Blockchain centralization and trust coefficients.

Yuriy Shumilov * [a,b], Alexey Bardin[c], Anatoliy Novitsky[c], and Andrei Kondratenko[d]

[a] *Bauman Moscow State Technical University, 2-ya Baumanskaya ul. 5, str.1, Moscow, 105005 Russian Federation*

[b] *Financial University Under Government of the Russian Federation, 49 Leningradsky Prospect, 125993, Moscow, Russian Federation*

[c] *Sumus Company Limited 2103, 21/F, Tower 1, Lippo Centre, 89 Queensway, Admiralty, Hong Kong, PR China*

[d] *PCBtech, 105082 26b, Bolshaya Pochtovaya, Moscow, Russian Federation*

Abstract

The paper proposes a method for evaluation the trust of a blockchain network user in the results of its work. The relevance of the research topic follows from the fact that today there are no ways to quantitatively determine the level of trust in the network, which does not depend on the user's degree of familiarity with the algorithm of its functioning and is sufficiently objective. To obtain the desired estimates, the concepts of three main types of network functioning are introduced: decentralized, chaotic and centralized. It is assumed that a necessary condition for trusting the results of the network operation is its decentralization, since the other two types of functioning indicate the emergence of random (chaos) or deterministic (centralization) coalitions of nodes, which contradicts the principle of equality of nodes in the network.

*Correspondig author.

Email addresses: **shumilovyy@gmail.com** (Yuriy Shumilov), **lovvi@mail.ru** (Alexey Bardin), **a@sumus.team** (Anatoliy Novitsky), **a.kondratenko@pcbtech.ru** (Andrei Kondratenko)

A function that is being considered that maps the set of values of the number of nodes to the set of values of the number of blocks closed by them. A quantitative evaluation of the centralization of the blockchain network is proposed in the form of a coefficient calculated on the basis of the median obtained on the basis of this distribution function. According to the values of the coefficient, it is assessed whether the network operates chaotically, centrally or decentralized. The paper presents the results of an evaluation of the centralization of the BITCOIN network based on PoW principles and a network using the stake distributed Byzantine Fault Tolerant (sdBFT) cryptographic consensus search algorithm, which indicate that the second network is significantly more decentralized than the first.

For decentralized networks, a sufficient condition of user confidence in the results of the network operation is proposed, which has the form of a coefficient calculated on the basis of the number of nodes potentially admitted in a given network to the block closing procedure and the number of nodes required in accordance with the network operation algorithm to close each current block. The trust coefficient takes into account both the availability of the block closing procedure for all nodes, and the security of the network in terms of blocking the activities of unscrupulous nodes and its resource intensity, on which its performance depends, which also affects the user's desire to work with this network, and therefore affects the trust to it.

The comparison of estimates of trust in the BITCOIN network over the time interval where it is decentralized and in the sdBFT network is presented; it is shown that the trust in the second network is two orders of magnitude higher than the trust in the first network.

Keywords :
centralization of the network, trust coefficients, hash function, digital signature, consensus.

1 Introduction.

Currently, there are many blockchain networks whose algorithms are based on different principles: Proof-of-Work (PoW) algorithm [1], PoS consensus algorithm, DPoS, LPoS, PoE, PoIT, pBFT [2, 3, 4], algorithm stake distributed Byzantine Fault Tolerant (sdBFT) [5], which eliminates the shortcomings of the pBFT algorithms.

However, an answer has not yet been formulated to one of the main questions that arise when evaluating the operation of a blockchain network: is the equality of nodes in

the network ensured in terms of uniformity of their loading when the blocks are closed. This state of the network can be called decentralized. Ideally, in a decentralized network, all nodes that work for a long time close the same number of blocks.

The decentralized state of the network is opposed by two other states that should be recognized as undesirable: randomness and centralization.

The operation of the network can be considered chaotic, as a result of which, at the time interval chosen for the study, most of the nodes close a small number of blocks each, but there is a smaller set of nodes in which each node closes enough blocks so that together these nodes close about the same number of blocks how many nodes from the first set cover.

The operation of the network can be considered centralized when, at the time interval chosen for the study, most nodes close one or two blocks each, and several nodes close all other blocks, the number of which is much greater than the number of blocks closed by the majority of nodes.

We can assume that a chaotically operating network violates the principle of equality of nodes to a lesser extent than a network operating centrally, but chaos indicates the possibility of the emergence of short-term and small coalitions of nodes covering about the same number of blocks as all the remaining nodes combined, which can be regarded as a conspiracy to carry out illegal actions, which the algorithm of this network cannot resist. A chaotic network is unstable, and the functioning of loyal nodes in it is unsafe.

