Simulation of a Double Spending Attack on the Proof of Work Consensus Protocol

Nikolay Poluyanenko [1] [0000-0001-9386-2547], Nadia Pisarenko [1] [0000-0002-3122-9129], Vladyslav Safonenko [1] [0000-0001-9514-0433], Tymur Makushenko [2] [0000-0001-7124-9610], Olha Pushko [3] [0000-0002-7507-3541] and Yevhena Zaburmekha [4] [0000-0003-2223-3887], Kateryna Kuznetsova [1][0000-0002-5605-9293]

[1] V. N. Karazin Kharkiv National University, Kharkiv, Ukraine
nlfsr01@gmail.com, pisarenko108@gmail.com, vladyslavsafonenko@gmail.com, k.kuznetsova@ukr.net
[2] Kharkiv University of Technology «STEP», Kharkiv, Ukraine
makushenko@itstep.org
[3] Sumy State University, Sumy, Ukraine
o.syniavska@uabs.sumdu.edu.ua
[4] Khmelnytskyi National University, Khmelnytskyi, Ukraine
zaburmehaem@ukr.net

Abstract. A critical analysis of the well-known analytical estimates of the probability of successful implementation of a double-spending attack on the Proof of work consensus protocol has been carried out. In particular, the so-called “The player’s ruin task”, it is shown that the basic assumptions about the probability space (the set of elementary outcomes and the probability of their occurrence) do not correspond to the real processes that occur when the “Proof of work” consensus is established in the blockchain system. A model of “independent players” is proposed, which eliminates the main inaccuracies and inconsistencies. The convergence of the results of theoretical calculations with the data of experiments to simulate the “race” between honest players and attackers is shown.

Keywords: blockchain; consensus protocol; double spending attack; simulation and modeling.

1 Introduction

The most famous attack on the consensus protocols of blockchain systems is the so-called double spending attack, when an dishonest participant in a decentralized system makes a repeated alienation (sale) of the same digital assets (units of cryptocurrency, tokens, coins, etc.), i.e. realizes several illegal payments from the same starting state [1, 2]. If a significant period of time elapses between the conclusion of the transaction and the registration of the transfer of ownership, then the seller may try to sell the same product several times to different buyers, receiving several times payment for the same asset. The most urgent task of preventing double spending becomes in electronic payment systems. Digital assets are easily copied, and a dishonest participant can transfer their copies to a large number of customers. Each recipient can make sure that the received asset is fully consistent with the declared characteristics, but will not be sure that they did not pay the same copy with another participant in the system.

In traditional (centralized) systems, the task of preventing double spending is solved by the application of administrative measures, when the centralized (to which everyone is subordinate) node controls the admissibility of an operation. In order to prevent double spending in a decentralized distributed system, consensus adoption protocols are responsible for which transaction is considered true [1-3]. This mechanism allows (in the ideal case) to ignore attempts to double the use of the same digital assets.

The first and most studied protocol for the consensus of decentralized systems is the Proof of work algorithm [3, 4]. It is based on the solution of a complex computational problem (as a rule, the search for the inverse image of the cryptographic hashing function). And only the one who first solves this problem (finds a suitable prototype) will get the right to make a change to the state of the system [4]. In fact, this means the possibility of a transaction with the alienation (sale, payment, etc.) of digital assets. Thus, the task of preventing double spending is to exclude (or at least reduce the likelihood) the possible formation of the prototype by the same participant in the system. In practice, this is achieved by involving a huge number of participants with an appropriate distribution of their computing capabilities to search for prototypes of the cryptographic hash function. In addition, each participant has the right to transfer rights to his assets only after a certain number of pre-formed prototypes, thereby reducing the likelihood of double spending.

The first results on estimating the probability of double spending in a decentralized Bitcoin system were published in the original article by S. Nakamoto [4], as well as in the work of M. Rosenfeld [5]. These are the most popular and cited works in the field. There are also other publications that, for different cases, refine and supplement the results obtained by S. Nakamoto and M. Rosenfeld:

- the results of Carlos Pinson and Camilo Rocha [6], constructing models of double-spending attacks based on not only the hashrate (computational capabilities) of the attacker and an honest network, but also taking into account the influence of time parameters. The equations governing these models use the Erlang probability distribution, in contrast to the work of S. Nakamoto, who uses the Poisson probability distribution, and the work of M. Rosenfeld, who uses the negative binomial probability distribution;
- the results of Kovalchuk [7], which generalize and partially develop well-known estimates, which also take into account the time of transaction confirmation;
- Azzolini's work [8] in which probabilistic logical programming is used. Allegedly, this method allows you to take into account the time-varying hash rate and the variable complexity of the Proof of work algorithm;

- the results of Kevin Liao, presented in [9] and considering a whale attack, in which a minority attacker increases his chances of successfully conducting a double waste attack, encouraging miners to undermine the agreed protocol and enter into conspiracy through whale transactions or anomalously large transactions fees.

It should be noted that the known estimates are obtained as a result of some simplifications and assumptions, i.e. the models used, as a rule, give approximate values, and the main criticism of these estimates is their unrealism, isolation from real processes occurring in decentralized systems. In particular, island inaccuracies and false assumptions in the works of S. Nakamoto and M. Rosenfeld are:

- the probabilities of forming a block by an honest network and the attacker in total should be equal to one. However, the above expressions do not give an answer what will be the result with independent values of these probabilities [10];
- the economic opportunity to form blocks by an attacker, as well as economic feasibility, are not taken into account. The resources of an attacker to maintain a race between an attacker and an honest network are considered unlimited, which cannot be true [11];
- it is assumed that the probability of success in forming a block does not change during the experiment, although, in reality, miners can change their probabilities of finding the right prototype and forming a block, increasing or decreasing their computing resources [11];
- in the work of M. Rosenfeld, a theorem on the probability of success by an attacker is presented without proof and obtained with the assumption that the block propagation time in the network is zero, in [12] it is mentioned that the network synchronization time must be taken into account;
- the assumption about the formation of blocks in accordance with the average waiting time of a block made by S. Nakamoto is erroneous [13].

Unfortunately, there are not many works in which an attempt is made to experimentally confirm or refute the obtained theoretical calculations, i.e. empirically justify the adequacy of the selected mathematical model. Such works include [11] and [8].

In all the mentioned works, the player’s ruin model is used, and it is verified by Monte Carlo methods. Based on this model, a formula is derived for calculating the probability of a successful double spend attack.

The aim of this work is a critical analysis of known analytical estimates of the likelihood of a successful implementation of a double spending attack on the Proof of work consensus protocol. In particular, we consider the so-called “The player’s ruin problem” underlying the models of S. Nakamoto and M. Rosenfeld and show that the basic assumptions about the probability space (the set of elementary outcomes and the probability of their occurrence) do not correspond to the real processes that take place during the establishment of the consensus “Proof of work” in the blockchain system. Further, for a theoretical assessment of the probability of a successful double-spending attack, we propose using the “independent players” model, which, in our opinion, eliminates the main inaccuracies and inconsistencies. Empirically, we show the convergence of the results of theoretical calculations with the data of experiments to simulate a "race" between honest players and attackers. The most interesting, in our opinion, is a comparison of the results of theoretical calculations obtained using various models and the empirical results obtained by simulating a “race”.

2 “Player Busting Challenge” can be applied to a double spend attack

Consider the “player ruin problem”, or rather, its small modification, which S. Nakamoto refers to, citing the well-known textbook of Feller 1968 [14], or M. Rosenfeld, modeling the race process as a process equivalent to the Markov chain with discrete time, where each a step is defined as someone looking for a block.

First, we cite an excerpt from section 11 of S. Nakamoto's work [4], which discusses the modeling of a double spending attack:

“The race between honest participants and an attacker can be imagined as a binomial random walk. A successful event, when an “honest” chain increases by one block, leads to an increase in separation by one, increasing its advantage by, and an unsuccessful one when an attacker creates another block leads to its reduction by one block, reducing the gap by. The probability of an attacker to make up the difference in several blocks is the same as in the task of “ruining a player”. Imagine that a player has unlimited credit, starts with some deficit and has infinitely many attempts to recoup.”

