Publ. Math. Debrecen
Supplementum **100** (2022), 597–615
DOI: 10.5486/PMD.2022.Suppl.4

Probability of double spend attack for network with non-zero time delay

By Lyudmila Kovalchuk, Mariia Rodinko, Roman Oliynykov, Dmytro Kaidalov and Andrii Nastenko

Abstract. The paper presents the formulas for probability of a double spend attack on blockchain with Proof-of-Work consensus protocol for a network with a non-zero synchronization time. The results show that probability of a double spend attack depends essentially on the block delivery time and intensity of block generation. More precisely, the probability of such attack increases when the product of these two values increases. The analytical results obtained in this paper allow not only to calculate the exact value of attack probability, but also to define the minimal number of confirmation blocks sufficient to guarantee the security against this attack with an arbitrary preset probability value.

1. Introduction

The idea of the double spend attack appeared at the same time when the idea of the blockchain itself – for the first time this attack was described in the paper by S. Nakamoto [6]. Since that time, it was analyzed in multiple papers, and also was mentioned in the Princeton University course"Blockchain and Cryptocurrency" (Coursera). Its essence lies in the fact that an attacker tries to use his coin at least twice.

Technically, it happens as follows. An adversary carries out some transaction in the, say, block #5, transferring money to a supplier of goods for some purchase. The supplier receives that amount of money, and accordingly supplies the goods

Mathematics Subject Classification: 68P20,60-08.
Key words and phrases: blockchain, double spend attack, Proof-of-Work, consensus protocol, Bitcoin.

to the buyer. Having received the goods (or maybe earlier), the adversary quickly starts mining of a different block with the same number 5, that is a block following the block #4, but one that either does not contain this transaction, or he transfers the money to himself to some other address. And to guarantee acceptance of this alternative chain by honest miners, he tries to "hook" as many blocks as possible after the alternative block numbered 5. If he succeeds in making the alternative chain longer, then exactly this chain, according to the mining protocol, will be the one to be considered correct. Obviously, the larger the ratio of the adversary (it is not essential whether is it computational power in the case of Proof-of-Work (PoW), or a stake in the case of Proof-of-Stake (PoS)), the higher the chance of his attack to be successful. In particular, if the share of the adversary exceeds 1/2, then the probability of the attack success is equal to 1.

To ensure protection against this attack, Nakamoto proposed not to supply the goods as soon as the transaction occurred, but to wait for some time, more precisely – for a certain number of blocks after this transaction, and only then, if the transaction has not disappeared from the blockchain, to supply the goods. In this case, the adversary cannot build a visible fork immediately after the payment, as then the vendor will see the fork and will not send goods. For this reason, the adversary first waits until the branch with the transaction "grows" by the required number of confirmation blocks. During this waiting period, he can invisibly generate a fork that starts before the block with the transaction, that is, in our notations, may generate an alternative fifth block with the blocks to follow, but in no case he shares this alternative chain during the confirmation period, so that the supplier will not suspect anything bad. This is the first stage of the attack. But when the confirmation blocks are formed and the goods are received, the adversary tries to "catch up" with the existing chain, and this is the second stage of the attack. Suppose that while 6 confirmation blocks are being generated, the adversary was able to generate 4 blocks of the alternative chain. And now he lags behind by at least 2 blocks. If ever in the future he is able to generate as many blocks as it is needed to "catch up" with the existing chain, which, in turn, will also grow all the time, then the attack will be successful. In particular, if he managed to generate 7 or more blocks at the first stage of the attack while he waited for the confirmation blocks, then the attack becomes already successful, there is nothing to catch up. Having received the goods, he simply shares his own longer chain, in which the money remains with him.

Probability of the attack success was also calculated in [6], depending on the network parameters and the number of confirmation blocks. Unfortunately, these calculations were made with serious probabilistic mistakes, one of which was

replacement of a random variable by its mathematical expectation. As a result of this and other mistakes, the attack success probability appeared to be significantly underestimated.

After publication of [6], a lot of other papers on estimation of double spend attack probability appeared. But some important issues in this area are still unresolved. In particular, most results were obtained using various simplified models and assumptions, such as a model with zero synchronization time (i.e., zero block delivery time) and under a simplified assumption on discrete time (more detailed analysis of these papers will be given in the following section).