Obviously, the centralized network is stable, but its state indicates the usurpation of the rights of the vast majority of nodes by a few select nodes. Evaluation of the level of centralization of the blockchain network is obviously relevant for the user.

Another problem is the evaluation of the level of trust on the part of node owners in the results of the blockchain network. The concept of trust in the results of a blockchain network is determined not only by the level of its centralization, but also by the extent to which the success of an individual offending node (unscrupulous node) in it is possible. You should also take into account the speed capabilities of the network algorithm when a large number of transactions are received per unit of time. The higher the speed of the network algorithm, the higher the level of trust in it from the side of the community applying it. Revealing the relationship between the level of centralization and the level of trust in the network is relevant.

It is fundamentally important that the sought evaluations of the level of centralization of the blockchain network and trust in it proceed from the objective results of its operation,

and not rely on the analysis of the algorithms of their functioning, since such evaluations would inevitably be influenced by the opinion of the algorithm developers laid down in the technical description of the network algorithms and turned out to be would be largely subjective.

2 Problem formulation.

Let the set Ω of cardinality α be the set of blocks closed at the moment t^*, $B_n \subseteq A_N$ -a set of n nodes, that took part in the closure of blocks from the set Ω, A_N — the set of all network nodes. As a rule, $\alpha \gg n$. We will assume that n and N are constant [6, 7]. It is assumed that the time segment chosen for the study of the blockchain network operation is $[0, t^*]$.

The task is to build a quantitative evaluation of the results of the blockchain network, which allows, based on the simplest parameters of its operation, to get a quick answer to the question about the level of centralization / decentralization / chaos of the network on the segment $[0, t^*]$.

Separately, the problem of evaluation the trust in the blockchain network should be posed. Let all the blockchain nodes included in the set be splited into K disjoint subsets B_n^i.

$$A_N = \cup_{i=1}^{K} B_n^i \tag{1}$$

Each set B_n^i is similar to the set B_n. We will say that a community focused on working with a given blockchain network trusts it if: the probability of a node getting into a set is high enough, the probability of success of an individual offending node in it is quite low, while the network should provide satisfactory performance at a high transaction rate into it. It is necessary to build a quantitative evaluation of the trust in the blockchain system.

3 Evaluation of the centralization of the blockchain network.

The approach proposed in this paper to evaluation the centralization of a blockchain network is based on the analysis of the results of the network operation, and not on the analysis of its algorithm in the sense of its ability to ensure the equality of nodes in the

network. It is this approach that seems to be the most significant from a practical point of view.

Consider the function $y(i)$, where $i = 0, 1, \ldots, \alpha$. Let's set the values of this function equal to the number of nodes in the blockchain network that have closed i blocks by the time. For example $y(0) = n_0$ — the number of nodes that did not close any block (did not work); $y(1) = n_1$ — the number of nodes that closed one block at a time; $y(2) = n_2$ — the number of nodes that closed 2 blocks each, ..., if $i = \left[\frac{\alpha}{n}\right]$, then $y(i) = n_{\left[\frac{\alpha}{n}\right]}$ is the number of nodes that have closed blocks by $\left[\frac{\alpha}{n}\right]$ blocks, ..., $y(\alpha) = n_\alpha$ is the number of nodes that have closed blocks by α. Obviously, it is either 0 or 1.

Consider some important special cases of the values of the function $y(i)$:

$n_\alpha = 1$ — the network is as centralized as possible and all blocks were closed by one node and the study of the centralization of the network can be completed;

$n_\alpha = 0$ — the study of network centralization can be continued;

$n_1 = n$, therefore $n = \alpha$ and each node closed one block at a time, respectively, the condition $\alpha \gg n$ is not met and this case cannot be considered typical for most networks, since this state of the network cannot be maintained for a long time: the number of closed blocks grows, and the number nodes in the network constantly. If we assume that the parameters n, N do change over time, then this happens regardless of the number of closed blocks, so it takes a long time to ensure that the equality $(1) = n$ is impossible. However, this case refers to the decentralized state of the network;

$n_{\left[\frac{\alpha}{n}\right]} = n$ — each node has closed $\left[\frac{\alpha}{n}\right] > 1$ nodes, which corresponds to ideal decentralization without chaos;

n_0 is close to $n, n_0 < n$: there are many non-working nodes in the blockchain and the problem statement needs to be revised.

Let's consider the simplest version of the centralization coefficient

$$i^* = \underset{i=0,\ldots,\alpha}{\operatorname{argmax}} \left(y(i)\right) \tag{2}$$

Consider the case when i^* is unique. Let's introduce the coefficient of centralization

$$K_c = 1 - \frac{\alpha}{n \cdot i^*},\ K_c \in [1 - \frac{\alpha}{n}, 1] \tag{3}$$

If $K_c < 0$, then this indicates a tendency towards chaotic operation of the system, when a significant majority of nodes close a small number of blocks each, but there is

a smaller set of nodes in which each node closes a lot of blocks, as a result of which all nodes from a smaller set close approximately as many blocks as most nodes cover.

