**ФОРМАЛИЗАЦИЯ СИСТЕМ УПРАВЛЕНИЯ РАСПРЕДЕЛЕННЫМИ ДАННЫМИ НА ОСНОВЕ ТЕХНОЛОГИИ «БЛОКЧЕЙН»**

КАЗЛОВСКИЙ М.А.

*Белорусский государственный университет*

 *Минск, 220030, Республика Беларусь*

1. **Введение**

Впервые появившаяся в криптовалюте *Bitcoin*, в настоящее время технология «блокчейн» используется не только в многочисленных криптовалютах, но и в криптоконтейнерах (реестры данных) и смарт-контрактах (финансовые сделки). Однако, несмотря на большую популярность технологии, до сих пор не построена ее формализованная модель, не проведен сравнительный анализ алгоритмов консенсуса. В данной работе описываются базовые элементы и операции, необходимые для формализации данной модели, сравниваются алгоритмы достижения консенсуса, рассматриваются свойства систем, построенных на технологии «блокчейн». Материалы работы могут быть использованы при проектировании и аудите систем, основанных на технологии «блокчейн».

1. **Объекты технологии «блокчейн»**

В качестве объектов технологии «блокчейн» будем рассматривать базовые элементы, над которыми субъекты могут производить операции:

* Транзакция (*tr*) – минимальная структурная единица, представляющая собой описание некого элементарного события (перечисление цифровых монет, добавление записи в реестр и т.п.) в рамках технологии «блокчейн». Обычно, помимо данных, которые описывают элементарное событие, содержит дополнительную служебную информацию (временную метку, хэш-значение данных, ЭЦП хэш-значения инициатором транзакции).
* Блок (*bl*) – заголовок (*hd*) и тело (*bd*). Заголовок – это служебная часть блока, которая содержит хэш-значение заголовка текущего блока, ссылку на предыдущий блок (хэш-значение заголовка предыдущего блока), индекс (порядковый номер блока в блокчейне) и временную метку. Также заголовок обычно содержит корень дерева Меркля (*mrklRoot*), построенного по транзакциям из тела блока, и параметр *nonce*, используемый при майнинге. Тело – это набор (список) транзакций.
* Блокчейн (*bc*) – набор блоков, каждый из которых указывает на своего предшественника(-ов) вплоть до генезис-блока (блок с индексом 0).

Формально, *bc = {bl0, bl1, …, bln-1},* где *bli = hdi + bdi = hdi + {tri1, tri2, …, trik}.*

Отметим, что все три объекта (*tr, bl, bc*) являются обязательными, то есть присутствуют в любой системе, основанной на технологии «блокчейн».

1. **Субъекты технологии «блокчейн»**

В качестве субъектов технологии «блокчейн» будем рассматривать основные роли, которые могут выполнять пользователи технологии:

* Клиент (*client*) – лицо, которое пользуется услугами системы.
* Майнер (*miner*) – лицо, которое добровольно используют свои вычислительные ресурсы для поддержания состоятельности блокчейна и получает за это определенное вознаграждение.
* Агент (*agent*) – выбранное или назначенное лицо, поддерживающее состоятельность блокчейна и, возможно, получающее за это вознаграждение; в случае выявления фактов участия агента в недобросовестных действиях, нарушающих стабильность системы, он может потерять часть своих средств.

Отметим, что обязательной в системе является только роль клиента. Роли майнера и агента являются опциональными, то есть могут отсутствовать. Если, при описании операции нам не важна роль, которую выполняет субъект, в качестве роли будем писать *client*, так как данная роль является базовой и обязательной.

