Моделирование и анализ информационных систем. Т. 26, №2 (2019), с. 244–255
Modeling and Analysis of Information Systems. Vol. 26, No 2 (2019), pp. 244–255

Theory of computing

DOI: 10.18255/1818-1015-2019-2-244-255

УДК 51-37

К вопросу использования «полезных» задач для обеспечения работой блокчейн систем

Мурин Д. М., Князев В. Н.

Поступила в редакцию 13 марта 2019
После доработки 17 мая 2019
Принята к публикации 20 мая 2019

Аннотация. Статья является продолжением работы о возможных подходах к решению задачи «Useful Proof-of-work for blockchains». Мы предлагаем некоторые альтернативные направления поиска полезных задач для обеспечения работой, основанные на том, что процесс решения хеш-головоломки близок к многократному независимому повторению следующего эксперимента: пусть задано достаточно большое по мощности множество (например, состоящее из 2^n элементов, для достаточно большого n), только незначительная часть элементов которого обладает определенным свойством. Эксперимент состоит в равномерном выборе элемента из этого множества с последующей проверкой наличия у него указанного свойства. Таким образом, процесс решения хеш-головоломки может быть заменен, например, поиском редких астрономических объектов или поиском позиций игры Го, удовлетворяющих определенным условиям. Кроме того, мы описываем возможную атаку на блокчейн-систему, в которой алгоритм генерации индивидуальных представителей задач для обеспечения работой заменен алгоритмом выбора индивидуальных представителей из имеющейся базы данных, со стороны недобросовестных поставщиков индивидуальных представителей задач, в случае их публичного сбора, и обсуждаем некоторые способы защиты от этой атаки.

Ключевые слова: доказательство работой, блокчейн, алгоритм

Для цитирования: Мурин Д. М., Князев В. Н., "К вопросу использования «полезных» задач для обеспечения работой блокчейн систем", *Моделирование и анализ информационных систем*, **26**:2 (2019), 244–255.

Об авторах:
Мурин Дмитрий Михайлович, orcid.org/0000-0002-8068-0784, канд. физ.-мат. наук, доцент кафедры КБиММОИ,
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: nirum87@mail.ru

Князев Владимир Николаевич, orcid.org/0000-0003-4967-0825, ассистент кафедры КБиММОИ
Ярославский государственный университет им. П.Г. Демидова,
ул. Советская, 14, г. Ярославль, 150003 Россия, e-mail: darknyaz@yandex.ru

1. Модель классического блокчейна

Напомним некоторые определения.

Определение 1. *Система* ***Proof-of-Work (PoW)*** *— это 3 алгоритма:*

1. `Gen`(I) — *вероятностный полиномиальный по времени алгоритм, получающий на вход двоичное слово длины* $n - I$ *и выводящий индивидуального представителя* TI *некоторой задачи.*

2. `Solve`(TI) — *алгоритм, решающий индивидуального представителя* TI *и выводящий (в случае завершения работы) его решение* S*.*

3. `Verify`(TI, S) — *вероятностный полиномиальный по времени алгоритм, проверяющий решение* S *индивидуального представителя задачи* TI*.*

Алгоритм `Solve` *должен иметь известную сложность в среднем* $t(n)$*, причем любые алгоритмы со сложностью в среднем, асимптотически меньшей, чем* $t(n)$*, не должны выдавать истинные решения* TI*, а приближенные решения, выдаваемые такими алгоритмами, должны с большой вероятностью отвергаться алгоритм* `Verify`*.*

Системы PoW, предназначенные для использования в блокчейн технологиях, имеют несколько отличающееся определение. Мы будем называть такие системы **Proof-of-Work-for-Blockchain** (**PoWB**).

Определение 2. *Система PoW называется системой* ***PoWB****, если*

1. *Алгоритм* `Gen` *получает на вход двоичное слово длины* $n - I$ *и натуральное число* $- C$ *и возвращает индивидуального представителя задачи* TI*, как и в системе PoW.*

2. *Сложность в среднем алгоритма* `Solve` *зависит от входа* C*. Изменяя* C *можно управлять сложностью алгоритма* `Solve`*.*

Определение 3. *Система PoW называется системой* ***Useful Proof-of-Work (UPoW)****, если алгоритм* `Gen` *генерирует индивидуальных представителей полезных за пределами блокчейна задач.*

Аналогичным образом вводится определение для систем **Useful Proof-of-Work-for-Blockchain** (**UPoWB**).

