Об одном классе дискретных функций для синтеза протоколов согласования Proof-of-Space в блокчейнах

О. А. Логачев, С. Н. Федоров

***Аннотация*—Классический подход к построению блокчейнов, в которых процедура майнинга основана на поиске пользователем решения некоторой «умеренно трудной» задачи (Proof-of-Work), как известно, обладает рядом недостатков: в первую очередь критике подвергается необходимость проводить большое количество вычислений, что влечет соответствующие энергозатраты.**

В связи с этим одним из направлений исследований блокчейнов является разработка других подходов к процедуре добавления нового блока в цепочку. В частности, в 2015 году Дзембовский и др. сформулировали идею концепции Proof-of-Space (PoS), суть которой сводится к тому, что вместо необходимости в среднем потратить значительное время на вычисления пользователи для добавления нового блока должны занять определенный объем памяти на своих компьютерах. Последнее условие реализуется, например, требованием обратить некоторую дискретную функция, для чего пользователю предлагается сохранить в памяти таблицу значений этой функции. В 2017 году Абусалах и др., решая проблемы исходного варианта (простого) PoS-блокчейна, предложили использовать дискретную функцию, являющуюся особой композицией двух других функций.

В настоящей работе проводится анализ протокола согласования Абусалаха и др. Показано, что данное предложение не удовлетворяет требованию PoS-блокчейнов о резервировании заданного объема памяти. Кроме того, обсуждаются вопросы анализа математических моделей блокчейнов как криптографических примитивов.

***Ключевые слова*—блокчейн, метод Хеллмана, протокол консенсуса, протокол согласования, Proof-of-Space**

I. Введение

A. Понятие блокчейна

Блокчейн, рассматриваемый как криптографический примитив (см. [1]), представляет собой базу данных специального вида, состоящую из линейно упорядоченных записей. К этой базе данных можно обращаться с запросами только двух типов:

- запрос на чтение любой записи (доступен всем);
- запрос на добавление записи в конец базы данных (право пользователей на это действие определяется способом реализации блокчейна и конкретным приложением).

База данных такого вида называется *блокчейном*, если удовлетворяет требованиям *незыблемости* (persistency) и *живучести* (liveness). Первое требование предполагает невозможность изменения записей и их порядка в базе, а второе — неизбежность добавления в некоторый момент времени (в конец базы) каждой записи, удовлетворяющей всем условиям данной реализации системы.

Литература по блокчейнам для широкого круга читателей (без использования математического аппарата; см., например, [2], [3]) представляет собой, по сути, рекламные материалы с футуристическими сюжетами экономических и финансовых преобразований, а также с акцентированием внимания читателя на подробностях инфраструктур криптовалют. Сугубо математические публикации по этой теме не столь многочисленны и не отвечают в полной мере на первостепенные вопросы соответствующей теории: даже существование блокчейнов доказано лишь при предположениях, которые трудно назвать естественными (см., например, [4]).

В понятие блокчейна заложены основополагающие элементы — майнинг и протокол консенсуса (согласования) в сети. Любой пользователь (участник сетевого взаимодействия) может добавить новый блок в цепочку (запись в конец базы), если решит некоторую «умеренно трудную» задачу. Попытки решения этой задачи и называются майнингом. При этом «трудность» может иметь принципиально разные трактовки. Чаще всего речь идет о средней трудоемкости (количестве элементарных операций) специальным образом подобранной задачи обращения криптографической хэш-функции. В литературе соответствующая конструкция именуется PoW-блокчейном (от слов Proof-of-Work).

Имеющиеся у PoW-блокчейнов недостатки мотивировали ряд исследователей на разработку иных концепций достижения консенсуса. В [5] предлагается подход, основанный на использовании относительно больших объемов памяти (вместо предположения о необходимости затратить существенное время на решение «умеренно трудной» задачи). Соответствующую разновидность блокчейнов называют PoS- или PoSpace-блокчейнами — от словосочетания Proof-of-Space (существуют и синонимичные понятия Proof-of-Capacity и Proof-of-Storage). В статье [6] обосновывается неприемлемость прямого использования описанного в [5] (простого) PoS-блокчейна и предложена новая конструкция, по мнению авторов, лишенная недостатков первоначальной. В разделе III настоящей работы приводится описание этой конструкции, а в разделе IV — ее анализ с целью выяснить, действительно ли новая конструкция решает все проблемы и имеет преимущества над PoW-блокчейнами.