Next, we would like to give an excerpt from section 3 of the work of M. Rosenfeld [5]:

“Let’s denote $z = n - m$by the number of blocks in which the honest network has an advantage over the attacker. Whenever a block is found, z value changes; if this block was detected by an honest network, z increases by 1, and if this block was detected by an attacker, z decreases by 1. Formally, this is a Markov chain with p/T_0 continuous time and q/T_0 speed for moving up a step, and speed for moving down a step. ”

As you can see, in these works, a model is used in which the attacker wins in each test (forming the next block) or the attacker loses and it is considered that the honest network wins (forming the next block). However, the articles do not provide any justification for the selected model. The authors admit that if the attacker did not form the block, then, in this case, the block necessarily forms an honest network, and this assumption does not substantiate in any way.

Indeed, the probability space with two elementary events is used in the definition of a player’s bankruptcy problem: “the first player won”; “won the second player”. When simulating a double-spending attack, S. Nakamoto and M. Rosenfeld interpret the elementary outcomes of this task as “a block is formed by an honest network” (according to tradition, the probability of such an outcome is designated as p) and “a block is formed by an attacker” (with probability q), moreover $p = 1 - q$. However, in real blockchain systems, the probability of the formation of a block (finding the prototype of the hash function) is determined exclusively by the hashrate (computational capabilities) of each participant, i.e. the condition $p=1-q$ is not required to be met. For example, if the hash of the participants exceeds the complexity of searching for the inverse image for a certain time interval, each participant is guaranteed to find the inverse image, i.e. will form a block and, in this case, $p=1$ and $q=1$. In real systems, the complexity of searching for the prototype is adjusted based on the computing capabilities of the participants, so that the prototype is found for a certain time interval (for example, in cryptocurrency, bitcoin is 10 minutes). If we assume that such an adjustment is performed on two players: the "honest network" and "the attacker," and p and q are the corresponding probabilities of the formation of a block for a certain time interval, then the assumption $p=1-q$ is justified. However, in a real situation, an attacker attacks the system without disclosing his computing capabilities and, most likely, hiding the very fact of the alleged attack, i.e. the assumption $p=1-q$ has no reason.

If we leave the introduced notation (probabilities p and q) and refuse to fulfill the condition $p=1-q$, then as a result of each attempt (or series of attempts during a given time interval), the space of elementary outcomes contains the following events:

- the elementary event "a block is formed by an honest network and the attacker did not form a block" with probability $p(1-q)$;
- the elementary event "the block is not formed by an honest network and the attacker formed a block" with probability $(1-p)q$;
- the elementary event "the block is not formed by an honest network and the attacker did not form a block" with probability $(1-p)(1-q)$;
- the elementary event "the block is formed by an honest network and the attacker formed a block" pq.

The set of all elementary outcomes makes up a complete group of events:

$$p(1-q)+(1-p)q+(1-p)(1-q)+pq=1.$$

This model with four elementary outcomes (we will call it hereinafter "the model with independent players") describes the real probabilistic process in the blockchain system when consensus is established based on the "Proof of work" algorithm.

3 Comparison of probabilistic events in two investigated models

In the model of independent players, the formation of the next block by the attacker and the honest network occurs independently of each other, the probabilities of searching for the inverse image of the hash function (for forming the block) are determined by their hashrates (computational capabilities). For comparison with the results obtained in the works of S. Nakamoto and M. Rosenfeld (for the player's ruin model), we will use generally accepted simplifications:

- the propagation time of the block over the network is negligible, i.e. information exchange between nodes occurs almost instantly (synchronization time is zero);
- the attacker's hashrate, fair network hashrate and mining complexity does not change over time throughout the race;
- the ability of an attacker to maintain the state of the race is large enough, but not infinite;
- except for the attacker, all other network users act strictly in accordance with the rules of the blockchain network protocol;
- we consider the victory of the attacker to be the formation of the required number of confirmation blocks earlier or simultaneously (it is believed that the attacker formed one block in advance) or otherwise, the subsequent formation of a chain of blocks of equal length with an honest network.

In the double-spending task, an attacker wins if he generates an equal number of blocks with an honest network, provided that the honest network has already formed blocks. Here we use the same formulation as in the work of M. Rosenfeld [5] assuming that one block was previously mined by the attacker before the start of the attack and, therefore, the total length of the chain formed by the attacker will be one more, which is sufficient for accepting it an honest network as the main blockchain.

If we assume that the resources of the attacker are finite or the gain by the attacker does not cover his financial expenses for maintaining the further race, then it is logical to assume a restriction on the formation of the maximum number of blocks in the competition [11]. Suppose that an attacker refuses to continue the race if an honest network has formed $N+n_{\max}$ blocks. All conditions in which the attacker did not win will be losing for him.

It is necessary to pay attention to two points in the player’s ruin model:

- an attacker cannot win the race earlier than for $2\cdot N$ attempts (at least N attempts to form N blocks by an honest network and as many attempts to form N blocks by an attacker are necessary);
- an attacker can win with an odd number of attempts only if he is ahead of an honest network before it forms N blocks (the probability of which is much less with less mining power by an attacker).

In contrast to the player’s ruin model, in an independent player model, an attacker can win starting from the N attempt and since the events of the formation of blocks by both participants are independent and there is no dependence of the probability of winning on the parity or oddness of the current attempt.

Consider an example of calculating the probability of the occurrence of an event for the two models considered. For definiteness, we assume the probability of the formation of a block by an attacker for each attempt $q=0,3$ (the probability of not forming a block will be $(1-q)=0,7$). To agree with the player’s ruin model, let us put $p=0,7$ (the probability of not forming a block with an honest network will be $(1-p)=0,3$). The required number of confirmations $N=1$. Limit on the formation of the maximum number of blocks in the match $N+n_{\max}=2$ attempt.

Lets analyze the probabilities of different outcomes for different models.

Consider the player ruin model:

1. First attempt (two possible outcomes)

- formation of a block by an honest network, the attacker is one block behind, the race continues, the probability of such an event is equal $p=0,7$;
- formation of a block by an attacker, an honest network is one block behind, an attacker's[1] victory, the probability of an event being equal $q=0,3$;

2. Second attempt (four possible outcomes, we consider only the case of forming a block by an honest network in the first attempt, i.e., when the race continues)

- in the first and second attempts, a block was formed by an honest network, the attacker lost the race, the race is completed, the probability of an event $p \cdot p = 0{,}49$;
- in the first attempt, a block is formed by an honest network, but in the second attempt, the block is formed by an attacker, the attacker won, the race is completed, the probability of an event $p \cdot p = 0{,}21$.

Thus, in the player’s ruin model, the attacker will be able to defeat with probability $0,3+0,21=0,51$.

