — независимая переменная, соответствующая текущему времени. Будем считать, что значения $s \in Y_{B_n}$ функции (2) соответствуют текущему состоянию B_n.

$\boxed{\text{Вставка 2}}$ — подробное описание ф-и $s(t)$.

Пусть задана функция $f: Y_{B_n} \mapsto \mathbb{N}_N$

$$f = f(s), \quad s \in Y_{B_n}, \qquad (3).$$

где $\mathbb{N}_N$ — конечное подмножество множества натуральных чисел $\mathbb{N}$ мощности N.

Пусть случайным образом из B_n выбрано подмножество $B_{n'}$, причем n' задается. Тогда получим

$$B_{n'} \subset B_n \subset A_N. \qquad (4)$$

Будем считать, что $Y_{B_{n'}} \subseteq Y_{B_n}$ множество значений всех s, соответствующих всем текущим состояниям всех узлов из $B_{n'}$.

Пусть функция f отображает $Y_{B_{n'}}$ на множество $\mathbb{N}_{n'}$, $\overline{\mathbb{N}_{n'}} = n'$. Будем считать, что номера узлов, полученных таким отображением есть j_k, $k = 1, \dots, n'$. Будем полагать, что $j_{\hat{k}}$ — номер назначенного мастер-узла, $1 \le \hat{k} \le n'$. Если b-формируемый блок (в — нек. число) — объяснить, в отношении которого в некоторый момент времени t' множество узлов $B_{n'}$ стремится достичь консенсуса, ~~где в — номер блока~~ то функцию хеширования SHA-3 [ ] над этим блоком обозначим $H(b)$, значение которой обозначим h. Тогда результат вычисления электронной подписи, например по алгоритму EdDSA [ ] с параметрами эллиптической кривой edwards25519 [ ] есть $sig(h)$.

Описание алгоритма

(1) (2) Выберем все j_k, включая $j_{\hat{k}}$ с помощью функции f. Выработка консенсуса осуществляется на полуинтервале времени $[t, t')$. Пусть в момент времени $t'' \in [t, t')$ узел с номером k $(1 \le k \le n)$ осуществляет запись I в блокчейн B_n. (3) В случае признания мастер-узлом с номером $j_{\hat{k}}$ допустимым включение записи I в блок b, узел с номером $j_{\hat{k}}$ передает всем узлам из $B_{n'}$ эту запись для проверки и включения в блок b. В противном случае запись I отвергается без уведомления. (4) Новая запись включается в блок до наступления момента t'. Мастер-узел рассылает сообщение тем же узлам о фиксации блока b. Все узлы из $B_{n'}$ вычисляют значение хеш-функции $H(b)$, равное, допустим, h. (5) Каждый узел из $B_{n'}$ вычисляет электронную подпись

$$s_k = sig(h), \quad k = 1, \dots, n', \; k \ne \hat{k} \qquad (5)$$

и передает ее на узел $j_{\hat{k}}$. В (5) номера электронных подписей изменены и могут не совпадать с j_k.

(6) Узел с номером $j_{\hat{k}}$ ожидает электронные подписи время Δt после наступления момента t'. В момент $t' + \Delta t$ на мастер-узле формируется ?вектор?

$$\vec{s}_b = (s_1, \dots, s_j), \quad 1 \le j \le n'. \qquad (5)$$

Мастер-узел проверяет каждую подпись из (5) и подсчитывает число корректных подписей. Подписи некоторых узлов из $B_{n'}$ могут оказаться некорректными в том случае, когда в $B_{n'}$ окажется узел с некоторым номером $j_{\tilde{k}}$, $1 \le \tilde{k} \le n'$, ~~выбранный ранее в качестве мастер-узла (очередного узла) для закрытия очередного блока b и мастер-узел может сформировать для $\tilde{b}$ вектор вида (5) так, что подписи одних и тех же узлов в векторах $\vec{S}_b$ и $\vec{S}_{\tilde{b}}$ окажутся разными, например~~ по следующим причинам:

как в исх. тексте

а) присвоение записи Γ узлом с номером $j_{\tilde{k}}$ некорректной

б) состояние s для узла с номером $j_{\tilde{k}}$ имеет разное значение для блоков b и ~~$\tilde{b}$ всвязи, например, с разными~~ текущими ~~моментами~~ времени t и $\tilde{t}$ ~~для этих блоков.~~

7. Узел $j_{\hat{k}}$ ~~получает~~ (вычисляет) количество корректных подписей μ и проверяет выполнение неравенства

$$\mu > \left[\frac{2}{3} n'\right]. \qquad (6)$$

Если (6) не выполняется, то узел с номером $j_{\hat{k}}$ делает вывод, что консенсус не достигнут; в противном случае ~~для~~ блока b составляется число

$$b\|s_{k_1}\|\cdots\|s_{k_\mu}, \quad 1 \le k_l \le n', \quad l = 1, \dots, \mu \qquad (7)$$

для которого вычисляется $H(b)$ и sig, являющаяся электронной подписью узла с номером $j_{\hat{k}}$.

8. Число

$$b\|s_1\|\cdots\|s_\mu\|sig(H(b\|s_1\|\dots\|s_\mu)) \qquad (8)$$

назовем новым закрытым блоком, точнее "номером" нового закрытого блока. Узел с номером $j_{\hat{k}}$ рассылает (8) всем узлам из множества A_N

9. На $\forall$ узле из A_N, допустим с номером $m \le N$, осуществляется проверка $\vec{S}_b$ и sig (7). Если проверка пройдена, то блок b добавляется в блокчейн узла

с номером m и блокчейн на узле m переходит в состояние $S' = S(t')$. Если этот узел не получил указанных выше подписей для проверки в промежутке времени $(t'+\Delta t, t'+\Delta t+\lambda]$, где λ — время задержки передачи информации, то узел с номером m сочтет консенсус недостигнутым и выберет новое множество B_{n^*}, применяя (3).

Построение функции f

Вычислим двойной хеш от (8), обозначим полученное число ν. Построим псевдослучайную битовую последовательность вида

$$H(\nu), H(\nu+1), \ldots \quad (10)$$

следующее число

$$R = H(\nu) \| H(\nu+1) \| \ldots \quad (11)$$

Битовую запись (11) разделим на последовательно без пропусков и перекрытий расположенные r-мерные векторы ~~...~~ из которых строим множество номеров узлов из $B_{n'}$, $k = 1, \ldots, n'$; $j_k \leq N$. Если случайно повторится уже полученный номер j_k, то повторно полученное число пропускается (игнорируется).

Литература