For example, for $i^* = 1;\ \alpha \gg n$ we get $K_{c_{\min}} = 1 - \dfrac{\alpha}{n} < 0$

If the approximate equality is valid

$$K_c \approx 0, \tag{4}$$

decentralization without chaos is achieved at $i^* \approx \dfrac{\alpha}{n},\ y(i^*) \approx n$. If equality (3) becomes strict, then this corresponds to ideal decentralization in accordance with the assumption that the equality of nodes in the network is ensured in the sense of uniformity of their loading when the blocks are closed. Ideally, in a decentralized network, all nodes that work for a long time close the same number of blocks.

If the inequality

$$K_c > 0, \tag{5}$$

then this indicates a tendency towards centralization in the network. With complete centralization of the network, $K_{c_{\max}} = 1 - \dfrac{1}{n} > 0,\ i^* = \alpha,\ n_\alpha = 1$.

If $K^* = \max\left\{\left|K_c\right|, K_{c_{\max}}\right\}$, then you can enter the normalized centralization coefficient

$$K_{c_{\text{norm}}} = \frac{K_y}{K^*} \tag{6}$$

The disadvantage of calculating the centralization coefficient using formula (1) is that i^* is the mode of distribution determined by the function $y(i)$, which may not accurately estimate the ratio of the number of blocks covered by different sets of nodes.

It is proposed to calculate the coefficient K_c, using instead of the distribution mode i^* the average number $\langle i \rangle$ of blocks closed by one node, which is determined by the expression

$$\langle i \rangle = \underset{\tilde{i}=0,\ldots,\alpha-1}{\operatorname{argmin}} \left| \sum_{i=0}^{\tilde{i}} y(i) \cdot i - \sum_{i=\tilde{i}+1}^{\alpha} y(i) \cdot i \right|, \tag{7}$$

then, coefficient of centralization is

$$K_c = 1 - \frac{\alpha}{n \cdot \langle i \rangle} \tag{8}$$

In this case $\langle i \rangle$, will be the median of the distribution. Here, the uniqueness of $\langle i \rangle$ is also primarily assumed, since we choose this case of blockchain network operation as the main one. By the uniqueness of $\langle i \rangle$ we mean the existence of a unique value $i = \tilde{i}$ at which the minimum of the right-hand side of (7) is attained.

However, there are practically significant cases where $\langle i \rangle$ is not unique. The first such case is the performing an inequality $y(i) \neq 0$ calculations by formulas (7, 8) are not required here, since the network is maximally centralized for $\langle i \rangle = \alpha,\ K_c = 1 - \frac{1}{n}$.

The second case corresponds to the acceptance by the right-hand side of (7) of the minimum value on a finite ordered set of values $\langle i \rangle$ that follow in a row without gaps. Here, as $\langle i \rangle$, one should choose the largest value of $\tilde{i}$ from this set, since the change in the value of the right-hand side of (7) closest to the right will lead to a shift in the estimate of the centralization of the network towards an increase, which is possible only if the subtracted value in (8) decreases.

Using (7) requires more computation, but the result will be a more accurate estimate of centralization. The most preferred case is the coincidence of the K_c values obtained by formulas (3) and (8). If these values are significantly different, then the network cannot be recognized as decentralized. Note that the use of the mathematical expectation $\bar{i}$ instead of the mode or median will always lead to the same previously known result $\bar{i} = \alpha/n$, which will not allow studying systems with different levels of centralization, since K_c will always be zero.

4 Evaluation of the trust in the blockchain network.

As indicated in [8, 9, 10], all the nodes of the blockchain constitute the set A_N, distributed into m disjoint subsets B_n^i

$$A_N = \bigcup_{i=1}^{m} B_n^i \tag{9}$$

Nodes are included in each B_n^i, in accordance with the hypothesis of a uniform distribution of their numbers. Each set B_n^i has the same cardinality n and in its meaning is equivalent to the set B_n.

If B_n^i is the set of nodes that have the potential to take part in the closure of the

current block, then we should consider the set $B_{n'} \subseteq B_n^i \subseteq A_N$ of cardinality n', which includes the nodes required to execute the algorithm for closing each block.

Is it decentralized enough to trust the blockchain system? If the centralization coefficient is close to zero, then the nodes in such a system can be considered equal, but the question of the security of such a system and its ability to cope with the task of forming and closing a block at a high transaction rate and large n remains unresolved. With this formulation of the question, the decentralization of the system should be considered as a necessary condition for trusting it. In this section, an evaluation of the level of trust in the decentralized network is developed to determine which of the considered decentralized networks is more trustworthy from the user's point of view. As a result, a sufficient condition of trust in the blockchain network should be formulated.