In this paper, we obtained for the first time mathematically substantiated formulas for probability of a double spend attack on blockchain that is based upon Proof-of-Work consensus protocol and the longest chain rule, for a network with a non-zero time of block propagation and in the model with continuous time. In other words, we get rid of two simplifying assumptions, common for the previous papers, and get the results in a model much closer to reality. Also, for the first time, it was shown that probability of such attack depends on the value equal to the product of the block propagation time and of the block generation intensity. The larger is this value, the larger is the attack success probability. The formulas obtained allow not only calculating of the attack success probability for various network parameters, but also to determine the number of confirmation blocks allowing reduction of the attack probability below some given small threshold, e.g., 10^{-3}.

2. Related work

As it was already mentioned, the first results on the double spend attack probability for PoW protocol were published by Nakamoto [6]. However, they were obtained under assumptions that do not quite correspond to the real model. The *first* assumption that is also present in almost all other papers is the assumption that the time of the block generation and the time of its appearance in the network coincide, so the block propagation delay is zero. But from this assumption it follows that the probability of an "accidental" fork is zero, and reality shows that such forks happen about 6 times per month. The *second* assumption is even more incorrect: for simplicity, some random variable, described the block generation by honest miners, is replaced by its mathematical expectation.

Under such simplifying assumptions, analytical expressions obtained for the probability of an attack success are not correct, and the probability turns out to be underestimated that later was pointed out by some authors.

In the paper by Rosenfeld [8], other, and, as it turned out, more accurate analytical expressions for these probabilities were proposed, while a slightly different model was chosen for their production than the one used by Nakamoto. However, this paper did not provide any substantiation for their chosen model. The authors simply assumed that the appearance of "honest"/"dishonest" blocks in the network is described by a negative binomial distribution; though, this assumption was not substantiated there. In [8], the results were also obtained under the assumption that the propagation time of the block in the network is zero. Regarding the Nakamoto's second assumption, it is unclear how far the authors have noticed this fallacy; however, they did not use this simplifying assumption. For this reason, the numerical results in this paper differ from the results by Nakamoto, i.e., for the same probability of attack, Rosenfeld's paper requires more confirmation blocks that is natural.

Pinzon's paper [7] first drew attention to the incorrectness of the Nakamoto's second assumption. More precisely, it said that success of the attacker at the first stage of the attack will substantially depend on the time it took to generate confirmation blocks. There was a comment regarding the results by Rosenfeld that they were not quite accurate, and that some (unclear) function for calculation of the probability of success at the first stage was proposed. But the authors did not provide any numerical results obtained by their formulas.

At the end of the paper, it was said that the results of the papers [6], [8], [7] are approximately the same, since the formulas for the probability of the number of blocks at the first stage of attack are approximately equal. However, calculations show that the numerical results of Nakamoto and Rosenfeld differ significantly (see Table 1 in [4]). It should be also noted that the first assumption on instantaneous propagation of blocks in the network is also presented in the Pinzon's paper.

Wonderful from the mathematical point of view, Grunspan's paper [3] impresses with the mathematical rigor of presentation and substantiation. In this paper, the authors prove what Rosenfeld suggested without proof – that the process of generating "honest"/"dishonest" blocks in the network is described by a negative binomial distribution. It was first proved in this paper, using special functions, that the fork probability decreases exponentially with growth of its length. However, the authors could not, and even did not try to get rid of the same assumption on the instantaneous propagation of the block in the network.

The issue of the need to take into account timing of synchronization for the first time was voiced in the paper by Zohar, Sompolinsky [9], but the authors did not go beyond this statement.

Further, in the *paper* [4] we managed to obtain analytical expressions for attack probability with account of network synchronization time. The model was taken from the paper [1]. It assumed that the synchronization time is non-zero, but equal for honest miners and for malicious miners. The expressions obtained in [4] are much more cumbersome than those in the papers [6], [8], [7], [3], but they can be (and were) used to obtain numerical values. The numerical results of papers [8], [7], [3] and [4] are rather similar. Based on these numerical results, we can assume that synchronization time has a significant effect on the stability of a blockchain only in the case when it is essentially different for honest miners and for malicious miners.