Определение 4. *Блокчейн, в котором система PoW является системой UPoWB, мы будем называть* ***UPoWB-блокчейном****.*

Определение 5. ***Блокчейн*** *(от англ. blockchain, буквально — «цепочка блоков») — растущий список записей, называемых* ***блоками****, которые криптографически связаны друг с другом.*

Под классическим блокчейном мы, в первую очередь, понимаем блокчейн системы Bitcoin, описанный в работе [6], упрощенная схема которого представлена на рис. 1. Классический блокчейн состоит из следующих компонентов:

1. **Система PoWB**. Система PoWB обеспечивает формирование блоков с определенной периодичностью.

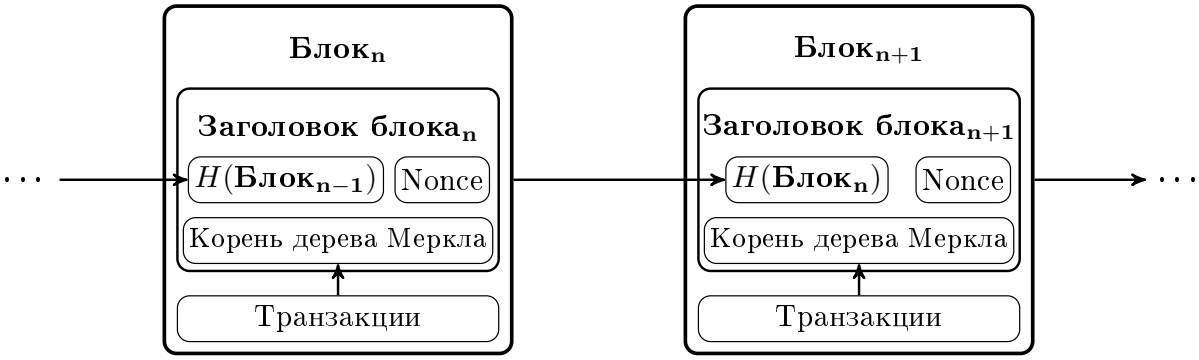

Рис. 1: Классический блокчейн
Fig. 1: Classical blockchain

2. **Транзакция**. В системе Bitcoin транзакции хранят информацию о переводах денежных средств. В общем случае это некоторые данные, чья целостность и аутентичность обеспечиваются с помощью электронной подписи. Транзакции делятся на два вида: *неподтвержденные* и *подтвержденные*. При создании транзакция считается неподтвержденной. Для изменения вида транзакция должна быть связана с блоком, а блок, в свою очередь, должен войти в «победившую» цепочку блоков.

3. **Блок**. Состоит из заголовка блока и списка, вошедших в блок транзакций. Служит для обеспеченного вычислительной работой подтверждения транзакций путем их криптографического связывания с блоком с помощью алгоритма `Solve` системы PoWB. Каждый блок криптографически связан с предыдущим блоком, таким образом, блоки образуют связанный список — *блокчейн.*

4. **Блокчейн-сеть**. Представляет собой распределенную вычислительную систему, вычислительные узлы которой практически все время решают индивидуальных представителей TI задач системы PoWB для подтверждения транзакций и периодически обмениваются между собой информацией в целях синхронизации и актуализации данных о сгенерированных блоках.

Отметим, что блокчейн-сеть обычно разбивается на вычислительные **кластеры** (**пулы**), в каждом из которых решается один и тот же индивидуальный представитель задачи. Пусть блокчейн-сеть обладает зависящей от времени вычислительной мощностью $M(t)$ и в момент времени t существует $K(t)$ кластеров, каждый их которых обладает идентификатором — числом из множества $\{1, \ldots, K(t)\}$, и i-й кластер обладает вычислительной мощностью $M_i(t)$. Вычислительная мощность блокчейн-сети $M(t) = \sum_{i=1}^{K(t)} M_i(t)$, но вычислительная мощность, направленная на решение одного индивидуального представителя TI, не превосходит $\max_{1 \le i \le K(t)} M_i(t)$.