Статья получена 17 ноября 2020.

Олег Алексеевич Логачев, кафедра информационной безопасности факультета ВМК МГУ имени М. В. Ломоносова, (email: ollog@inbox.ru).

Сергей Николаевич Федоров, Институт проблем информационной безопасности МГУ имени М. В. Ломоносова, (email: s.n.feodorov@yandex.ru).

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта 18-29-03124 мк.

B. Обозначения

При произвольном натуральном числе n множество $\{0,1\}^n$ состоит их всех двоичных (битовых) строк (наборов) длины n. Черта над выражением, принимающим значения в любом таком множестве, означает побитовое логическое отрицание.

Для любой функции φ стандартным образом через $\mathcal{O}^\varphi$ обозначается оракул, который в ответ на запрос x из области определения φ возвращает значение $\varphi(x)$. Как принято считать, время работы оракула не влияет на общую трудоемкость: мы полагаем, что ответ оракула на любой запрос — это результат выполнения одной элементарной операции.

В дальнейшем изложении во многом сохранены обозначения работы [6]. В частности, для натурального числа N с помощью $[N]$ записывается произвольное множество двоичных строк (наборов) мощности N, элементы которого можно выбирать с помощью эффективного вероятностного алгоритма. Для простоты (без ограничения общности) будем считать, что $N = 2^n$ и $[N] = \{0,1\}^n$.

II. Основные принципы PoW- и PoS-блокчейнов

В 1992 году С. Дворк и М. Наор [7] предложили механизм для защиты электронного почтового ящика от спама. Основная идея этого механизма заключалась в следующем: для того чтобы почтовый ящик принял письмо, отправитель вынужден произвести вычисления определенного (но «умеренного») объема. При этом должен существовать эффективный способ проверки того, что эта работа проведена. Для обычного пользователя, посылающего одно или несколько писем, данная работа занимает приемлемое время, но в случае массовой рассылки (спама) объем необходимых вычислений вырастает до таких величин, что сама рассылка становится нецелесообразной. Такое предложение явилось, по существу, принципиальным прототипом майнинга и протокола согласования, построенного на понятии Proof-of-Work (PoW) — доказательства работы.

На подобных идеях базируется и наиболее популярный тип рассматриваемого криптографического примитива — PoW-блокчейн. Содержательно это означает, что (в модели без контроля доступа) добавить запись в базу данных может всякий, кто решит «умеренно трудную» вычислительную задачу обращения (в определенном смысле) некоторой дискретной функции. Данный подход основан на проверке выполнения вычислительной работы при условии, что честные участники контролируют бóльшую часть вычислительных ресурсов.

При том что предлагается большое число разнообразных приложений, следует иметь в виду, что PoW-блокчейны обладают некоторыми очевидными недостатками. Во-первых, это стоимость энергии, потребляемой соответствующими системами (особенно если речь идет о масштабных реализациях, таких, как криптовалютные платежные системы). Во-вторых, вынужденное использование особо мощных вычислительных средств вместо обычных «бытовых» процессоров при реализации PoW-блокчейнов. Список недостатков можно продолжать и далее (например, упомянув необходимость передавать по сети и синхронизировать большие объемы информации и т. п.), однако дальнейшее изложение будет касаться главным образом этих двух проблем.

Собственно говоря, эти обстоятельства обусловили проведение исследований, направленных на поиск новых концепций протоколов согласования. В настоящее время имеется некоторое количество соответствующих предложений. Нас будет интересовать лишь одно из них: в 2015 году С. Дзембовский, С. Фауст, В. Колмогоров и К. Петшак [5] предложили концепцию Proof-of-Space, коротко обозначаемую буквосочетанием PoS (не путать с Proof-of-Stake). Блокчейны, построенные с использованием данного подхода, будем называть PoS-блокчейнами.