For the independent player model:

1. First attempt (four possible outcomes)

- the block is formed by an honest network and the attacker did not form the block, the attacker is one block behind, the race continues, the probability of the event occurring $p \cdot (1-q) = 0{,}49$;
- the block is not formed by an honest network and the attacker formed the block, the victory of the attacker[1], the probability of the occurrence of the event $(1-p) \cdot q = 0{,}09$;
- the block is not formed by an honest network and the attacker did not form a block, the race continues, the probability of an event $(1-p) \cdot (1-q) = 0{,}21$;
- the block is formed by an honest network and the attacker formed a block, the attacker's victory, the race is completed, the probability of the event $p \cdot q = 0{,}21$;

2. The second attempt (sixteen possible outcomes, we consider only those cases when after the first attempt the outcome of the race is not determined)

- (in the first attempt, the block is formed by an honest network and the attacker did not form a block)

[1] In this case, the victory will be counted only after the formation of the $N=1$ block by an honest network

- in the second attempt, the block is formed by an honest network and the attacker did not form the block, the attacker lost the race, the race is completed, the probability of the event $p \cdot (1-q) \cdot p \cdot (1-q) = 0{,}2401$;
- in the second attempt, the block is not formed by an honest network and the attacker formed a block, the attacker's victory, the race is completed, the probability of the event $p \cdot (1-q) \cdot (1-p) \cdot q = 0{,}0441$;
- in the second attempt, the block is not formed by an honest network and the attacker did not form a block, the race continues, the probability of the event occurring $p \cdot (1-q) \cdot (1-p) \cdot (1-q) = 0{,}1029$;
- in the second attempt, the block is formed by an honest network and the attacker formed a block, the attacker lost the race, the race is completed, the probability of an event $p \cdot (1-q) \cdot p \cdot q = 0{,}1029$;
- (in the first attempt, the block is not formed by an honest network and the attacker did not form a block)
- in the second attempt, the block is formed by an honest network and the attacker did not form a block, the race continues, the probability of the event occurring $(1-p) \cdot (1-q) \cdot p \cdot (1-q) = 0{,}1029$;
- in the second attempt, the block is not formed by an honest network and the attacker formed a block, the attacker's[2] victory, the probability of the event $(1-p) \cdot (1-q) \cdot (1-p) \cdot q = 0{,}0189$;
- in the second attempt, the block is not formed by an honest network and the attacker did not form a block, the race continues, the probability of the event occurring $(1-p) \cdot (1-q) \cdot (1-p) \cdot (1-q) = 0{,}0441$;
- in the second attempt, the block is formed by an honest network and the attacker formed a block, the attacker's victory, the race is completed, the probability of the event $(1-p) \cdot (1-q) \cdot p \cdot q = 0{,}0441$.

Thus, in the model of independent players, the attacker will be able to win in two attempts with probability $0,09 + 0,21 + 0,0441 + 0,0189 + 0,0441 = 0{,}4071$, which differs from the probability calculated for the player’s ruin model.

Using modeling, we will conduct a computational experiment and empirically evaluate the probabilities of an attacker gaining from an honest network with various models of forming a chain of blocks.

4 Simulation experiment

At the first stage, we test the probability of forming N blocks exactly for t attempts and the probability of forming N blocks during t testing.

[2] In this case, the victory will be counted only after the formation of the $N = 1$ block by an honest network

4.1 The probability of forming a chain of blocks of a given length

At the first stage, we test the probability of forming a block exactly for t attempts.

As input parameters we will use:

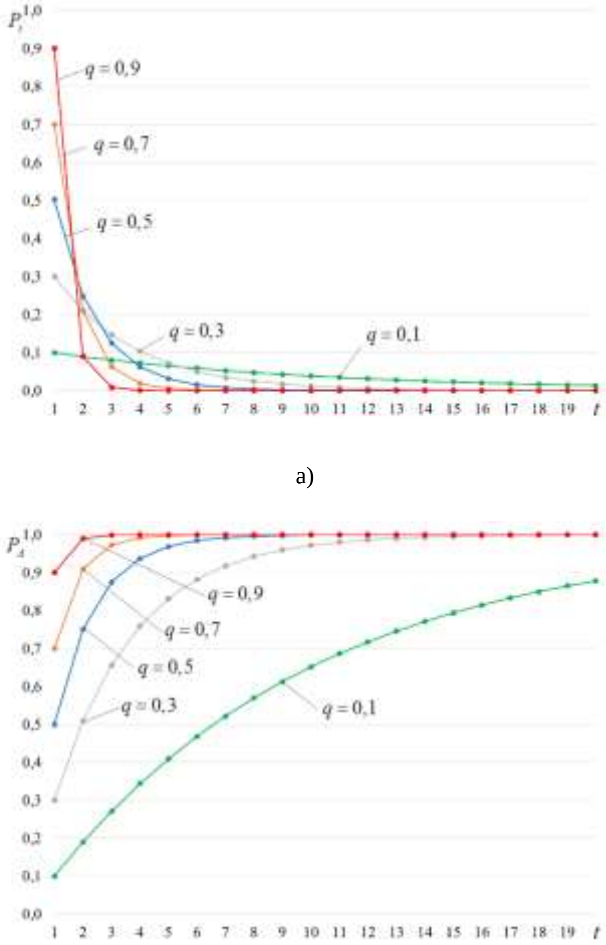
- q is the likelihood of an attacker successfully forming a block at each test. The probability depends on the computational capabilities of the attacker (i.e., it is proportional to the hashrate of the attacker);
- p is the probability of successfully forming a block by honest participants at each test (in proportion to the hash rate of an honest network). In modeling, we will assume that $p = 1 - q$, since such an interconnection underlies the player's ruin model. In the general case, for a model of independent players, the condition $p = 1 - q$ may not be satisfied;
- N is the number of blocks in the network after which the transaction is considered confirmed;
- t is current attempt number.

In a software environment, create a process that iteratively tries to form blocks. Each test takes place according to the following rule:

- generate a pseudo-random number (we use the implementation based on the Mersenne Vortex) in the interval $[0,1]$ (the software implementation uses the minimum generation step $5.4 \cdot 10^{-20}$, which allows testing $q \geq 10^{-19}$);
- compare the generated random number with the value q;
- if the generated number $\leq q$, then we believe that the block generation was successful and increase the counter of the blocks generated by the stream (*k_block1*) by one. We check *k_block1 1=N* if the necessary number of blocks is formed, then we increase the array $Mass1[t]$ by one, which corresponds to a successful attempt to form a chain of blocks of the desired length on the t -th attempt.

To achieve the specified accuracy of the test, we perform it N_{test} times (the selection N_{test} is described below). At the end of all tests, the result (array $Mass1[t]$) is normalized by the total number of tests, and thus we obtain the empirical distribution of the probability *N* of block formation from the number of attempts (P_t). Summing up all the obtained probabilities from 1 to the given t , we obtain the empirical distribution function of the probability of block formation (P_A).

The test results are shown in Figure 1 (points). The solid lines correspond to the negative binomial distribution and its distribution function for the same probabilities.

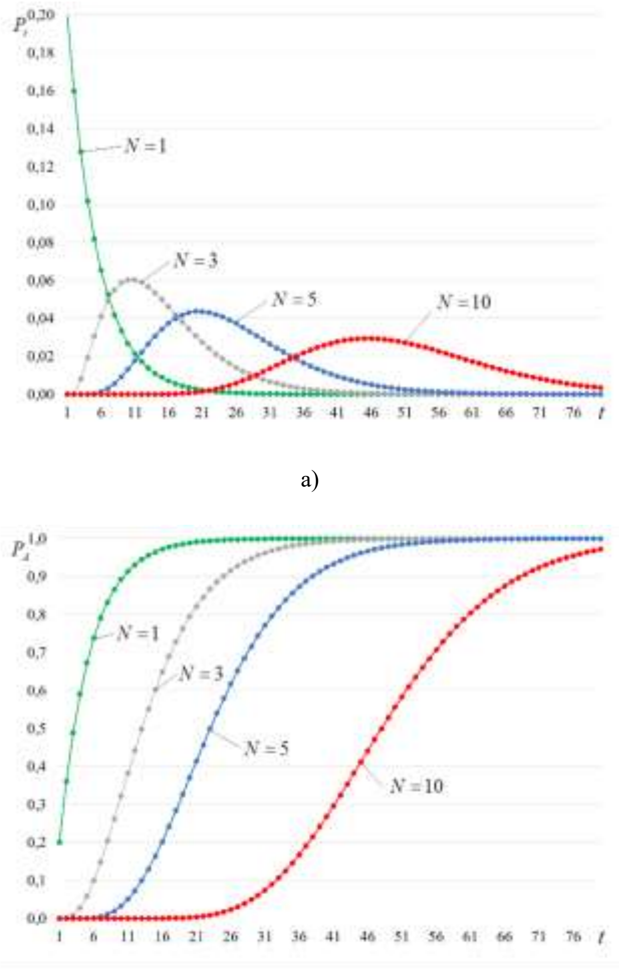


a)

b)

Fig. 1. The probability function (a) and the probability distribution function (b) of the formation of the block during each test (the lines show the calculated values corresponding to the negative binomial distribution, the points show the experimental data)

We construct the probability of forming a chain of blocks. Figure 2 presents the obtained similar probabilities, but for a fixed value and a different number.



a)

b)

Fig. 2. The probability function (a) and the probability distribution function (b) of forming a chain of N blocks at $q=0,2$ (calculated values are shown by lines, points correspond to experimental data)

As you can see, the values obtained by computational modeling are in good agreement with the negative binomial distribution.