Также в каждом кластере (а возможно и на каждом вычислительном узле) хранится копия цепочки блоков, поэтому по сути она является распределенной базой данных. В каждом вычислительном кластере в момент времени t может формироваться собственный блок для хранимой цепочки блоков, но при получении в процессе обмена с другими кластерами информации о более длинной цепочке блоков, кластер обязан переключиться на работу с ней, отказавшись от результатов своей

предыдущей работы. Одновременно может существовать несколько различных цепочек блоков, но длина каждого альтернативного ответвления от основной цепочки не может превышать некоторой величины $Q(t)$. Времени, за которое формируются $Q(t)$ блоков, должно быть достаточно для оповещения всех кластеров о наличии более длинной цепочки (для блокчейн-системы Bitcoin на момент написания статьи $Q(t) = 6$). Таким образом, одного блока для подтверждения транзакции работой недостаточно. Транзакция считается подтвержденной в случае, если она включена в блок, за которым была сформирована цепочка из $Q(t)$ блоков. Это сделано для уменьшения вероятности **атаки двойного расходования** [6].

2. О некоторых альтернативах хеш-головоломкам

В качестве задач для обеспечения работой в классическом блокчейне используются задачи поиска двоичного слова, хеш-значение от которого обладает определенными свойствами хеш-головоломки. В качестве примера мы рассмотрим вариант хеш-головоломок, для которого хеш-значение должно иметь начало из не менее l_1 нулей (в более «эластичном» варианте хеш-значение должно быть ассоциировано с натуральным числом меньшим, чем l_2). l_1 и l_2 являются управляющими параметрами блокчейн-системы, отвечающими за сложность решаемых индивидуальных представителей хеш-головоломок.

Использование хеш-головоломок в блокчейн-системах позволяет решить две главные задачи: криптографически привязать транзакции к блоку и обеспечить транзакции вычислительной работой. Обратим внимание, что эти задачи могут решаться разными способами (не обязательно искать методы решения обеих задач сразу, для каждой задачи может быть предложен свой вариант решения). Основным недостатком хеш-головоломок является их бесполезность за пределами блокчейн-систем.

Рассмотрим более подробно процесс решения хеш-головоломки.

Пусть $H : \{0,1\}^n \to \{0,1\}^m$ является качественной криптографической хеш-функцией. Выходное распределение, порождаемое функцией H на равномерном входе — $H(U_n)$, где U_n — случайная величина, равномерно распределенная на $\{0,1\}^n$, должно быть вычислительно неразличимо с равномерным распределением на множестве $\{0,1\}^m$. При решении хеш-головоломки (за неимением хороших алгоритмов решения, существенно отличающихся от перебора) необходимо многократно выбирать аргумент хеш-функции x и проверять, удовлетворяет ли хеш-значение $y = H(x)$ условию головоломки. Этот процесс близок к многократному повторению эксперимента с равномерным выбором $y \leftarrow U_m$ и последующей проверкой того, удовлетворяет ли этот элемент условию головоломки. При этом множество $\{0,1\}^m$ делится на 2 подмножества. Элементы первого подмножества удовлетворяют условиям головоломки, а элементы второго — нет (например, все двоичные слова длины m делятся на те, которые имеют начало из l_1 нулей, и те, которые не имеют).

Следовательно, у каждого эксперимента есть два возможных исхода и случайная величина X, которая принимает значение 1 с вероятностью p в случае успешного завершения эксперимента и 0 — в противном случае (с вероятностью $q = 1-p$), имеет распределение Бернулли. Определим случайную величину Y, которая принимает значение, равное номеру первого «успеха». Эта случайная величина имеет геомет-

рическое распределение: $Pr[Y = n] = q^{n-1}p$ и математическое ожидание $E[Y] = \frac{1}{p}$. Таким образом, в среднем нам необходимо провести $\frac{1}{p}$ экспериментов до получения первого успеха. Тогда, если мы случайно и равномерно выбираем слово длины n, то вероятность выбрать слово, имеющее начало из l_1 нулей, составляет $p = \frac{1}{2^{l_1}}$, и в среднем первое такое слово мы выберем через 2^{l_1} экспериментов.

По данным сайта www.blockchain.com, для блокчейн системы Bitcoin число проводимых экспериментов при решении хеш-головоломок достигало в 2017 году $3,6 * 10^{22} \simeq 2^{76}$ [2]. При этом порядка 2^{73} экспериментов могли быть проведены на одном кластере [3].