Разработка PoS-блокчейнов частично мотивирована следующим наблюдением: обычные пользователи вычислительной техники часто имеют на своих компьютерах значительные объемы свободной памяти. Эту память и предлагается использовать для решения «умеренно трудной» задачи — снижая тем самым количество необходимых вычислений.

Опишем кратко схему согласования в концепции PoS как протокол с двумя участниками — доказывающим P и проверяющим V.

В фазе инициализации доказывающий участник P (желающий поместить запись в блокчейн) получает определенную информацию объема $N \log N$, которую он должен сохранить в памяти. Проверяющему этот факт участнику V в процессе доказательства использования памяти требуется получить от P лишь небольшую часть этой информации.

Конкретизируя данный подход, предположим, что V предоставляет участнику P описание функции-перестановки $f : [N] \to [N]$ (например, посредством доступа к оракулу $\mathcal{O}^f$). Доказывающий P заполняет память объемом $N \log N$ таблицей значений функции f и производит ее сортировку, имеющую трудоемкость $O(N \log N)$. Участник V случайно и равновероятно выбирает элемент y из $[N]$ и пересылает его участнику P, который в свою очередь по таблице определяет $x = f^{-1}(y)$ и направляет x участнику V. В завершение протокола согласования V проверяет, выполняется ли равенство $y = f(x)$, и в зависимости от этого выносит свое решение, принимается ли доказательство.

Блокчейн с таким вариантом общей схемы достижения консенсуса будем условно называть *простым PoS-блокчейном*.

В работе [6] Абусалах и др. кратко обосновывают опасность использования такого простого протокола согласования, указывая на существование метода М. Хеллмана [8] обращения дискретных функций, использующего память порядка $\sqrt{N}$.

III. Усовершенствованная конструкция PoS-блокчейна

Своей целью авторы статьи [6] видят реабилитацию простого и элегантного, по их мнению, способа реализации протокола согласования в концепции PoS, основанного на задаче обращения функции. Ими предлагается строить для этого функцию, «ведущую себя, как случайная» на $[N]$, на основе двух других функций: перестановки f на $[N]$ и функции $g\colon [N] \times [N] \to [N]$. При этом предполагается считать f и g «случайными» функциями, значения которых участники протокола получают с помощью оракула.

Замечание 1. Авторы работы [6] значительное внимание (в разных ее частях) уделяют рассуждениям об использовании в их конструкции «случайных» функций. Однако эти рассуждения имеют нестрогий характер ввиду имеющихся сложных и еще не проявленных в полной мере взаимных связей между понятиями «случайная функция», «эффективно вычислимая функция», «сложность случайной функции» и т. п.

Относительно функции f авторы [6] указывают, что свойство перестановочности f не является обязательным, приводя в качестве доказательств нестрогие рассуждения. Учитывая, что сами авторы далее используют именно перестановку f, будем также придерживаться этого предположения.

Итак, с помощью пары функций f и g строится функция $g_f\colon [N] \to [N]$, где

$$g_f(x) = g\big(x, f^{-1}(\overline{f(x)})\big)$$

для любого $x \in [N]$ ($\overline{\ }$ — знак покомпонентного логического отрицания).

Участник протокола V (проверяющий) высылает доказывающему P описание функций f и g (предполагается, что это действие реализуется предоставлением участнику P доступа к оракулам $\mathcal{O}^f$ и $\mathcal{O}^g$, вычисляющим значения этих функций). Участник V выбирает случайным и равновероятным образом набор (строку) y из $[N]$ в качестве значения функции g_f. Далее V пересылает y участнику P и предлагает обратить функцию g_f в следующем смысле: для завершения протокола согласования P должен предъявить пару (x, x'), для которой $f(x') = \overline{f(x)}$ и $g(x, x') = y$. Проверяя справедливость этих равенств, V принимает или же отвергает доказательство.

Ожидается, что доказывающий P для нахождения прообраза некоторого случайного значения функции g_f использует память объема $O(N \log N)$: в типичном случае он строит таблицу значений функции f и сортирует ее (по $f(x)$), чтобы для всех x быстро находить $f^{-1}(\overline{f(x)})$, и, используя это, посредством вычисления функции g находит таблицу значений g_f, которая также сортируется. Благодаря хранению этих таблиц P после получения y может без труда найти сначала такой x, что $g_f(x) = y$, потом такой x', что $f(x') = \overline{f(x)}$. Проверяющему остается вычислить $g(x, x')$ (с помощью $\mathcal{O}^g$) и сравнить полученное с y.