At the second stage, we will simulate two competing participants.

We will investigate two models of competition (race) of an attacker with an honest network to form a chain of blocks:

- player ruin model;
- model of independent players.

4.2 Player Ruin Model

In a software environment, we create a process (corresponding to an attacker) that iteratively tries to form blocks. Each test takes place according to the following rule:

- generate a random number in the interval $[0,1]$;
- compare the resulting number with q;
- if the generated number $\leq q$, then we believe that the block generation was successful and increase the counter of the blocks generated by the attacker (*k_block2*) by one. Check *k_block2* $\geq N$, if yes, then check: did the attacker form the chain of the required length:
- if the attacker also managed to generate the required number of blocks (i.e. *k_block1* $\geq$ *k_block2*), then increase $Mass1[t]$ by one, which corresponds to the attacker's successful attempt to form a chain of blocks of the required length for t attempts (the attacker won the race). We complete the test;
- if the attacker has not yet generated the required number of blocks (i.e. *k_block1* < *k_block2*), then continue the test;
- if *k_block2* = $N + n_{max}$ we end the test by assigning the victory to an honest network (increase the array $Mass2[t]$ by one).

In the calculated experiments, we put $n_{\max} = 1000$ (which corresponds to the almost unlimited resources of the attacker). When choosing $n_{\max}$, we mention the work [11] which states that for the $q < 0.45$ choice, the value $n_{\max} = 35$ practically does not affect the result, in addition, this issue will be considered below.

4.3 Independent Player Model

In a software environment, we create two independent processes (the first process corresponding to the attacker, the second to honest users), iteratively trying to form blocks. Each test takes place according to the following rule:

- generate a random number in the interval $[0,1]$;
- compare the resulting number with q;
- if the generated number $\leq q$, then we believe that the block generation was successful and increase the counter of the blocks generated by the attacker (*k_block1*) by one.
- generate a random number in the interval $[0,1]$;
- compare the resulting number with p;
- if the generated number $\leq p$, then we believe that the block generation was successful and increase the counter of the blocks generated by the attacker (*k_block2*) by one. Check *k_block2* $\geq N$, if yes, then check: did the attacker form the chain of the required length:

- if the attacker also managed to generate the required number of blocks (i.e. *k_block1* ≥ *k_block2*), then increase $Mass1[t]$ by one, which corresponds to the attacker's successful attempt to form a chain of blocks of the required length for t attempts (the attacker won the race). We complete the test;
- if the attacker has not yet generated the required number of blocks then continue the test;
- if *k_block2* = $N + n_{max}$ we end the test by assigning the victory to an honest network (increase the array $Mass2[t]$ by one).

4.4 Ensuring the accuracy and reliability of simulation results

Using simulation, the exact value of a random variable (denoted by Θ) cannot be determined, since the number of model implementations is limited. With a finite number of model implementations, the approximate value of a given characteristic is determined. We denote this approximation as Θ^*. The approximate value is called the assessment of the corresponding characteristics [14-16].

The accuracy of characterization Θ^* is called the value ε relative

$$\left|\Theta^* - M\left[\Theta\right]\right| < \varepsilon ,$$

where $M\left[\Theta\right]$ is mathematical expectation of a random variable [14-16].

The value ε represents the absolute value of the error in determining the value of the desired characteristic.

The reliability of the evaluation of characteristics Θ^* is called the probability α that a given accuracy is achieved [14-16]:

$$P\left(\left|\Theta^* - M\left[\Theta\right]\right| < \varepsilon\right) = \alpha .$$

Reliability characterizes the repeatability, stability of the experiment and is interpreted as follows: if for estimation $M\left[\Theta\right]$ we use a value Θ^*, then on average for every 1000 uses of this rule in $1000 \cdot \alpha$ cases the value Θ^* will differ from the value by smaller ε.

In some cases, it is advisable to use relative accuracy

$$d = \varepsilon / M\left[\Theta\right].$$

In this case, the reliability of the assessment is:

$$P\left(\left|\frac{\Theta^* - M\left[\Theta\right]}{M\left[\Theta\right]}\right| < d\right) = \alpha .$$

If we assume the assumption regarding the normal distribution of a random variable[3], then the functional relationship between relative accuracy and reliability with the number of implementations N_{test} has the form [15]:

$$N_{test} = \frac{t_\alpha{}^2(1-P)}{Pd^2},$$

where t_α – Laplace function argument $t_\alpha = \Phi_0{}^{-1}\left(\frac{\alpha}{2}\right)$, Laplace integral tabulated, therefore, given the value of reliability α, we can determine t_α.

It follows from the last formula that in determining estimates of low probabilities with acceptable accuracy, it is necessary to perform a very large number of model implementations. In the absence of a high-performance computer, the application of statistical modeling becomes problematic.

For experimental studies were selected $\alpha = 0.99$ and $d = 0.01$, the value N_{test} was calculated according to the above formula.

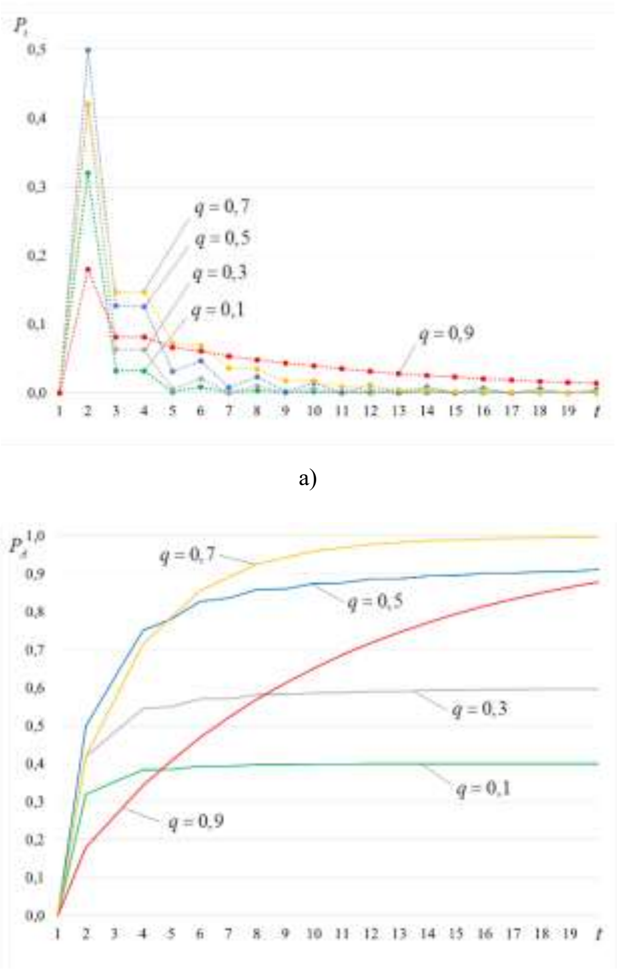
5 Calculation results

Using the considered models, empirical estimates of the probability of a successful formation of a chain of blocks by an attacker were obtained for different values of q and N. Figures 3-8 show the results obtained depending on the number of attempts for each test, and also presents the corresponding probability distribution functions depending on the number of attempts for each test.

Summing up the probabilities described above for all possible tests, that is, for all $t = 1, 2, 3, \ldots$, we get the integral (or general) probability of the successful formation of an alternative chain of blocks for confirmation by an attacker (${}^1PI$).