Предположим, что мы исследуем некоторые объекты или физические процессы, причем интерес для исследования представляют чрезвычайно редкие события (например, встречающиеся с вероятностью $p = \frac{1}{2^{l_1}}$). Тогда любое вновь обнаруженное событие могло бы выступать в качестве обеспечения работой для некоторой блокчейн системы. Например, для блокчейн системы Bitcoin такими объектами могли бы служить астрономические объекты, обладающие особыми, редкими свойствами (число звезд в наблюдаемых галактиках по оценкам составляет $\sim 10^{23}$, и обнаружение объектов, обладающих редкими свойствами, например, при анализе снимков, сделанных космическими телескопами, могло бы обеспечивать транзакции вычислительной работой), объекты возникающие при изучении элементарных частиц (редкие события, выявляемые при анализе треков частиц, получаемых с помощью адронного коллайдера), удовлетворяющие особым условиям шахматные партии (оценка их числа $\sim 10^{120}$ [4]) или позиции (оценка их числа $\sim 10^{43}$ [4]), позиции игры Го (оценка их числа для игрового поля размером 19 на 19 клеток $\sim 10^{171}$ [5]) и другие.

При этом алгоритм `Gen` должен указывать на некоторый массив данных, в котором должен осуществляться поиск редкого события или явления (например, на конкретный снимок или подмножество снимков, сделанных космическими телескопами, или множество шахматных партий, начинающегося с определенной последовательности ходов, и так далее). Алгоритм `Solve` производит перебор объектов из массива данных. Детерминированный алгоритм `Verify` проверяет, что обнаруженный объект действительно удовлетворяет необходимым требованиям.

Повторное использование одного и того же объекта можно исключить, осуществляя поиск по ранее обнаруженным объектам, поскольку подтверждение работой потребует публикации данных об обнаруженном объекте.

3. Некоторые подходы к построению UPoWB-блокчейна

В этом разделе мы опишем возможную атаку на блокчейн систему, предложенную в [1], со стороны недобросовестных поставщиков индивидуальных представителей задач в случае их публичного сбора, и обсудим некоторые способы защиты от этой атаки.

3.1. Разработанная модель UPoWB-блокчейна

Наша идея построения системы UPoWB основана на отказе от алгоритма «генерации» `Gen` индивидуального представителя задачи системы UPoWB. Вместо этого мы предполагаем, что имеется база данных индивидуальных представителей полезных задач (например, полученных в ходе практической деятельности). В базе данных в процессе предобработки необходимо выбрать таких индивидуальных представителей, решение которых можно принять в качестве обеспечения работой. То есть индивидуальных представителей, решение которых должно потребовать существенных вычислительных ресурсов. Мы считаем, что некоторый алгоритм `Solve` решения индивидуальных представителей является общеизвестным (например, для задач из класса NP в качестве такого алгоритма может выступать SAT-решатель), поэтому все узлы блокчейн-сети имеют возможность решать индивидуальных представителей. Также мы считаем, что решение индивидуального представителя задачи можно относительно быстро проверить с помощью алгоритма `Verify` (что, естественно, выполнено для задач из класса NP).

Компоненты разработанного блокчейна:

1–4. Совпадают с компонентами 1–4 классического блокчейна (см. раздел 1.), за исключением того, что в блок или дерево Меркла могут включаться требующие решения индивидуальные представители полезных задач.

5. **База данных индивидуальных представителей полезных задач**. В базе данных содержатся индивидуальные представители полезных задач требующие решения. Базу данных можно считать открытым ресурсом, находящимся под управлением третьей доверенной стороны — **администратора базы данных**. Администратора базы данных можно рассматривать как публичный сервис. Перед открытой публикацией индивидуальных представителей проводится их предварительный анализ (предобработка), от которой требуется исключение из рассмотрения легких индивидуальных представителей и, по возможности, выбор представителей, которые могут быть решены за определенный интервал времени системой блокчейн.

3.2. Атака на систему со стороны недобросовестных поставщиков задач

В работе [1] предложены некоторые подходы к использованию NP-полных задач в качестве задач для обеспечения работой. В одном из подходов предлагается формировать базу данных индивидуальных представителей задач для обеспечения работой из «внешних» источников. При этом в указанной работе неявно предполагалась либо добросовестность поставщиков (источников) задач, либо достаточная сила алгоритма управления сложностью задач, позволяющая исключать из рассмотрения задачи недобросовестных поставщиков. Если ни одно из этих условий не обеспечивается, то следует учитывать описанную далее атаку. Будем считать, что индивидуальные представители независимо и равномерно выбираются из множества доступных для решения индивидуальных представителей, находящихся в

базе данных. Выбор может быть основан на информации из текущего блока. Будем считать, что он зависит от выбранных транзакций.