Consider the influence of the cardinality of the set of nodes B_n^i, on the trust in the results of the network. Obviously, the larger n, the higher the level of trust in the network, since the more nodes are potentially allowed to make a decision to close the block, but at the same time, the smaller the ratio n/N, the lower the computational costs for closing the block and the lighter the network. copes with its tasks at a high transaction rate and a large number of nodes in the network.

Network security is also directly related to the ratio n/N, since the smaller it is, the lower the probability of getting into the set B_n^i of unscrupulous nodes. Indeed, suppose that the set B_n^i contains r "bad" nodes. Then, proceeding from the Bernoulli scheme, it is easy to show that the probability of l such nodes falling into the set $B_{n'}$ $(l \leq r, l \leq n')$ is defined by the formula

$$p = \frac{n'!}{l!(n'-l)} \left(\frac{r}{n}\right)^l \left(1 - \frac{r}{n}\right)^{n'1-l} \tag{10}$$

For $\dfrac{n'}{n} \to 0$, for example, when n, l and r are fixed, and $n \to \infty$, we obtain $p \to \infty$.

Then, the larger each of the quantities n', $\dfrac{n}{n'}$ the less $z(n') = \dfrac{1}{n'} + \dfrac{n'}{n}$. It is possible to consider $z(n')$ as the coefficient of "distrust". Finding the minimum for n', this function, we get the optimal value

$$n'_{opt} = \sqrt{N} \tag{11}$$

Then the coefficient of confidence in the results of the network operation can be set to the reciprocal, $K_T = 1/z(n')$ is the coefficient of confidence in the network, taking into

account its operability and security

$$K_T = (n' \cdot n)/(n' + n) \tag{12}$$

This coefficient reaches its maximum at $n' = n_{opt}$

$$K_{T_{\max}} = \frac{\sqrt{n}}{2} \tag{13}$$

The normalized confidence factor will have the form

$$K_{T_{\text{norm}}} = \frac{K_T}{K_{T_{\max}}} = (2n'\sqrt{n})/(n'^2 + n) \in [0, 1] \tag{14}$$

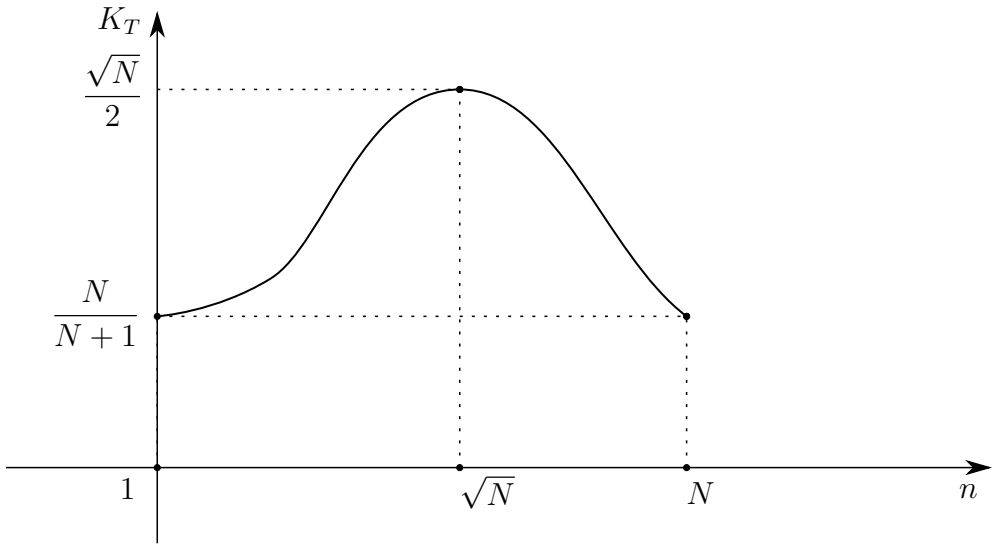

Figure 1: Dependence of the confidence coefficient K_T on the number of nodes in the set B_n.

5 Application of centralization and trust coefficients.

Let us take the BITCOIN network [11, 12, 13] as the first network for analyzing centralization and trust and consider the results of its operation for four consecutive sufficiently long time intervals, which are usually called epochs. For each of the four epochs, the function $y(i)$ was constructed and the centralization coefficient was calculated

First epoch: $\alpha = 210000$; $n = 186660$; $\left[\frac{\alpha}{n}\right] = 1$; $\langle i \rangle = 1$; $K_c = -0.125$. The blockchain works in an almost decentralized manner. Such a mode of operation cannot

be maintained for an arbitrarily long time, since it requires almost complete fulfillment of the condition $n = \alpha$.