В связи с обоснованием конструкции PoS-блокчейнов возникают, среди прочих, следующие вопросы:

- если участник V получает от P пару наборов (x, x'), удовлетворяющих вышеприведенным условиям, то как он может удостовериться, что участник P для нахождения этой пары хранил таблицы функций f и g_f объемом $O(N \log N)$ битов?
- существуют ли методы (алгоритмы) обращения функции g_f, использующие объемы памяти, существенно меньшие, чем $O(N \log N)$?

Частичный ответ дается авторами подхода в следующей формулировке требования к стойкости протокола: если нечестный доказывающий $\tilde{P}$ на самом деле не использует память требуемого объема, то ему, для того чтобы V принял доказательство, необходимо произвести достаточно много вычислений, то есть затратить определенное количество времени.

Анализу возможного ответа на эти вопросы и соответствия системы заявленным требованиям стойкости посвящен следующий раздел.

IV. Анализ конструкции PoS-блокчейна

По предположению авторов статьи [6], осуществляемое на предварительном этапе вычисление таблицы значений функции f с помощью $O(N)$ обращений к оракулу $\mathcal{O}^f$ и сортировка этой таблицы по значениям f с трудоемкостью $O(N \log N)$ являются приемлемыми вычислениями для участника P.

Замечание 2. Известно, что сортировка массива из N чисел может быть произведена за время $O(N \log N)$, например с использованием алгоритма пирамидальной сортировки (HeapSort).

Далее мы будем опускать множитель $\log N$, считая, что элементарные операции применяются к строкам из множества $[N]$, а не к их битам по отдельности, и оценивая требуемую память количеством ячеек, каждая из которых содержит одну строку из $[N]$ длиной $\log N$ битов.

Центральным местом данного подхода к построению протокола согласования является использование объема памяти порядка $N \log N$ битов (N ячеек по $\log N$ битов каждый). Конструкция основной функции g_f, построенной с помощью g и f, по мнению авторов, заставляет участника P занять именно такой объем памяти для вычисления прообраза заданного значения функции g_f. Однако доказательство этого утверждения имеет характер «правдоподобного рассуждения».

Рассмотрим метод определения прообраза, использующий память гораздо меньшего объема. При этом мы будем исходить из тех же предположений, что и авторы работы [6]. В частности, мы полагаем f полноцикловой перестановкой.

Пусть V передал участнику P набор $y \in [N]$ и предоставил ему доступ к оракулам $\mathcal{O}^g$ и $\mathcal{O}^f$. Как мы знаем, для того чтобы поместить некоторую запись в блокчейн, участнику P необходимо найти такую пару $(x, x') \in [N] \times [N]$, что $f(x') = \overline{f(x)}$ и $g_f(x) = g(x, x') = g(x, f^{-1}(\overline{f(x)})) = y$.

Уклоняясь от предписанного поведения, некоторый (нечестный) доказывающий $\tilde{P}$ может действовать следующим образом, используя модификацию метода обращения дискретных функций, входящего в класс методов, известных в криптографии под общим названием «больших и малых шагов» (baby-step/giant-step, см. [9], [10], [11]).

Выберем произвольный набор $x_1 \in [N]$ и запишем его в ячейку памяти доказывающего. Осуществив N обращений к оракулу $\mathcal{O}^f$, $\tilde{P}$ вычисляет орбиту этого элемента относительно перестановки f и, выделяя память из $\sqrt{N}$ ячеек (будем считать, что N — полный квадрат), заполняет ее, помещая в ячейку с адресом x_i набор $x_{i+1} = f^{\sqrt{N}}(x_i)$ для всех $1 \le i \le \sqrt{N} - 1$, а ячейке с набором x_1 присваивая адрес $x_{\sqrt{N}}$, то есть $\tilde{P}$ запоминает $\sqrt{N}$ элементов орбиты, «отстоящих» друг от друга на $\sqrt{N}$ в этой орбите. Массив данных $x_1, \ldots, x_{\sqrt{N}}$ обозначим через M_{gs}.