Предположим, что в качестве источника задач может выступать злоумышленник. Тогда для обеспечения работой злоумышленник может выбирать индивидуальных представителей задач, которые просты для решения, маскировать их под индивидуальных представителей общего положения и предлагать их для включения в базу данных. То есть злоумышленник может пытаться предлагать для включения в базу данных индивидуальных представителей с известными ему секретами. Например, злоумышленник может выбрать индивидуального представителя задачи о рюкзаке, имеющего сверхрастущий вектор или вектор с малой плотностью, замаскировать его с помощью сильного модульного умножения (или другого преобразования) и предложить для включения в базу данных.

При этом злоумышленник, с нашей точки зрения, может преследовать две цели:

1. *Тривиальная.* Заключается в получении выгоды от формирования блока. При выборе для обеспечения работой индивидуального представителя, для которого злоумышленнику известен секрет, он (злоумышленник) получает преимущество в скорости формирования блока.

2. *Основная.* Получение выгоды от быстрого формирования альтернативной цепочки «тяжелых» блоков. Если доля индивидуальных представителей с секретами, известными злоумышленнику, в базе данных достаточно велика, то это обеспечивает высокую вероятность выбора выгодных злоумышленнику индивидуальных представителей для обеспечения работой, и злоумышленник получает возможность быстрого формирования достаточно длинной альтернативной цепочки «тяжелых» блоков (понятие «тяжелый» блок введено в работе [1]).

Рассмотрим случайную величину X', которая принимает значение 1 с вероятностью p', если выбран индивидуальный представитель задачи с секретом, известным злоумышленнику, и 0 — в противном случае (с вероятностью $q' = 1 - p'$). Вновь случайная величина X' имеет распределение Бернулли. Соответственно, случайная величина Y', которая принимает значение равное номеру первого «успеха» (то есть выбору индивидуального представителя с секретом, известным злоумышленнику), имеет геометрическое распределение. Математическое ожидание случайной величины $E[Y'] = \frac{1}{p'}$. Если $E[Y'] << E[Y]$, то злоумышленник следующим образом может быстро сформировать цепочку «тяжелых» блоков.

Злоумышленник случайным образом выбирает неподтвержденные транзакции для привязки к блоку, если выбор оказывается неудачным (то есть для полученной выборки транзакций ассоциированный индивидуальный представитель не обладает секретом, известным злоумышленнику), то злоумышленник выбирает новое множество транзакций. Поскольку среднее число экспериментов, необходимых для выбора индивидуального представителя с секретом $E[Y']$, значительно меньше, чем среднее число экспериментов, необходимых для решения хеш-головоломки $E[Y]$, то злоумышленник получает существенное преимущество при формировании цепочки «тяжелых» блоков.

Пример 1. *Пусть мы используем для формирования «легких» блоков хеш-головоломки с условием, что хеш-значение должно иметь начало из l_1 нулей. Предположим, что злоумышленник может обеспечить ситуацию, при которой из всего множества доступных для решения блокчейн-сети индивидуальных представи-*

телей $\frac{1}{poly(l_1)}$*-доля имеет известные ему секреты (*$poly(l_1)$ *— некоторый полином от* l_1*). Тогда «честному» пользователю для формирования одного «легкого» блока в среднем требуется* 2^{l_1} *экспериментов, трех «легких» блоков, эквивалентных одному «тяжелому» блоку (в соответствии с [1]),* $-\ 3 \cdot 2^{l_1}$*, а злоумышленнику для формирования одного «тяжелого» блока требуется* $poly(l_1) + 2^{l_1}$ *экспериментов.*

Для достижения обеих целей (особенно второй цели) злоумышленнику необходимо добиться, чтобы доля принятых в базу данных индивидуальных представителей с известными ему секретами была очень велика. Если мы не можем различать индивидуальных представителей с секретами (или, возможно, более широкие множества индивидуальных представителей, подмножествами которых являются индивидуальные представители с секретами) от задач общего положения на этапе управления сложностью задач (в противном случае, при обнаружении они подлежат исключению из рассмотрения), то необходимо предложить иные способы противодействия рассмотренной атаке.