Second epoch: $\alpha = 210000;\ n = 9774;\ \left[\frac{\alpha}{n}\right] = 5020;\ \langle i \rangle = 1;\ K_c = 0.996$. Corresponds to the centralized operation of the blockchain.

Third epoch: $\alpha = 210000;\ n = 517;\ \left[\frac{\alpha}{n}\right] = 8065;\ \langle i \rangle = 1;\ K_c = 0.953$. Corresponds to the centralized operation of the blockchain.

Fourth epoch: $\alpha = 45755;\ n = 118;\ \left[\frac{\alpha}{n}\right] = 2089;\ \langle i \rangle = 1;\ K_c = 0.814$. This epoch has not yet been completed, therefore the values of n, α differ significantly from the values for the previous completed epochs, however, from the value of the centralization coefficient, one can see that the work of the blockchain remains centralized.

In accordance with the necessary condition of trust in the blockchain network, the operation of the BITCOIN network in the second, third and fourth epocs cannot be considered credible: the centralization coefficient for these epochs is close to one.

The necessary trust condition is met only for the first epoch of the BITCOIN network operation, since the centralization factor in this case is close to zero. Let's calculate the coefficient of trust to this network for the first epoch. Since in this network, when making a decision on each block, the node that closes the block (we will call it the master node) can be any of N nodes, then we should put $n = N$. In this case, to close the block, the algorithm of this network requires one node, which means $n' = 1$ and as a result we get

$$K_T = \frac{n' \cdot n}{n'^2 + n} = \frac{n}{n+1} \approx 1 \tag{15}$$

For sufficiently large n, and for the first epoch $n = 186660$, K_T will always be close to 1. How high the level of trust in this network can be judged by the value of the normalized trust coefficient

$$K_{T_{\text{norm}}} = \frac{2\sqrt{n}}{n+1} \tag{16}$$

for $n = 186660$, we get $K_{T_{\text{norm}}} = 4.6 \cdot 10^{-3}$, which indicates a low level of trust in this network, despite its decentralization ($K_c = -0.125$) and security due to the large value of n and $n' = 1, p = \frac{r}{n}$. The resource intensity of the BITCOIN network, expressed by the number n of nodes potentially allowed to close the current block, is too large, which leads to a significant slowdown in the process of forming and closing blocks, the time of which reaches 10 minutes. The optimal n for a given network in the first epoch should be

approximately equal to 430, which would require a complete revision of the algorithm for the operation of this network.

Consider the second blockchain network, the algorithm of functioning of which is so different from the algorithm of the BITCOIN network that it becomes possible to achieve higher rates of decentralization and trust. This is the already mentioned cryptographic enterprise blockchain network built on the sdBFT algorithm, which is an improvement of the pBFT family of algorithms.

6 Initial positions of the sdBFT algorithm.

Achieving consensus is reduced to the receipt by the participants of the distributed system of an agreed decision, if a certain number of them did not take part in the coordination of the decision for the following reasons: an error in the transmission of a message about the adoption of a decision by one of the participants; too slow transmission of the message about the decision made by one of the participants; failure of a participant in the system; misleading when making a decision in the system, both intentional and unintentional.

For the first three reasons, a decision is possible if following condition $n > m + 1$ is fulfilled, where m is the number of nodes, which did not participate in the consensus building, and n is the number of participants which made a decision. In the case of the fourth reason, we get the problem of the Byzantine generals, which has a solution when $n > m + 1$[14, 15, 16].

Let us specify the function

$$f : Y_{B_n} \mapsto J_n,\ f = f(d),\ d \in Y_{B_n} \tag{17}$$

where J_n, $|J_n| = n$, set of node numbers from B_n.

The values of the function f are determined as follows: a double hash of a binary number $d \in Y_{B_n}$, is calculated, a pseudo-random bit sequence is constructed [17, 18, 19] $j_1 = H(H(d)),\ j_2 = H(H(d+1)),\ \ldots.$ We get the next bit record $R = j_1 \| j_2 \| \ldots$, sequentially splitted without gaps and overlaps into tuples of r bits in each, the first of which are the numbers of the nodes that form the set $B_{n'},\ j_k \in J_n,\ k = 1, \ldots, n'$.

The master node will always be the node whose number $j_{\hat{k}}$ is formed first ($\hat{k} = 1$). If an already received number $j_{\hat{k}}$ is accidentally repeated, then the repeated received

number is skipped. Thus, from randomly [20] a subset B_n is selected with a given n', then $B_{n'} \subset B_n \subseteq A_N$, and $Y_{B_{n'}}$ is a set of values d corresponding to all current states where is a set of values corresponding to all current states B_n.