Получая от V значение y, участник $\tilde{P}$ перебирает (например, в лексикографическом порядке) наборы мно-

жества $[N]$, пока не придет к успеху, и для каждого текущего набора x выполняет следующие действия:

1) получает от оракула $\mathcal{O}^f$ значение $f(x)$;
2) обращая биты этого значения, вычисляет $\overline{f(x)} = z$;
3) формирует и запоминает массив M_{bs} из значений $z, f(z), \ldots, f^{\sqrt{N}-1}(z)$, для чего требуется $\sqrt{N}$ обращений к оракулу $\mathcal{O}^f$ и $\sqrt{N}$ ячеек памяти;
   [Массивы M_{gs} и M_{bs} имеют ровно один общий элемент!]
4) объединяет массивы M_{gs} и M_{bs} и сортирует их с целью найти общий элемент, при этом сортировка требует $O(\sqrt{N}\log N)$ операций и $O(\sqrt{N})$ ячеек памяти, а нахождение общего элемента в худшем случае $2\sqrt{N}-1$ попарных сравнений наборов; пусть общим элементом оказался набор x_j из M_{gs}, а $t \in \{0, \ldots, \sqrt{N}-1\}$ таково, что $f^t(z) = x_j$ (оно существует, так как $x_j \in M_{bs}$);
5) переходит к x_{j-1}, хранящемуся в ячейке с адресом x_{j-2}, и, обращаясь к оракулу $\sqrt{N}-t-1$ раз, вычисляет $x' = f^{\sqrt{N}-t-1}(x_{j-1})$;
6) обращаясь к оракулу $\mathcal{O}^g$, проверяет условие $g(x, x') \stackrel{?}{=} y$: если оно выполнено, ответом $\tilde{P}$ является (x, x'), иначе $\tilde{P}$ переходит к следующему набору x и повторяет для него шаги 1—6.

Данная процедура находит для y указанную пару, если она существует (в нашем случае, когда f — перестановка, это существенно зависит от свойств функции g). Действительно, по построению $f(x') = f^{\sqrt{N}-t}(x_{j-1}) = z$, так как $f^t(z) = x_j = f^{\sqrt{N}}(x_{j-1})$, а значит, $x' = f^{-1}(\overline{f(x)})$. Тем самым перебор всех $x \in [N]$ и построение для них x' указанным способом дает все возможные аргументы функции g, среди которых и прообраз значения y (если y принадлежит к образу функции g).

Нетрудно видеть, что описанная процедура в худшем случае предполагает $O(N^{3/2}\log N)$ обращений к оракулам и операций сравнения и логического отрицания и требует памяти объемом $O(\sqrt{N})$ ячеек (или $O(\sqrt{N}\log N)$ битов).

Таким образом, даже при достаточно грубом подходе к задаче взлома протокола и к оценке трудоемкости возможного метода криптоанализа у нечестного доказывающего нет необходимости занимать заявленный объем памяти порядка $O(N\log N)$, при этом количество вычислений, которые ему придется осуществить, по порядку отличается от «регламентного» лишь множителем $\sqrt{N}$.

Можно заключить, что в такой ситуации стойкость протокола согласования основывается на временно́й сложности задачи обращения функции g_f, то есть в итоге мы приходим к концепции Proof-of-Work. Выводом из такого заключения является, по крайней мере, то, что нельзя считать конструкцию PoS-блокчейна с подобным протоколом согласования достаточно обоснованной, а преимущества перед PoW-подходом очевидными. Анализ последних затрудняется отсутствием четкого описания функционирования протокола в условиях, определяемых характером использования блокчейнов: в частности, в сети с большим количеством участников, когда одновременно происходит множество попыток добавить новые блоки в цепочку.

Заметим еще, что описанный выше алгоритм использует лишь общую информацию о строении и свойствах функций g и f. Авторы работы [6] не формулируют никаких требований к функции g за исключением ее «случайности». Очевидно, что знание более тонких структурных особенностей этих функций позволило бы существенно понизить трудоемкость обращения функции g_f.