Two recent papers [2], [5] are also devoted to the issues related to security against double spend attacks. They provide estimates for security threshold of network – the minimal ratio of adversaries that can implement the attack with probability 1 despite the number of confirmation blocks. Results of both of these papers were obtained taking into account network synchronization time. The main difference is in models in which the relevant results were obtained. The paper [2] presents an estimate of the security threshold for the Bitcoin protocol in the model with discrete time, while the paper [5] works with more realistic model, in which time is continuous. The paper [5] was the first one to state how exactly the block propagation time affects security threshold of the consensus protocol against the double spend attack. In particular, one of results of [5] is analytical expressions for calculation of the security threshold for various network parameters showing that the larger the block propagation time in the network, the larger the security threshold differs (downward) from 50%.

This paper is a logical continuation of the paper [5] and essentially uses its results. We obtained rigorously substantiated analytical expressions for double spend attack probability that allowed not only explicit calculation of such probability, but also calculation of the number of confirmation blocks which is sufficient to guarantee security against such attack with arbitrary high preset probability. Using these analytical expressions for attack probability, we obtained the relevant numerical results that also appeared to be valuable and confirmed consistence of models, assumptions, and considerations in this paper.

3. Our results

In this paper we give a comprehensive answer to the question formulated with the most common assumptions on the network parameter values. The model in which our results were obtained has the following characteristics:

- time is a continuous parameter;
- synchronization time between honest miners is set to a given arbitrary value;
- synchronization time of an adversary is zero (we make this assumption in favor of an adversary);
- block generation rate is set to an arbitrary value (both for honest miners and for the adversary);
- the fraction of adversarial hash power is arbitrary.

We adduce strictly proved expressions for the probability of double spend attack depending on the following parameters: block generation intensity, network synchronization time, adversary's ratio, and the number of confirmation blocks at the same time; depending on the network parameters, the probability of such attack may be equal to 1 even in the case when this ratio for the adversary is essentially smaller than 50%.

Using the results obtained, we also give the answer to another question: what is the minimal number of confirmation blocks which prevents the attack with some given probability close to 1?

For all analytically obtained results, we will provide numerical examples and the relevant graphs.

4. Main assumptions, notations and results used

In this paragraph, we describe the basic assumptions of our model and introduce main notations.

Sometimes we will also refer to some statements proved in [3] and [5].

We will use HMs for "Honest Miners" and MMs for "Malicious Miners". Let us define the following random variables (RVs):

T_H is the RV that measures the time it takes to mine a block for HMs,

T'_H is the RV that measures the time it takes to mine and share the block for all HMs,

T_M is the RV that measures the time it takes to mine a block for MMs,

T'_M is the RV that measures the time it takes to mine and share the block for all MMs.

As it was shown in [3], RVs T_M and T_H have exponential distributions:

$$F_{T_H}(t) = P(T_H < t) = 1 - e^{\alpha_H t}, \qquad F_{T_M}(t) = P(T_M < t) = 1 - e^{\alpha_M t}, \quad (1)$$

for some $\alpha_H > 0$, $\alpha_M > 0$. The physical meaning of these two parameters is that $\frac{\alpha_H}{\alpha}$ and $\frac{\alpha_M}{\alpha}$ are the ratios of HMs and MMs, respectively, where $\alpha = \alpha_H + \alpha_M$ – block generation intensity.

We will also assume that D_H denotes the time it takes for HMs to share a block (after it was generated) for all nodes in the network (at least for all honest nodes). The value D_M is the corresponding time for MMs. In this paper, we will assume $D_M = 0$, i.e., MMs are well-synchronized and act as one person. It also means that $T'_M = T_M$ and $F_{T'_M}(x) = F_{T_M}(x)$.