To solve the problem of an increase in the time to reach a consensus with an increase in the number of nodes that can potentially be involved in consensus building, it is proposed to select from B_n a subset $B_{n'}$ $(|B_{n'}| = n')$ in the manner described above. Due to the use of double hashing, the numbers of nodes in the set B_n, when choosing the numbers of nodes from it for building the set, will be "mixed" well enough so that the numbers of nodes from B_n can be considered uniformly distributed, that is a well-known property of hash functions and follows directly from [21]. In this case, it is possible to solve the problem not on a set, but on, which will reduce the computational complexity of the algorithm for achieving consensus by about a factor $\left[\frac{n}{n'}\right]$ and significantly reduce the block closing time in comparison with the algorithms developed earlier.

Since the algorithms of the pBFT family have acceptable performance when the order of 10^4 transactions per second enters the blockchain with only a small number of nodes in the blockchain, the intruder needs to solve the cryptanalytic problem of obtaining keys for only 2/3 of this number of nodes. In the case of the spBFT algorithm, when the number of nodes increases by a factor of k, the complexity of the cryptanalytic problem for the intruder also increases by a factor of k and, accordingly, the cryptographic strength of the sdBFT algorithm increases.

If is a block in relation to which at some point in time a set of nodes strives to reach a consensus, then the SHA-3 hashing function [22] over this block will be denoted by $H(b)$, and its values will be denoted by h. Almost any of the modern algorithms is suitable for calculating an electronic signature, since their influence on the characteristics of the main algorithm is insignificant. In this work, the popular algorithm for calculating the electronic signature EdDSA with the parameters of the elliptic curve *edwards25519* was chosen [23, 24]. In the accepted notation, the result of calculating the signature will be equal to $s = \mathrm{sig}(h)$.

7 sdBFT algorithm.

1. Let at the moment of time $\hat{t} \in [t, t')$, where $[t, t')$ is the half-interval at which the block should be created, the node with the number writes record I to the blockchain.

2. Let's select all j_k, including $j_{\hat{k}}$ by means of f function. Generating multiple nodes

consensus is carried out on a half-interval $[t, t')$.

3. If the master node admits the inclusion of the record I into a block, the master node sends this record to all nodes from $B_{n'}$ for verification and inclusion it in the block. Otherwise, the record I will be rejected without notice.

4. The new record is included in the block until the moment t'. The master node sends a message to the same nodes to commit block b. All nodes from $B_{n'}$ calculate the value of the hash function $H(b)$ equal to, say, h.

5. Each node computes an electronic signature $s_b = (s_1, \ldots, s_{j_k})$ and transmits it to the master node.

6. The master node is waiting for electronic signatures time Δt after the moment t'. At the moment $t' + \Delta t$, a tuple is formed on the master node

$$s_b = (s_1, \ldots, s_{j_k}),\ 1 \leq k < n' \tag{18}$$

The master node verifies each signature from (1) and counts the number of valid signatures. The signatures of some of the nodes from $B_{n'}$ may turn out to be voting negatively or incorrect when in $B_{n'}$ is a node with a certain number $j_{\tilde{k}}$, $1 \leq \tilde{k} \leq n'$ that: a) recognizes the entry as incorrect; b) at a moment in time has a state of the blockchain that is different from the state for the node; c) distort the record when forming the block of the blockchain.

7. The master node calculates the number of correct signatures and checks the inequality

$$\mu > \left[\frac{2}{3}n'\right] \tag{19}$$

If (18) fails, then the master concludes that no consensus has been reached and the transaction is not included in the block, otherwise the transaction is included in the block.

8. If the current time is in a half-interval $[t, t')$, then go to step 1. If not, go to step 9.

9. For the block b, a number $b\|s_{k_1}\| \ldots \|s_\mu$ is compiled and calculated $H(b\|s_{k_1}\| \ldots \|s_\mu)$, as well as sig, which is the electronic signature of the node with a number $j_{\hat{k}}$.

10. Number

$$d' = b\|s_1\| \ldots \|s_\mu\| \operatorname{sig}(H(b\|s_1\| \ldots \|s_\mu)) \tag{20}$$

corresponding to the new closed block, we will consider the new state of the blockchain d' at the moment t'. The master node sends it to all nodes in the set A_N.

11. At each node from A_N, s_b and sig are checked. If the check is passed, then the block b is recognized as correct and the blockchain on the node goes into the state $d' = d(t')$. If this node has not received a block for verification in the time interval $[t' + \Delta t,\ t' + \Delta t + \lambda]$, where λ is the information transfer delay time, then the node will consider the consensus unreached and will choose a new set $B_{n'}$ based on the old state d.