Приведем пример. Предположим, что участнику $\tilde{P}$ известно, что полноцикловая перестановка f является аффинным преобразованием векторного пространства $\{0,1\}^n$, то есть $f(x) = Ax \oplus b$, где A — невырожденная $(n \times n)$-матрица и $b \in \{0,1\}^n$, а $\oplus$ обозначает покомпонентное сложение по модулю 2. Тогда для произвольного $x \in \{0,1\}^n$ имеем $x' = f^{-1}(\overline{f(x)}) = A^{-1}(Ax \oplus 1^n) = x \oplus A^{-1}1^n$, где 1^n — n-мерный вектор, все компоненты которого равны 1. Трудоемкость вычисления x' для x составляет $O(n^2) = O(\log^2 N)$ битовых операций. Для каждой пары (x, x') выполнение условия $g(x, x') = y$ проверяется одним обращением к оракулу $\mathcal{O}^g$. Трудоемкость алгоритма обращения функции g_f в таком случае будет не более $O(N\log^2 N)$ элементарных операций над битами.

Таким образом, конкретизация классов, из которых выбираются функции f и g, может привести к значительному снижению трудоемкости методов взлома подобных PoS-блокчейнов. С другой стороны, без этой конкретизации конструкция не имеет определенных контуров, необходимых для построения прикладных систем.

V. Заключение

Временна́я трудоемкость обращения функции g_f и необходимая для этого память существенно зависят от свойств функций g и f и принимают значения в широком диапазоне. В этих условиях предложенная в [6] конструкция и сам принцип построения PoS-блокчейнов требует всестороннего и глубокого обоснования. Предлагаемые для PoS-блокчейнов протоколы согласования, как можно заключить из приведенного выше анализа, не предоставляют возможностей доказывать использование памяти требуемого объема. По сути, в конечном счете все сводится к PoW-блокчейнам с вытекающей отсюда необходимостью доказывать стойкость в соответствующей модели.

Единственным вероятным преимуществом предложенного подхода является потенциально меньший объем вычислений (и, соответственно, снижение энергозатрат) в случае честного выполнения протокола законными участниками. Однако, говоря о таких параметрах, мы в некотором смысле выходим за рамки сугубо теоретического подхода, и здесь не обойтись без более детального описания возможной реализации блокчейна. Нет достаточно веских оснований полагать, что реальные значения (не асимптотические оценки) вычислительных затрат у PoS-блокчейнов будут существенно меньше, чем для PoW-блокчейнов.

Сама вычислительная задача обращения дискретной функции является одним из центральных элементов в определениях основных криптографических примитивов, которыми оперирует математическая криптография, а также в определениях понятия стойкости криптографических систем. В ходе научных исследований разработано значительное число методов криптографического анализа, ориентированных на решение этой задачи.

Учет эффективности всего спектра известных методов криптоанализа является необходимым условием адекватности получаемых оценок возможности использования блокчейнов для защиты информации.

Библиография

[1] Варновский Н. П. Блокчейн как криптографический примитив // International Journal of Open Information Technologies (INJOIT). — 2020. — Т. 8, № 12. — С. 28–32.

[2] Тапскотт Д., Тапскотт А. Технология блокчейн. — М. : Эксмо, 2017.

[3] Дрешер Д. Основы блокчейна. Вводный курс для начинающих в 25 небольших главах. — М. : ДМК Пресс, 2018.

[4] Pass R., Seeman L., Shelat A. Analysis of the blockchain protocol in asynchronous networks // Advances in Cryptology—EUROCRYPT '17. — Vol. 10210 of Lecture Notes in Computer Science. — Berlin, Heidelberg : Springer, 2017. — P. 643–673.

[5] Proofs of space / S. Dziembowski, S. Faust, V. Kolmogorov, K. Pietrzak // Advances in Cryptology—CRYPTO '92. — Vol. 9216 of Lecture Notes in Computer Science. — Berlin, Heidelberg : Springer, 2015. — P. 585–605.

[6] Beyond Hellman's time–memory trade-offs with applications to proofs of space / H. Abusalah, J. Alwen, B. Cohen et al. // Asiacrypt 2017, Part II. — Vol. 10625 of Lecture Notes in Computer Science. — Berlin, Heidelberg : Springer, 2017. — P. 357–379.