1. **Операции технологии «блокчейн»**

 В качестве операций будем рассматривать основные действия, которые субъекты могут производить над объектами в технологии «блокчейн»:

* Создание пользователя (*CreateUser*): *⊥ → (sk, pk)*. Знак ⊥ кодирует отсутствие входных данных; на выходе – пара из открытого и личного ключей, пригодная для электронной цифровой подписи (ЭЦП).
* Создание транзакции (*CreateTr*)*: (data, sk) → tr.* На вход поступают данные и личный ключ клиента; на выходе – транзакция.
* Рассылка транзакции (*BroadcastTr*): *client → client\*: tr*. Клиент распространяет транзакцию среди остальных субъектов системы.
* Получение транзакции(*GetTr*)*: (tr, tr[]) → tr`[], tr`[] = tr[] + tr.* На вход поступают буфер транзакций и транзакция; на выходе – обновленный буфер транзакций (с добавленной транзакцией).
* Поиск транзакции (*FindTr*)*: (tr, bc) → {0, 1}.* На вход поступает транзакция и блокчейн, в котором будет осуществляться ее поиск; на выходе: 0 – транзакции не найдена или 1 – транзакция найдена.
* Проверка транзакции (*СheckTr*)*: tr → {0, 1}.* На вход поступает транзакция; на выходе: 0 – транзакция недействительная или 1 – транзакция действительная.
* Выбор транзакций (*ChooseTr*)*:* {*tr1, tr2, … trn*} *→* {*tr1`, tr2`, … trk`*}, *k ≤ n.* На вход поступает список ожидающих включения в блокчейн транзакций; на выходе – список транзакций, которые попадут в блок.
* Создание блока (*CreateBl*)*: ({[tr0], tr1, tr2, … trk}, prevHash, [pk]) → bl*, *k ≥ 1.* На вход поступает выбранный список транзакций (опциональная *tr0 –* вознаграждение за майнинг), хэш-значение заголовка предыдущего блока и, опционально, открытый ключ агента; на выходе – готовый блок.
* Рассылка блока (*BroadcastBl*): *client → client\*: bl*. Клиент осуществляет распространение блока среди остальных субъектов системы.
* Получение блока (*GetBl*)*: (bc, bl) → bс`, bc` = bc + bl*. На вход поступают блокчейн и новый блок; на выходе – обновленный блокчейн (с добавленным блоком).
* Проверка блока (*СheckBl*)*: bl → {0, 1}.* На вход поступает блок; на выходе: 0 – блок недействительный или 1 – блок действительный.
* Получение блокчейна (*GetBc*): *⊥ → bc*. На вход данные не поступают, на выходе получаем полную версию блокчейна (то есть все блоки, находящиеся в блокчейне на текущий момент времени).
* Проверка блокчейна (*СheckBс*)*: bс → {0, 1}.* На вход поступает блокчейн; на выходе: 0 – блокчейн недействительный или 1 – блокчейн действительный.
* Выборы агента (*ElectAgent*)*: ({client1,…, clientn}, [info]) → clienti*. На вход подается список претендующих на роль агента клиентов и, опционально, информация, которая позволит увеличить или уменьшить шансы конкретного кандидата; на выходе – выбранный в качестве агента клиент.
* Алгоритм консенсуса (*ConsensusAlg*)*: ({bls,…, blt}, …, {bll,…, blp}) → {bla,…, blb}*. На вход подается две и более цепочек из одного и более блоков; на выходе – цепочка, которая, согласно алгоритму консенсуса, должна войти в блокчейн.

Отметим, что из перечисленных операции обязательными являются все, кроме *ElectAgent* и *ConsensusAlg*.

1. **Описание механизмов консенсуса в технологии «блокчейн»**

Механизм консенсуса является важнейшим звеном в технологии «блокчейн». Он позволяет субъектам разрешать неизбежные в децентрализованных системах противоречия данных, вне зависимости от того, возникли эти противоречия случайно или были специально созданы недобросовестным субъектом.