This article proposes an evaluation of the centralization of a blockchain network of this type. The network parameters are as follows: $N = 10^4,\ n = 50,\ n' = 5$, the number of blocks closed on the observation interval $\alpha = 647587$.

As a result, at $\langle i \rangle = 13293$, the centralization coefficient $K_c = 2.6 \cdot 10^{-2}$ was obtained. Thus, the level of decentralization of the network with sdBFT algorithm (second network) is about 5 times higher than that of the first. The prerequisite for trusting the sdBFT network has been met.

Calculation of the normalized confidence factor for the second network gave the following result: $K_{T_{\text{norm}}} = 0.93$. This trust factor is very close to one and is about 200 times higher than the BITCOIN network trust factor. In the second network, due to the lower resource consumption and high speed of the sdBFT algorithm, it takes no more than 20 seconds to close the block, which is 30 times less than in the first network, which is one of the reasons for the increase in the trust coefficient.

The second reason for the high confidence factor of the second network is its high security. Under the assumption of a uniform distribution of the properties of nodes in the network in the presence of r bad nodes in A_N, the probability that there will be one bad node in $B_{n'}$, and it is this node that will be the master node (the node that closes the current block), according to algorithm sdBFT, is equal to $p = r/N$, which for the same A_N in both networks gives the same value of p.

It is very unlikely that two unscrupulous nodes get into the set $B_{n'}$ in the second network, and most importantly, the possibility of these unscrupulous nodes influencing the block closure is blocked by the sdBFT algorithm. More than two unscrupulous nodes in the second network hitting $B_{n'}$ are so unlikely that this event can be neglected.

8 Conclusion.

The article proposes a holistic system for evaluation the level of trust in the results of the blockchain network operation, based on the consideration of three main types of network functioning: chaotic, decentralized and centralized. According to the principles

of blockchain, only decentralized functioning of the network is acceptable for the network user.

A necessary condition of trust in the blockchain network is formulated, based on the calculation of a convenient estimate of the uniformity of the load of the blockchain network nodes in the process of closing the blocks. This estimate is quite simple and is calculated as a coefficient that takes into account the number of closed blocks, the number of nodes that closed a certain number of blocks and is called the centralization coefficient.

In a normalized form, this coefficient takes values from -1 to 1, which corresponds to the state of the network from complete chaos (-1) to complete centralization (1), which allows you to quickly assess whether a given blockchain network at a certain time interval corresponds to the main idea of the blockchain — equality and independence of nodes (decentralization), in accordance with which the coefficient takes on a zero value.

As studies of the BITCOIN network have shown on four time intervals (epochs), this network can be in a decentralized state for a long time, provided that the number of nodes is linearly related to the number of blocks they close. However, this kind of network functioning existed only on one time interval, called the first epoch, for which the centralization coefficient was close enough to zero.

However, at other time intervals (second, third and fourth epochs), the centralization coefficient is very close to unity, which indicates an unacceptable level of centralization in its work.

Comparison of the BITCOIN network in terms of centralization and trust with a network using the cryptanalytic consensus algorithm sdBFT showed that the second network is much more decentralized than the first, which indicates the possibility of further investigation of the level of trust in it.

To study the trust in blockchain networks, a sufficient condition of trust in the blockchain network was formulated, based on the calculation of the trust coefficient for a decentralized network. This factor takes into account both the possibility for each node to become a master node (to become the node that closes the block), and the resource intensity and security of the network. The resource intensity of the network is considered primarily from the point of view of the number of nodes involved in the decision to close each block. Network security is assessed as the ability of the network to resist the influence of unscrupulous nodes on the results of its work. If the normalized trust coefficient is in a sufficiently small left neighborhood of unity, then the network has a high level of trust.

Comparison of the BITCOIN network during its operation in the first era and the

network with the sdBFT algorithm showed that the level of trust in the second network is two orders of magnitude higher than the level of trust in the first network.

Based on the results of the studies, it can be concluded that the proposed necessary and sufficient conditions of trust are a convenient and reliable method for evaluation the acceptability of the blockchain network for users. Networks using the sdBFT algorithm have a significantly higher level of confidence in their results than networks based on PoW.