3.3. Некоторые способы защиты от описанной атаки

Описанная в предыдущем разделе атака имеет мотив в виде увеличенного вознаграждения за «тяжелый» блок и возможна благодаря тому, что злоумышленник, зная решение индивидуального представителя с известным ему секретом, может *единолично* сформировать «тяжелый» блок (как описано в [1]).

Одним из возможных способов защиты от описанной атаки может являться маскирование полученной от поставщика задачи администратором базы данных индивидуальных представителей полезных задач. Например, полученный от поставщика индивидуальный представитель полезной задачи может быть полиномиально сведен в задачу выполнимости булевой формулы, в полученной формуле осуществлена случайная замена и случайная перестановка переменных, возможно, с раскрытием скобок или вынесением отдельных элементов за скобки. Дополнительно можно свести полученного индивидуального представителя задачи выполнимости в иную (случайно выбранную) NP-полную задачу с последующим сведением в выполнимость. Описанный способ может существенно затруднить злоумышленнику поиск «своих» индивидуальных представителей. Однако он обладает некоторыми недостатками. Во-первых, мы не можем гарантировать, что не существует индивидуальных представителей, сохраняющих определенную структуру после указанных преобразований, позволяющую эффективно установить связь с исходным индивидуальным представителем. Во-вторых, указанные преобразования могут приводит к увеличению размерности индивидуального представителя.

Делая бессмысленной основную цель злоумышленника, можно потребовать, чтобы «тяжелые» блоки формировались редко, например, не чаще, чем один на $Q(t)+1$ блоков, где $Q(t)$ такое же, как в разделе 1. В этом случае не только злоумышленник, но и все кластеры (узлы) блокчейн-сети лишаются возможности строить цепочки «тяжелых» блоков. При этом злоумышленник сохраняет возможность получить выгоду от формирования одного «тяжелого» блока. И введение платы за решение задачи (от поставщика индивидуального представителя — решившему задачу) не исправляет ситуации, поскольку злоумышленник в этом случае заплатит сам себе.

Мы видим основным способом защиты от описанной в предыдущем разделе атаки изменение схемы мотивации создания «тяжелого» блока. В [1] предлагалось оставить в блокчейн системе как «легкие», так и «тяжелые» блоки. При этом «легкие» блоки использовались в качестве своеобразного метронома. В измененной схеме мы потребуем, чтобы каждый блок нес полезную нагрузку, то есть был «тяжелым». Соответственно за формирование каждого блока устанавливается фиксированная плата. Кроме этого, мы разъединим процесс решения индивидуального представителя полезной задачи и процесс формирования блока. А именно, мы потребуем, чтобы решение индивидуального представителя из базы данных публиковалось в базе данных вместе с подписью администратора. С точки зрения компонентов блокчейн потребуются небольшие изменения: в блоке должно появиться поле для подписи администратора базы данных, а в самой базе данных должны будут публиковаться не только индивидуальные представители, но и их решения, подписанные электронной подписью администратора.

Обозначим через `Info`(**Block**) информацию, которой будет достаточно для формирования блока (например, `Info`(**Block**) может состоять из указателя на предыдущий блок, корня дерева Меркла и списка транзакций). Отметим, что мы также используем `Info`(**Block**) для выбора индивидуального представителя из базы данных (например, можно использовать хеш-значение от `Info`(**Block**) или его части в качестве указателя на индивидуального представителя в базе данных). Приведем более формальный алгоритм формирования «тяжелого» блока в модифицированном блокчейне.

Решение индивидуального представителя:

1. Выбрать с помощью `Info`(**Block**) индивидуального представителя из базы данных (например, способом, описанным в [1]).

2. Решить индивидуального представителя полезной задачи и направить решение вместе с информацией о решенном индивидуальном представителе и формируемом блоке `Info`(**Block**) в базу данных.

3. После публикации решения в базе данных индивидуальный представитель считается решенным.

Администратор базы данных, получая предлагаемые решения, может подписывать их и публиковать, оставляя проверку правильности решений кластерам блокчейн-сети. Также администратора можно наделить функцией проверки правильности решения. В этом случае администратор проверяет правильность выбора индивидуального представителя для формируемого блока и правильность самого решения. Если решение верно, то подписывает его и публикует.

Формирование блока:

1. Выбрать опубликованное в базе данных решение.