For the above examples, the integral probability of an attacker successfully forming a chain for N confirmations is shown in Figure 9. For the convenience of analyzing the data obtained, the same result is shown in the usual scale (well illustrates the behavior of the curves at $q > 0,2$) and the logarithmic scale (to illustrate the curves at $q < 0,2$). In these and subsequent graphs, the value q was changed in increments of 0.02.

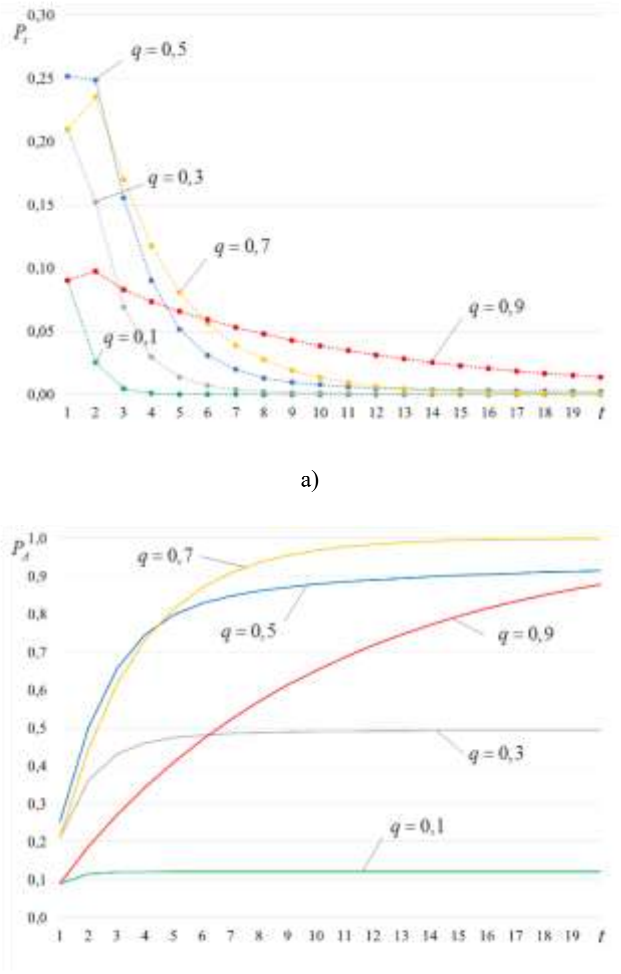
[3] By virtue of the central limit theorem, for a large number of tests, the binomial distribution is well approximated by the normal distribution [14-16]



a)

b)

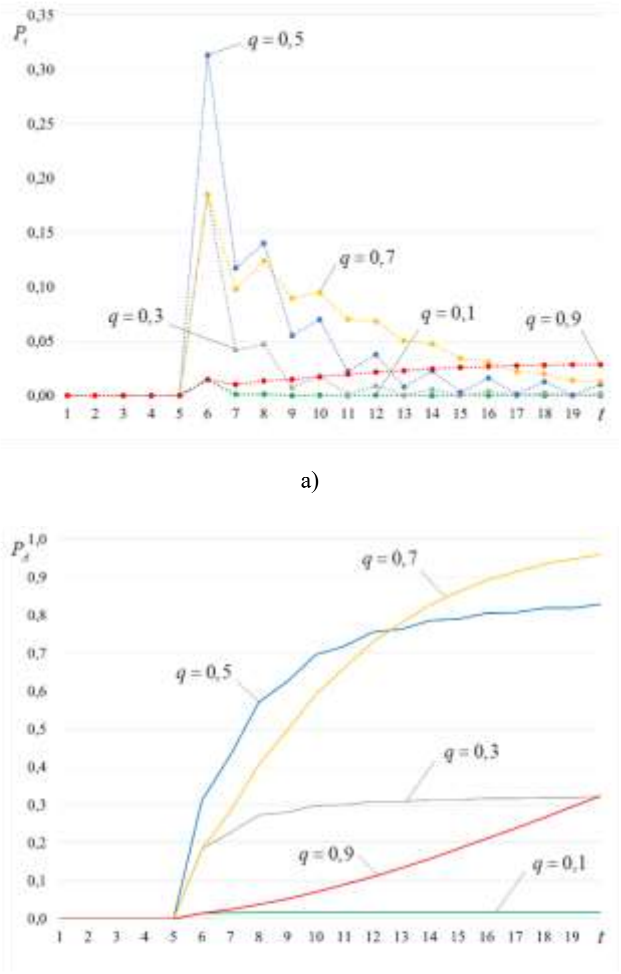
Fig. 3. The probability function (a) and the probability distribution function (b) of forming a chain for $N = 1$ confirmations by an attacker with the participation of two competing entities (player’s ruin model)



a)

b)

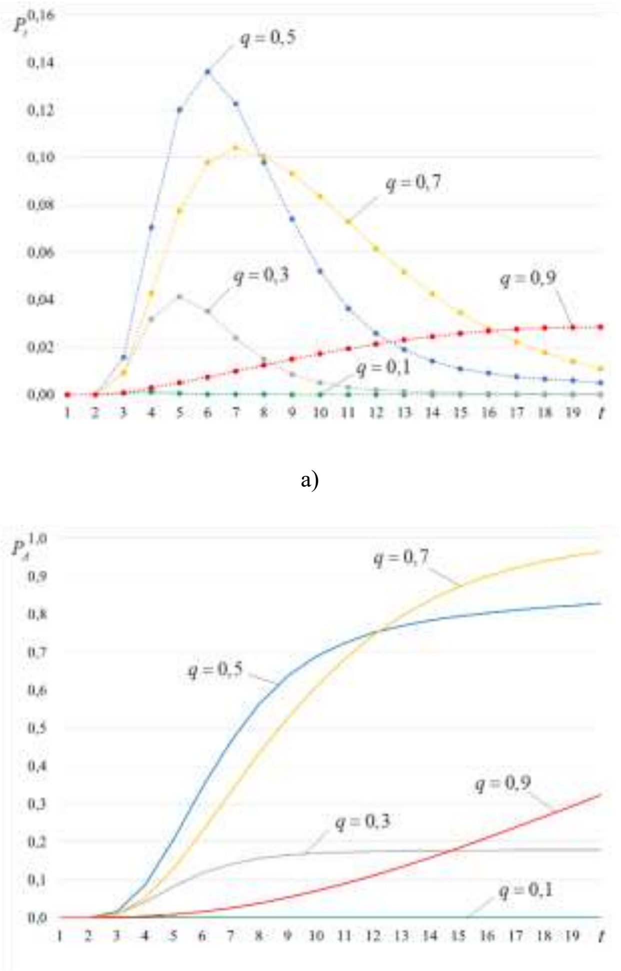
Fig. 4. The probability function (a) and the probability distribution function (b) of the formation of a chain for $N = 1$ confirmations by an attacker with the participation of two competing entities (independent players model)



a)

b)

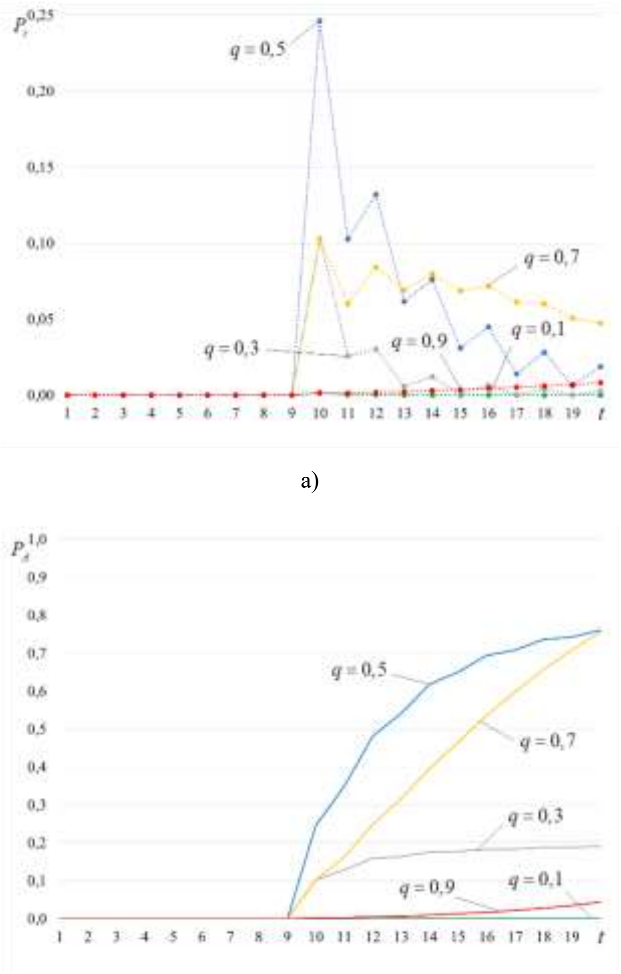
Fig. 5. The probability function (a) and the probability distribution function (b) of forming a chain for $N = 3$ confirmations by an attacker with the participation of two competing entities (player's ruin model)



a)

b)