References

[1] Satoshi Nakamoto (2009). Bitcoin: A Peer-to-Peer Electronic Cash System. Available at: www.bitcoin.org (accessed: 20.01.19)

[2] Chepurnoy A, Larangeira M, Ojiganovarxiv A. Rollerchain, a Blockchain With Safely Pruneable Full Blocks. — Available at: https://arxiv.org/pdf/1603.07926v3.pdf (accessed: 18.12.20)

[3] Transactions Speeds: How Do Cryptocurrencies Stack Up To Visa or PayPal? Available at: https://howmuch.net/articles/crypto-transaction-speeds-compared (accessed: 12.08.19)

[4] BitFury Group (2015.09.13). Proof of Stake versus Proof of Work. Available at: http://bitfury.com/content/5-white-apersresearch/pos-vs-pow-1.0.2.pdf (accessed: 02.10.19)

[5] Eric Budish (June 5, 2018). The Economic Limits of Bitcoin and the Blockchain. Available at: https://faculty.chicagobooth.edu/ (accessed: 28.01.21)

[6] Reza Barzegar Nozari, Hamidreza Koohi. Novel implicit-trust-network-based recommendation methodology. *Expert Systems and Applications*, 2021, December, vol. 186, pp.1-22. DOI://doi.org/10.1016/j.eswa.2021.115709

[7] De Groot. Optimal statistical solutions. M.: Mir, 1974, 492 p.

[8] Lehmann, Erich L; Romano, Joseph P (2006-03-30). Testing Statistical Hypotheses. Business Media, 2006, 786 p.

[9] Ethereum. Available at: https://bits.media/ethereum/ (accessed: 26.05.20)

[10] Kudryavtsev K.Ya. Proof of the normality of the distribution of a subset of random variables based on the transformation of block matrices. *Herald of NIYaU ≪MEPhI≫*. 2021. Vol. 10, 1, pp.58-64

[11] Deon, A.F., Menyaev, Yu.A. Polnoe faktorialnoe modelirovanie ravnomernykh posledovatelnostey tselykh sluchaynykh velichin. // *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana. Priborostr* [Herald of the Bauman Moscow State Tech. Univ., Instrum. Eng.], 2017, 5, pp.132-149 (in Russ.). DOI: http://dx.doi.org/10.18698/0236-3933-2017-5-132-149

[12] Applebaum, Benny. Pseudorandom generators with long stretch and low locality from random local one-way functions. *Proc. 44-th Annual ACM Symposium on Theory of Computing*, New York, ACM, 2012. pp.805-816. DOI: http://dx.doi.org/10.1145/2213977.2214050

[13] Lewis, T.G., Payne, W.H. Generalized feedback shift register pseudorandom number algorithm. // *J. ACM.* 1973. Vol. 20 3, pp.456-486. DOI: http://dx.doi.org/10.1145/321765.321777

[14] Sunklodas, J. On normal approximations to strongly mixing random fields. *Theory Probab. Appl*, 2010. Vol. 52, 1, pp.125-132.

[15] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, Dawn Song. The Honey Badger of BFT Protocols. Available at: https://eprint.iacr.org/2016/199.pdf (accessed: 26.05.2020)

[16] Consensus Algorithm for Bigger Blockchain Networks. Available at: https://files.sumustech.com/doc/1-86cd7815.pdf (accessed: 14.04.2021)

[17] Goldmint Blockchain Solutions: Consensus Algorithm Designed by Sumus Team for Bigger Blockchain Networks. Available at: https://blog.goldmint.io/goldmint-blockchain-solutions-consensus-algorithm-designed-by-sumus-team-for-bigger-blockchain-6ace5fd3ee6d. (accessed: 30.10.2020).

[18] Leslie Lamport, Robert Shostak, Marshall Pease. The Byzantine Generals Problem. // *ACM Transactions on Programming Languages and Systems.* 1982. Vol. 4, 3, pp.382–401.

[19] Classen K., Palka M.H. Splittable pseudorandom number generators using cryptographic hashing. *Proc. 2013 ACM SIGPLAN Symp. on Haskell*, New York, ACM. 2013. pp.47-58. DOI: http://dx.doi.org/10.1145/2503778.2503784

[20] Siham Hattab, Imad Fakhri, Taha Alyaseen. Consensus Algorithms Blockchain: A comparative study. *International Journal on Perspective and Cognitive Computing*, vol. 5, issue 2, pp.66-71.

[21] National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Available at: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf (accessed: 06.11.2020)

[22] S. Josefsson, I. Liusvaara. Edwards-Curve Digital Signature Algorithm (EdDSA). IETF RFC. Available at: https://tools.ietf.org/html/rfc8032 (accessed: 08.04.21)

[23] Varfolomeev, A.A. Some recommendations for improving security of the cipher with small key against brute force attack. Voprosy kiberbezopasnosti [Cibersecurity Issues], 2015, 5 pp.60-62 (in Russ.)

[24] Varfolomeev, A.A. The realization of one proxy digital signature scheme of the base of Russian standarts. Bezopasnost' informatsionnykh tekhnologiy [IT Security], 2010, vol. 17, 1, pp.50-51 (in Russ.)