2. Путем решения хеш-головоломки сформировать блок, в котором содержится информация о решенном индивидуальном представителе, его решение и подпись администратора базы данных.

Таким образом, любой кластер может закончить формирование «тяжелого» блока, решив стандартную хеш-головоломку для опубликованного в базе данных блока, связанного с решенным индивидуальным представителем. Так как подпись администратора трудно спрогнозировать заранее, злоумышленник не будет обладать преимуществом в формировании «тяжелого» блока. Рис. 2 иллюстрирует наше решение.

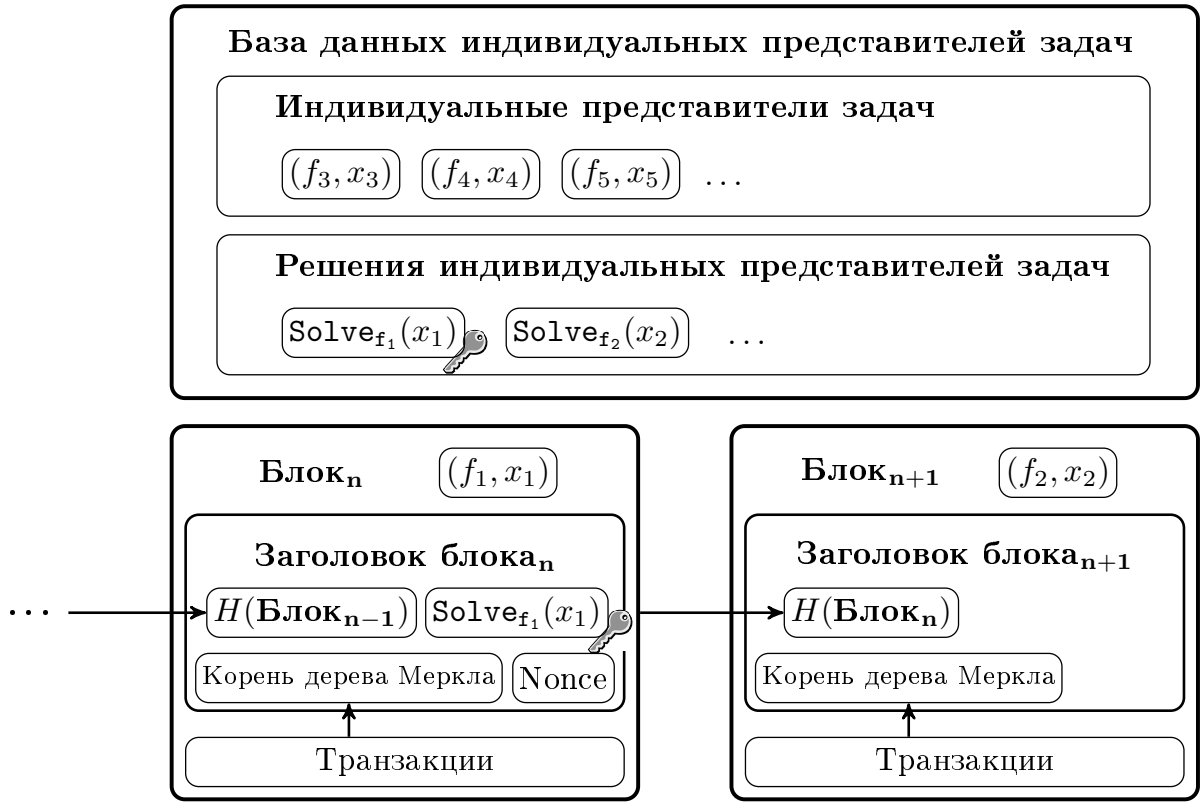

Рис. 2: Модель разработанного UPoWB-блокчейна. Рисунок иллюстрирует невозможность формирования $n + 1$ блока, так как ассоциированная с ним задача еще не прошла проверку администратором базы данных индивидуальных представителей полезных задач

Fig. 2: UPoWB-blockchain model. The picture illustrates the impossibility of forming $n + 1$ block since the associated task did not pass the check of individual representatives of useful tasks database administrator yet

В данном случае можно лишить смысла достижение тривиальной цели злоумышленника путем установления обязательной оплаты решения индивидуального представителя его поставщиком. Чтобы избежать выплаты вознаграждения злоумышленником себе самому, часть вознаграждения можно адресовать администратору базы данных. Таким образом, после опубликования решения кластер, предоставивший верное решение, получает за это награду, определенную поставщиком для соответствующего индивидуального представителя.