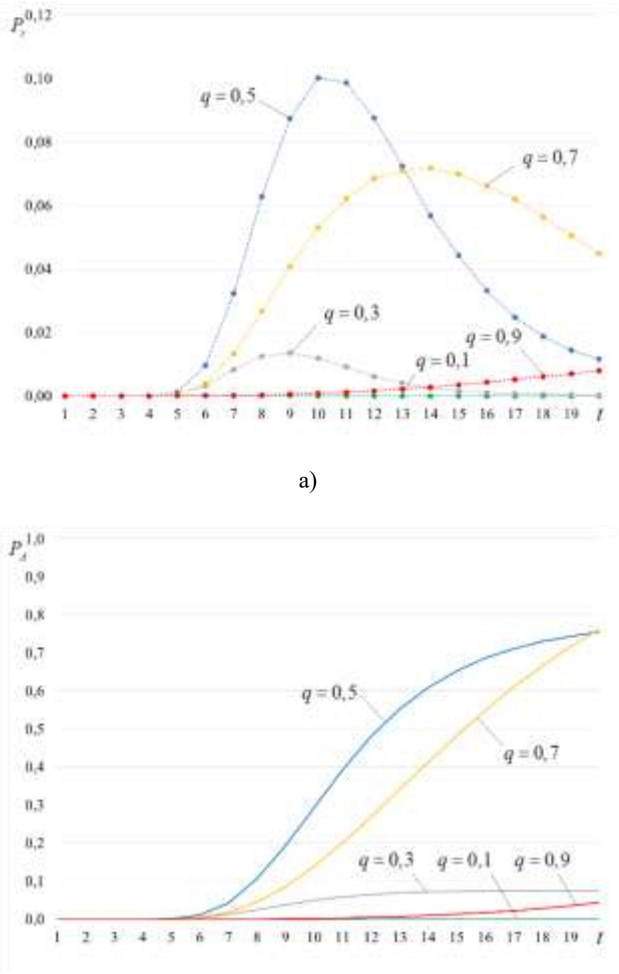
Fig. 6. The probability function (a) and the probability distribution function (b) of the formation of a chain for $N = 3$ confirmations by an attacker with the participation of two competing entities (independent players model)



a)

b)

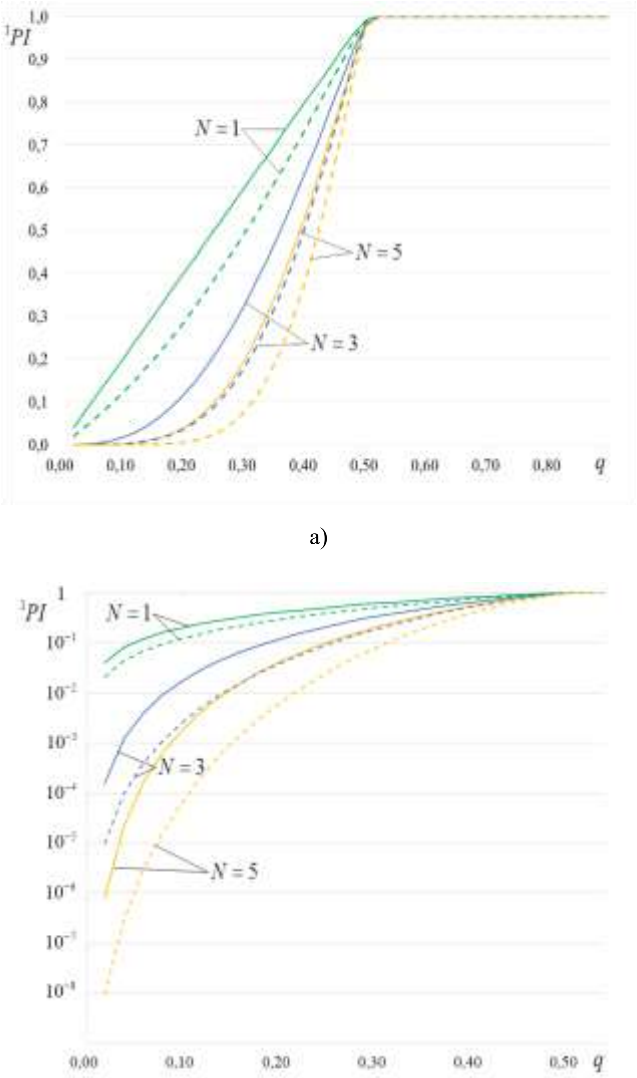
Fig. 7. The probability function (a) and the probability distribution function (b) of forming a chain for $N = 5$ confirmations by an attacker with the participation of two competing entities (player's ruin model)



a)

b)

Fig. 8. The probability function (a) and the probability distribution function (b) of the formation of a chain for $N = 5$ confirmations by an attacker with the participation of two competing entities (independent players model)



a)

b)

Fig. 9. The integral probability of an attacker successfully forming a chain of blocks for N confirmations involving two competing entities (experimental data). The solid line is the player's ruin model, the dotted line is the model of independent players: a) is the usual scale, b) is the logarithmic scale.

As you can see, the results of different models are significantly different from each other. Consider the relative modeling error (for the two models considered), defined as:

$$\frac{{}^{1}PI_{мри} - {}^{1}PI_{мни}}{{}^{1}PI_{мри}} \cdot 100\% ,$$

where ${}^{1}PI_{мри}$ is integral probability calculated on the basis of the player's ruin model; ${}^{1}PI_{мни}$ is integral probability calculated on the basis of the model of independent players,

With the designation given, the values of the relative modeling error are given in Table 1.

Table 1. The value of the relative error as a result of the application of various models (based on the ruin of the player and independent players)

	$q = 0,01$	$q = 0,2$	$q = 0,4$
$N = 1$	48%	29%	8%
$N = 3$	94%	68%	20%
$N = 5$	99%	85%	28%

As we can see from the table, the two double-spending attack models considered (the player ruin model and the independent player model) give different estimates of the probability of winning the race by the attacker (attack success). As the length of the block chain N increases, the discrepancy increases (the relative simulation error reaches 100%). This is observed for different probabilities q (i.e., for different ratios of the hash rates of the attacker and the honest network).

It should be noted that the results obtained on the basis of the player's ruin model correspond (within the limits of the given reliability and the chosen relative accuracy) to the analytical results obtained on the basis of the formulas of M. Rosenfeld (see expression 1 and Fig. 4 from [5]). The difference is observed only at the point $q = 0,5$ where the relative error between the experimental and analytical results was 1.7% for $N = 3$ and 2.2% for $N = 5$, which is associated with a limitation in $n_{\max} = 1000$ blocks.

As was shown in [11], the result has differences for different values $n_{\max}$. Let us analyze this issue in more detail.