[7] Dwork C., Naor M. Pricing via processing or combatting junk mail // Advances in Cryptology—CRYPTO '92. — Vol. 740 of Lecture Notes in Computer Science. — Berlin, Heidelberg : Springer, 1993. — P. 139–147.

[8] Hellman M. E. A cryptanalytic time–memory trade-off // IEEE Trans. on Information Theory. — 1980. — Vol. IT-26, no. 4. — P. 401–406.

[9] Katz J., Lindell Y. Introduction to Modern Cryptography. — London : CRC Press, 2007.

[10] Нечаев В. И. Элементы криптографии : Основы теории защиты информации. — М. : Высшая школа, 1999.

[11] Василенко О. Н. Теоретико-числовые алгоритмы в криптографии. — М. : МЦНМО, 2003.

On a class of descrete functions for Proof-of-Space blockchain consensus protocols

Oleg A. Logachev, Sergey N. Fedorov

Abstract—**The classic blockchain design implies the mining procedure, which is essentially reflected in the following: to add a new block to the chain, a user has to solve an instance of some moderately hard computational problem (Proof-of-Work framework, PoW). One of the most criticized points of this approach is that users have to perform a significant amount of computational work. As a result, the PoW-blockchains involve very high energy consumption.**

This led to the arising of a blockchain-related research area aimed at developing other ways to prove the right of a user to add a new block. In 2015, Dziembowski *et al.* suggested the Proof-of-Space concept (PoS): instead of spending a certain amount of time on computations, users should reserve a certain amount of disk space on their computers. This requirement can be implemented, for example, by requesting the user to invert some discrete function, so that a user who has stored the table of the function values could easily do it. In 2017, Abusalah *et al.*, trying to overcome some shortcomings of the original (simple) PoS, suggested to choose the function as a special composition of two discrete functions.

In this paper we analyze the idea of Abusalah *et al.* It is shown that this proposal does not meet the PoS requirement to reserve the specified amount of disk space. Also, we discuss the analysis of mathematical models of blockchains as cryptographic primitives.

Keywords—**blockchain, consensus protocol, Hellman method, Proof-of-Space**

References

[1] Varnovsky N. P. Blockchain as a cryptographic primitive // International Journal of Open Information Technologies (INJOIT). 2020. Vol. 8, no. 12. P. 28–32 [in Russian].

[2] Tapscott D., Tapscott A. The blockchain revolution: How the technology behind bitcoin is changing money, business, and the world. London : Penguin Books, 2016.

[3] Drescher D. Blockchain basics: A non-technical introduction in 25 steps. New York : Apress, 2017.

[4] Pass R., Seeman L., Shelat A. Analysis of the blockchain protocol in asynchronous networks // Advances in Cryptology—EUROCRYPT '17. Vol. 10210 of Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2017. P. 643–673.

[5] Proofs of space / S. Dziembowski, S. Faust, V. Kolmogorov, K. Pietrzak // Advances in Cryptology—CRYPTO '92. Vol. 9216 of Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2015. P. 585–605.

[6] Beyond Hellman's time–memory trade-offs with applications to proofs of space / H. Abusalah, J. Alwen, B. Cohen et al. // Asiacrypt 2017, Part II. Vol. 10625 of Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2017. P. 357–379.

[7] Dwork C., Naor M. Pricing via processing or combatting junk mail // Advances in Cryptology—CRYPTO '92. Vol. 740 of Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 1993. P. 139–147.

[8] Hellman M. E. A cryptanalytic time–memory trade-off // IEEE Trans. on Information Theory. 1980. Vol. IT-26, no. 4. P. 401–406.

[9] Katz J., Lindell Y. Introduction to Modern Cryptography. London : CRC Press, 2007.

[10] Nechayev V. I. Elementy kriptografii : Osnovy teorii zashchity informatsii. Moscow : Vysshaya shkola, 1999 [in Russian].

[11] Vasilenko O. N. Number-theoretic algorithms in cryptography. Providence (Rhode Island) : American Mathematical Society, 2006.