 Рассмотрим основные механизмы консенсуса, используемые в настоящее время:

* *Proof of Work* – алгоритм, при котором создателем блока выступает тот, кто первым выполнит вычислительно сложную задачу и построит блок, хэш-значение которого меньше наперед заданного числа, при этом шансы каждого участника прямо пропорциональны вычислительной мощности его системы. [1]
* *Proof of Stake* – алгоритм, при котором создателем блока выступает тот, кто первым выполнит вычислительно сложную задачу и построит блок, хэш-значение которого меньше наперед заданного числа, при этом его шансы прямо пропорциональны сумме денег на его счете. [2]
* *Delegated Proof of Stake –* алгоритм, при котором создателем блока выступает случайно выбранный системой агент из списка агентов, составленного участниками системы. [3]
* *Proof of Activity* – алгоритм, при котором заготовка блока создается по алгоритму *PoW*, а сам блок получается после подписи заготовки случайно выбранными участниками системы из создателей транзакций, попавших в блок. [4]
* *Proof of Burn* – алгоритм, при котором создателем блока выступает тот, кто первым выполнит вычислительно сложную задачу и построит блок, хэш-значение которого меньше наперед заданного числа, при этом шансы каждого участника прямо пропорциональны количеству уничтоженных им денег. [5]
* *Proof of Capacity* – алгоритм, при котором создателем блока выступает тот, кто первым выполнит вычислительно сложную задачу и построит блок, хэш-значение которого меньше наперед заданного числа, при этом шансы каждого участника прямо пропорциональны количеству выделенного им дискового пространства. [5]
* *Proof of Authority* – алгоритм, при котором блок подписывается доверенным участником системы (агентом), выбранным системой из списка агентов. [6]
* *Proof of IOTA* – алгоритм, при котором для публикации блока клиент должен заверить два блока других клиентов, выбранных по определенному алгоритму. [7]

**Таблица 1.** Сравнение моделей доказательств

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Механизм консенсуса | Удобство для клиента | Удобство для майнера / агента | Безопасность  | Потенциал |
| *Proof of Work* | Высокое | Низкое | Средняя | Низкий |
| *Proof of Stake* | Высокое | Высокое | Низкая | Низкий |
| *Delegated* *Proof of Stake* | Среднее | Среднее | Средняя | Средний  |
| *Proof of Activity* | Среднее | Низкое | Средняя | Средний |
| *Proof of Burn* | Высокое | Низкое | Низкая | Отсутствует |
| *Proof of Capacity* | Высокое | Низкое | Низкая | Отсутствует |
| *Proof of Authority* | Высокое | Среднее | Высокая | Средний |
| *Proof of IOTA* | Среднее | – | Средняя | Высокий  |

**Таблица 2**. Безопасность механизмов консенсуса

|  |  |
| --- | --- |
| Тип атаки | Механизм консенсуса |
| *PoW* | *PoS* | *DPoS* | *PoA* | *PoB* | *PoC* | *PoAu* | *PoI* |
| Атака «ничего на кону»  | – | + | – | – | + | + | – | – |
| Атака взятками | – | + | – | – | + | + | – | + |
| Атака издалека | – | + | + | – | + | + | – | + |
| Атака накоплением возраста монет | – | + | – | – | – | – | – | – |
| Атака предвычислением | – | + | – | – | + | + | – | – |
| Отказ в обслуживании | + | + | + | + | + | + | – | + |
| Атака Сибиллы | + | + | + | + | + | + | – | + |
| Эгоистичный майнинг | + | – | – | + | – | – | – | – |

1. **Свойства технологии «блокчейн»**

Исходя из описанной выше концепции, проект, построенный на блокчейне должен удовлетворять ряду обязательных свойств:

* Прозрачность (открытость документации и исходного кода).
* Доступность (система должна функционировать на достаточном количестве узлов, не находящийся в рамках одной и той же физической сети).
* Целостность (информация, попавшая в блокчейн должна оставаться неизменной навсегда).
* Достоверность (любой узел сети должен иметь возможность самостоятельно провести проверки и доказать состоятельность блокчейна).
* Непротиворечивость (протокол консенсуса должен обеспечивать непротиворечивость данных, включаемых в блокчейн).
* Сходимость (протокол консенсуса должен за разумное время обеспечивать гарантированный уровень доверия к данным в блокчейне).
* Устойчивость (система должна успешно функционировать даже при выходе из сети до 1/3 узлов).
* Синхронизируемость (протокол должен обеспечивать формирование одинакового набора данных у всех узлов сети за разумное время).
* Распределенность (система должна функционировать в одноранговой (*peer-to-peer*) сети).