6 Impact $n_{\max}$ on the probability of an attacker's victory

Given that the support of the race by the attacker constantly requires certain financial costs from the attacker, the race can only theoretically continue indefinitely. In real circumstances, it will not be profitable for an attacker to continue the race and spend more resources on maintaining it than he can recover by successfully conducting a double-spend attack, or he has at his disposal. Another option, if an attacker is able to form a certain number of blocks over a long period of time, then it may be more economical for him to publish them according to the rules of the network, receiving a reward for this, than trying to take advantage of dishonest (not corresponding to the rules of the network) behavior. And finally, if an attacker lags behind in a race with an honest network by a significant number of blocks, then, as shown above, his chances of winning are significantly reduced and he no longer needs to continue trying indefinitely.

For all the options considered, the value $n_{\max}$ is a finite number. Consider its effect on the probability of an attacker's victory.

As an illustration, Fig. 10 shows graphs of experimental values obtained in accordance with the player’s ruin model for $n_{\max} = 10,\ 35,\ 100,\ 1000$ and different $N = 1$ and 5, as well as comparisons with theoretical results obtained by M. Rosenfeld.

As can be seen from the above results, the increase $n_{\max}$ brings the obtained empirical data closer to the analytical results of M. Rosenfeld [5]. With a decrease in probability, the theoretical results obtained by M. Rosenfeld are well approximated for small $n_{\max}$.

The relative error between theoretical and experimental results is close to the value $q = 0{,}5$ and is more than 0.1% in the following ranges:

for $N = 1$:

- from $0{,}28 \le q \le 0{,}72$ at $n_{\max} = 10$;
- from $0{,}4 \le q \le 0{,}6$ at $n_{\max} = 35$;
- from $0{,}44 \le q \le 0{,}56$ at $n_{\max} = 100$;
- $q = 0{,}5$ at $n_{\max} = 1000$;

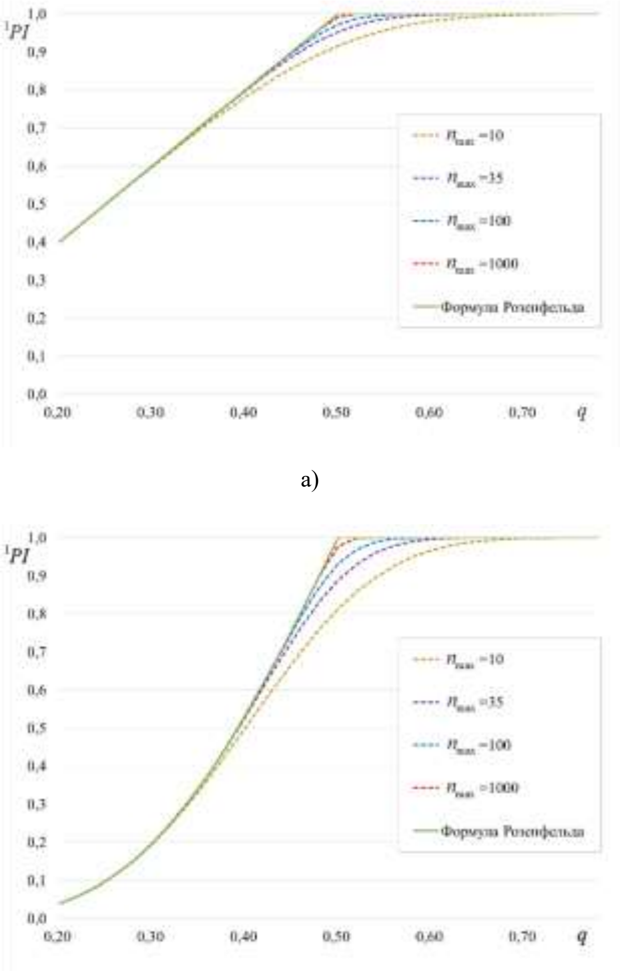
for $N = 5$:

- from $0{,}26 \le q \le 0{,}70$ at $n_{\max} = 10$;
- from $0{,}38 \le q \le 0{,}62$ at $n_{\max} = 35$;
- from $0{,}42 \le q \le 0{,}54$ at $n_{\max} = 100$;
- from $0{,}48 \le q \le 0{,}5$ at $n_{\max} = 1000$;

The convergence of experimental results with theoretical calculations based on known analytical expressions confirms the adequacy and validity of the research results.

Figures 11 show the experimental results obtained in accordance with the model of independent players with the same parameters ($n_{\max} = 10,\ 35,\ 100,\ 1000$; $N = 1,\ 5$). For clarity, the theoretical result obtained by M. Rosenfeld is left.

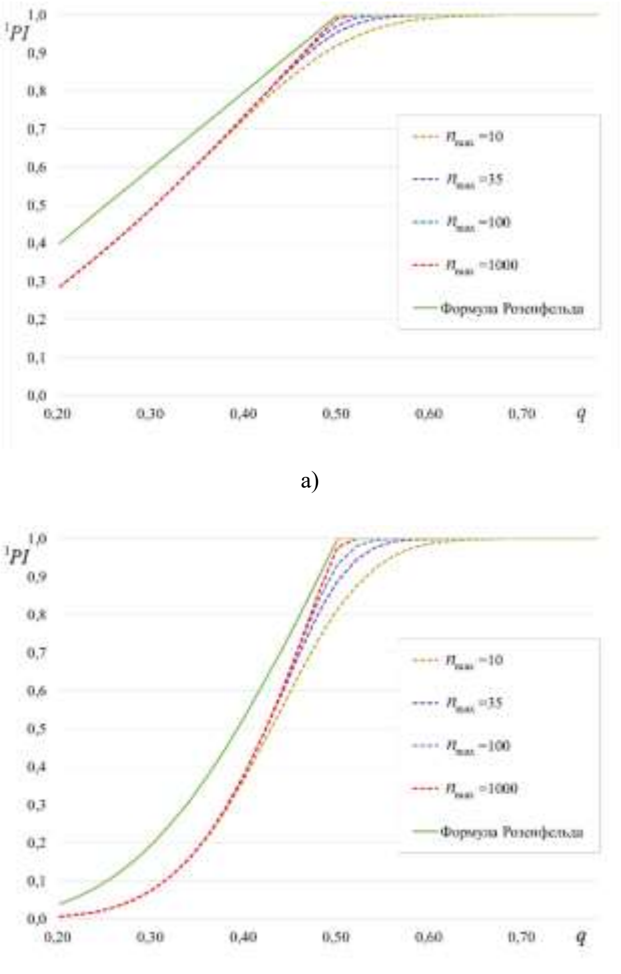
As you can see from the above graphs, the nature of the effect $n_{\max}$ on the result is expectedly preserved for the model of independent players. However, a comparison of the results obtained for different models confirms the thesis about the discrepancy between the estimates of the probabilities of a successful double-spend attack.



a)

b)

Fig. 10. The integral probability of an attacker successfully forming a chain of blocks for $N = 1$ (a) and $N = 5$ (b) confirmations with the participation of two competing entities (dotted line is experimental data). Player ruin model.



a)

b)

Fig. 11. The integral probability of an attacker successfully forming a chain of blocks for $N = 1$ (a) and $N = 5$ (b) confirmations involving two competing entities (dashed line — experimental data). Model of independent players.

7 Conclusions

In this paper, a critical analysis of the well-known works on estimating the probabilities of double spending in the protocol of consensus “Proof of work” is carried out. The presence of inaccuracies and unreasonable assumptions in the well-known works of S. Nakamoto [4] and M. Rosenfeld [5] is shown. In particular, it is shown that the basic assumptions about the probability space (the set of elementary outcomes and the probability of their occurrence) in the used model of player ruin (with two elementary outcomes) do not correspond to the real processes that occur during the establishment of the “Proof of work” consensus.

It is proposed to use a model of independent players with four elementary outcomes for a theoretical assessment of the probability of a successful double-spend attack. This model describes the real probabilistic process in the blockchain system when consensus is established based on the Proof of work algorithm, when each participant (an attacker and an honest network) independently form blocks with probabilities proportional to their hashrate (their computing capabilities).