Отметим, что в модифицированном блокчейне в качестве полезной нагрузки могут выступать задачи любой сложности. Однако в одном интервале времени необхо-

димо обеспечить доступность для решения индивидуальных представителей, имеющих приблизительно одинаковую сложность.

4. Заключение. О «мгновенной» сложности задач

В заключение мы хотим отметить, что технология блокчейн позволяет ставить вопрос о «мгновенной» сложности решаемых задач. Фактически, для обеспечения работой нам необходим индивидуальный представитель, решение которого мы готовы принять в качестве такого обеспечения в текущем временном интервале. Вне этого интервала могут поменяться как внутренние факторы, например, мощность блокчейн-сети, так и внешние факторы, например, будет разработан новый существенно более быстрый алгоритм для решения индивидуальных представителей определенного вида. С точки зрения классической теории сложности каждый индивидуальный представитель решается за время $O(1)$. С точки зрения «мгновенной» сложности нам необходим более точный анализ числа шагов алгоритмов, решающих индивидуальных представителей задач.

Список литературы / References

[1] Дурнев В. Г., Мурин Д. М., Соколов В. А., Чалый Д. Ю., “О некоторых подходах к решению задачи «Useful Proof-of-work for blockchains»”, *Моделирование и анализ информационных систем*, **25**:4 (2018), 402–410; [Durnev V. G., Murin D. M., Sokolov V. A., Chalyy D. Ju., “On Some Approaches to the Solution of the Problem ”Useful Proof-of-work for Blockchains””, *Modeling and Analysis of Information Systems*, **25**:4 (2018), 402–410, (in Russian).]

[2] “Hash Rate. The estimated number of tera hashes per second (trillions of hashes per second) the Bitcoin network is performing.”, 2018, https://www.blockchain.com/ru/charts/hash-rate.

[3] “Распределение количества перебора хешей.”, 2018, https://www.blockchain.com/ru/pools; [“Hashrate quantity distribution.”, 2018, https://www.blockchain.com/ru/pools, (in Russian).]

[4] Shannon C., “Programming a Computer for Playing Chess”, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **41**:314 (1950), 256–275.

[5] “Number of legal Go positions.”, 2016, https://tromp.github.io/go/legal.html.

[6] Nakamoto S., “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008, https://bitcoin.org/bitcoin.pdf.

[7] Marshall Ball, Alon Rosen, Manuel Sabin, Prashant Nalini Vasudevan, “Proofs of Useful Work”, 2017, https://eprint.iacr.org/2017/203.pdf.

[8] “Problem 11. Useful Proof-of-work for blockchains”, 2017, https://nsucrypto.nsu.ru/archive/2017/round/2/section/0/task/11/.

Murin D. M., Knyazev V. N., "On the Issue of Using “Useful” Tasks for Proof of Works in Blockchain", *Modeling and Analysis of Information Systems*, **26**:2 (2019), 244–255.

DOI: 10.18255/1818-1015-2019-2-244-255

Abstract. This paper is a logical continuation of the paper about possible approaches to solving the "Useful Proof-of-work for blockchains" problem. We suggest some alternative ways for searching useful tasks for Proof-of-work systems. These ways are based on the process of the multiple and independent repetition of a simple experiment. The experiment is to chose an element independently and uniformly from a quite large set and then to check if the chosen element has a specific rare property. In the classic blockchain of Bitcoin this experiment is a so-called hash-puzzle. In these terms the process of solving a hash-puzzle may be replaced by searching rare astronomical objects or Go positions with specific conditions. Moreover, we describe a possible attack on the blockchain systems in which the task instance generation algorithm is replaced by the algorithm of selecting the task instance from the existing database with public access for publication of task instances and discuss the way of protection.

Keywords: proof-of-work, blockchain, algorithm

On the authors:
Dmitry M. Murin, orcid.org/0000-0002-8068-0784, PhD, Docent,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: nirum87@mail.ru

Vladimir N. Knyazev, orcid.org/0000-0003-4967-0825, Assistant,
P.G. Demidov Yaroslavl State University,
14 Sovetskaya str., Yaroslavl 150003, Russia, e-mail: darknyaz@yandex.ru