Кроме того, в зависимости от целей и задач конкретной системы, к ней могут выдвигаться дополнительные требования:

* Децентрализованность (в системе нет центра – все узлы сети имеют одинаковые права и возможности) / Централизованность (наличие в системе определенных узлов, имеющих более широкие права, чем остальные узлы).
* Самодостаточность (отсутствие необходимости обращаться к внешним ресурсам для проведения каких-либо операций) / Зависимость (необходимость при работе обращаться к сторонним ресурсам, например, к ГосСУОК).
* Анонимность (отсутствие жестких требований к идентификации нового узла сети) / Аутентифицируемость (необходимость аутентификации каждого узла сети при его работе в системе).
* Открытость (данные, хранящиеся в блокчейне, должны быть доступны для прочтения любому желающему) / Конфиденциальность (данные, хранящиеся в блокчейне, должны быть защищены от их несанкционированного прочтения).
* Подотчетность (возможность однозначно проследить действия любого узла сети).
* Аутентичность (возможность идентификации любого узла сети).
* Сертифицируемость (необходимость использовать сертифицированные в конкретной стране криптографические алгоритмы).
1. **Заключение**

Таким образом, в результате проделанной работы, были получены следующие результаты:

* определены и описаны основные элементы (объекты, субъекты и операции), характерные для любой системы на базе технологии «блокчейн»;
* проведен сравнительный анализ механизмов консенсуса и сформулированы основные свойства, которым должны удовлетворять системы, использующие технологию «блокчейн».

**Список литературы**

* + - 1. Bitcoin: A Peer-to-Peer Electronic Cash System [Electronic resource] / Satoshi Nakamoto. – 2008. – Mode of access: https://bitcoin.org/bitcoin.pdf. – Date of access: 14.04.2018.
			2. Обзор альтернатив Proof of Work. Часть 1. Proof of Stake [Электронный ресурс] – 2015. – Режим доступа: https://habrahabr.ru/post/265561. – Дата доступа: 14.04.2018.
			3. Proof of Stake versus Proof of Work [Electronic resource] / BitFury Group – 2015. – Mode of access: http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf. – Date of access: 14.04.2018.
			4. Proof of Activity: Extending Bitcoin’s Proof of Work via Proof of Stake [Electronic resource] / Iddo Bentov[et al.]. – 2018. – Mode of access: https://eprint.iacr.org/2014/452.pdf. – Date of access: 14.04.2018.
			5. Обзор альтернатив Proof of Work. Часть 2. Proof of Activity, Proof of Burn, Proof of Capacity и генералы [Электронный ресурс] – 2015. – Режим доступа: https://habrahabr.ru/post/266619. – Дата доступа: 14.05.2018.
			6. Proof-of-authority [Electronic resource]. / Wikipedia. – 2018. – Mode of access: https://en.wikipedia.org/wiki/Proof-of-authority. – Date of access: 14.04.2018.
			7. The Tangle [Electronic resource] / Serguei Popov. – 2018. – Mode of access: https://iota.org/IOTA\_Whitepaper.pdf. – Date of access: 14.04.2018.

**Сведения об авторах**

Казловский М.А, студент Белорусского государственного университета.

**Адрес для корреспонденции**

220030, Республика Беларусь

Минск, пр. Независимости, 4

Белорусский государственный университет

+375445773850

maxx-96@tut.by

Казловский Максим Анатольевич