A comparison of the results obtained using computational modeling of a double-spending attack based on the player’s ruin model and the model of independent players is carried out. The comparison was made for different capabilities of the attacker (the probability of forming a block), a different number of formed blocks after which the transaction is considered confirmed, of a different duration of the race (the number of blocks during which the attacker continues to try to catch up with an honest network). A significant difference (relative error of the model up to 99%) of the results obtained in computational modeling when using the independent players model from the player’s bankruptcy model is shown.

All empirical estimates were obtained for high accuracy (relative error of no more than 1%) and reliability (confidence level of at least 99%).

To confirm the adequacy of the results obtained, a comparison of empirical results with theoretical calculations based on known analytical relationships is given. It is shown that the results of a computational experiment for the player’s ruin model completely coincide (within the limits of a given reliability and relative accuracy) with the analytical result given in the work of M. Rosenfeld [5].

Based on the results obtained, it is possible to argue about the fallacy of using the player’s ruin model to assess the likelihood of a successful double spending attack on the “Proof of work” consensus protocol. These results can be useful in improving various mechanisms of cryptographic protection [17-21], especially in the context of building decentralized systems using blockchain technology. These results can be used in other computer science applications [22-28].

References

1. The Double Spending Problem and Cryptocurrencies. Banking & Insurance Journal. Social Science Research Network (SSRN). Accessed 24 December 2017. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3090174
2. Mark Ryan. “Digital Cash”. University of Birmingham. Retrieved 2017-05-27. https://www.cs.bham.ac.uk/~mdr/teaching/modules06/netsec/lectures/DigitalCash.html
3. Varshney, Neer. “Why Proof-of-work isn't suitable for small cryptocurrencies”. Hard Fork. Retrieved 2018-05-25. https://thenextweb.com/hardfork/2018/05/24/proof-work-51-percent-attacks/
4. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System, 2009, 9 p.
5. Rosenfeld M. Analysis of hashrate-based double-spending / Meni Rosenfeld., 2014, 13 p.

6. Carlos Pinzón, Camilo Rocha. Double-spend Attack Models with Time Advantange for Bitcoin. Electronic Notes in Theoretical Computer Science. Volume 329, 9 December 2016, pp. 79-103 https://doi.org/10.1016/j.entcs.2016.12.006
7. Kaidalov D.S., Kovalchuk L.V., Nastenko A.O., Rodinko M.Yu., Shevtsov O.V., Oliynykov R.V. Comparison of block expectation time for various consensus algorithms. Radio Electronics, Computer Science, Control. 2018. № 4. pp. 159- 171 DOI 10.15588/1607-3274-2018-4-15
8. Azzolini D., Riguzzi F., Lamma E., Bellodi E., Zese R. Modeling Bitcoin Protocols with Probabilistic Logic Programming http://ceur-ws.org/Vol-2219/paper6.pdf
9. Kevin Liao, Jonathan Katz. Incentivizing Double-Spend Collusion in Bitcoin. 2017. https://www.cs.umd.edu/~gasarch/reupapers/katzbitcoin16.pdf
10. Kovalchuk L.V. The main signpost for the galactic blockchain and a detailed analysis of the results of the Nakamoto-Rosenfeld-Grunspan about the imminent attack of the vitriot. Zvit about the NDR (industrial), Kharkiv, AT IIT, 36 p.
11. Pinar Ozisik., Brian Neil Levine. An Explanation of Nakamoto's Analysis of Double-spend Attacks https://arxiv.org/pdf/1701.03977.pdf
12. Apostolaki M. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies / M. Apostolaki, A. Zohar, L. Vanbever. – San Jose, CA , USA, 2017, 18 p.
13. Grunspan C., Pérez-Marco R. Double spend races. 2017. hal-01456773 https://hal.archives-ouvertes.fr/hal-01456773.
14. W. Feller. An Introduction to Probability Theory and its Applications: Volume I, volume 3. John Wiley & Sons London-New York-Sydney-Toronto, 1968.
15. Smirnov N.V., Dunin-Barkovsky I.V. A course in probability theory and mathematical statistics for technical applications. M., "Science", 1969, 512 p.
16. A. N. Shiryaev, “Probability,” Graduate Texts in Mathematics, 1996.
17. Andrushkevych A., Gorbenko Y., Kuznetsov O., Oliynykov R., Rodinko M. A (2019) “A Prospective Lightweight Block Cipher for Green IT Engineering”. In: Kharchenko V., Kondratenko Y., Kacprzyk J. (eds) Green IT Engineering: Social, Business and Industrial Applications. Studies in Systems, Decision and Control, vol 171. Springer, Cham, pp. 95-112. DOI: 10.1007/978-3-030-00253-4_5,
18. Kuznetsov O., Potii O., Perepelitsyn A., Ivanenko D., Poluyanenko N. (2019) “Lightweight Stream Ciphers for Green IT Engineering”. In: Kharchenko V., Kondratenko Y., Kacprzyk J. (eds) Green IT Engineering: Social, Business and Industrial Applications. Studies in Systems, Decision and Control, vol 171. Springer, Cham, pp. 113-137. DOI: 10.1007/978-3-030-00253-4_6,
19. Krasnobayev V., Kuznetsov A., Koshman S., Moroz S. (2019) Improved Method of Determining the Alternative Set of Numbers in Residue Number System. In: Chertov O., Mylovanov T., Kondratenko Y., Kacprzyk J., Kreinovich V., Stefanuk V. (eds) Recent Developments in Data Science and Intelligent Analysis of Information. ICDSIAI 2018. Advances in Intelligent Systems and Computing, vol 836. Springer, Cham, pp. 319-328, 05 August 2018. DOI: 10.1007/978-3-319-97885-7_31.
20. A. Kuznetsov, V. Frolenko, E. Eremin and O. Zavgorodnia, "Research of cross-platform stream symmetric ciphers implementation," *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Kyiv, Ukraine, 2018, pp. 300-305. DOI: 10.1109/DESSERT.2018.8409148.
21. I. Gorbenko, A. Kuznetsov, V. Tymchenko, Y. Gorbenko and O. Kachko, "Experimental Studies Of The Modern Symmetric Stream Ciphers," *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkiv, Ukraine, 2018, pp. 125-128. doi: 10.1109/INFOCOMMST.2018.8632058.

22. Bondarenko, S., Liliya, B., Oksana, K., & Inna, G. (2019). Modelling instruments in risk management. International Journal of Civil Engineering and Technology, 10(1), 1561-1568.
23. S. Gnatyuk, V. Kinzeryavyy, M. Iavich, D. Prysiazhnyi, Kh. Yubuzova, High-Performance Reliable Block Encryption Algorithms Secured against Linear and Differential Cryptanalytic Attacks, CEUR Workshop Proceedings, Vol. 2104, pp. 657-668, 2018.
24. Runovski, K., & Schmeisser, H. -. (2004). On the convergence of fourier means and interpolation means. Journal of Computational Analysis and Applications, 6(3), 211-227.
25. S. Gnatyuk, A. Okhrimenko, M. Kovtun, T. Gancarczyk, V. Karpinskyi, Method of Algorithm Building for Modular Reducing by Irreducible Polynomial, Proceedings of the 16th International Conference on Control, Automation and Systems, Oct. 16-19, Gyeongju, Korea, 2016, pp. 1476-1479.
26. Tkach, B. P., & Urmancheva, L. B. (2009). Numerical-analytic method for finding solutions of systems with distributed parameters and integral condition. Nonlinear Oscillations, 12(1), 113-122. doi:10.1007/s11072-009-0064-6.
27. Gnatyuk S., Kinzeryavyy V., Kyrychenko K., Yubuzova Kh., Aleksander M., Odarchenko R. Secure Hash Function Constructing for Future Communication Systems and Networks, Advances in Intelligent Systems and Computing, Vol. 902, pp. 561-569, 2020.
28. Chornei, R., Hans Daduna, V. M., & Knopov, P. (2005). Controlled markov fields with finite state space on graphs. Stochastic Models, 21(4), 847-874. doi:10.1080/15326340500294520.