We should also note that for the sake of simplicity we assume that the block delay time D_H is the same for all HMs. Such assumption is made in favor of an adversary. Of course, this is some kind of restriction for a real model, but in the alternative case it is impossible to take into account all pairwise delays. On the other hand, we can consider D_H as the largest time delay in the network (for HMs). In these notations we have

$$T'_H = D_H + T_H, T'_M = T_M. \quad (2)$$

Let's designate p_H as the probability that HMs generate the next block before MMs (i.e., faster than MMs), and $p_M = 1 - p_H$ as the probability that HMs generate the next block before MMs. According to [3],

$$p_H = \frac{\alpha_H}{\alpha_H + \alpha_M}, p_M = \frac{\alpha_M}{\alpha_H + \alpha_M}. \quad (3)$$

We will call p_H (p_M) "the share of total hashrate that HMs (MMs) have", according to the nature of these values. But, actually, we are interested in other values that take into account the time delay $D_H > 0$. These values were first introduced in [5] and play an important role for our results:

p'_H is the probability that HMs will generate and share the next block for all (at least for all honest) nodes before MMs will generate their next block,

p'_M is the probability of the alternative event, $p'_M = 1 - p'_H$.

Then

$$p'_H = P(T'_H < T_M), p'_M = P(T_M \le T'_H), \quad (4)$$

and also $p'_H + p'_M = 1$.

These two values (4) are of much more importance than the values (3), because they take into account the time delay D_H and describe the state of the

network much more realistically. In what follows, we will show that the probability of a successful attack depends on these very values (4) rather than on the values (3). Thus, if D_H is rather large, the "real" hashrate p'_H of HMs is essentially smaller than p_H.

In what follows, we will use the following statement taken from [5]:

Lemma 4.1. *In our notations* [5]

$$p'_M = 1 - e^{-\alpha_M D_H} \cdot \frac{\alpha_H}{\alpha_M + \alpha_H} = 1 - e^{-\alpha_M D_H} \cdot p_H;$$
$$p'_H = e^{-\alpha_M D_H} \cdot \frac{\alpha_H}{\alpha_M + \alpha_H} = e^{-\alpha_M D_H} \cdot p_H.$$

Note 1. The Lemma 4.1 is of a great importance because it demonstrates that the inequality

$$p'_M > p'_H \tag{5}$$

is equivalent to the situation when MMs have a real majority and can carry out a "50% attack" even if $p_H > p_M$, as under the condition (5) the "malicious" chain will grow quicker.

In other words, when D_H is sufficiently large, then a so-called "50% attack" may take place even if HMs have prevailing computational power. More precisely, as it was shown in [5], a necessary and sufficient condition for double spend attack succeeds with probability 1 is the condition (5) instead of $p_M > p_H$. We also can rewrite (5) in the form of

$$1 - e^{-\alpha_M D_H} \cdot p_H > e^{-\alpha_M D_H} p_H,$$

that is equivalent to the inequality

$$D_H > \frac{\ln(2p_H)}{\alpha_M}, \tag{6}$$

that also may be considered as a necessary and sufficient condition for an attack with probability 1. Given p_H and α_M, the formula (6) defines the threshold time delay D_H for the network (i.e., the time delay, for which it becomes fully vulnerable to a double spend attack).

5. Main results

In this section we formulate our main results after some auxiliary lemmas. Let us designate as $T'_H(i)$ the time needed for HMs to form and share the $i-th$

block, i.e., the time from the event "$i-1$-th block is formed and available for all nodes " till the event "i-th block is formed and available for all nodes". Similarly to (2), we can also say that

$$T'_H(i) = T_H(i) + D_H, \tag{7}$$

where $T_H(i)$ is the time needed for HMs to generate the i-th block (after the $i-1$-th block becames available). Then $T'_H(i)$, $i \geq 1$ are independent identically distributed RVs with distribution functions.
$F_{T'_H(i)}(t) = F_{T'_H}(t) = F_{T_H}(t - D_H) = 1 - e^{-\alpha_H(t-D_H)}$, for all $i \geq 1$, where the latter equality follows from (1).
Also let us define RVs $T_M(i)$, $i \geq 1$ in the same way. Then their distribution functions are

$$F_{T_M(i)}(t) = 1 - e^{-\alpha_M t}, \text{for all } i \geq 1.$$

Also, for $n \geq 1$ let us define RVs $S_H(n)$, where

$$S_H(n) = \sum_{i=1}^{n} T_H(i) \tag{8}$$

and RVs $S'_H(n)$, where

$$S'_H(n) = \sum_{i=1}^{n} T'_H(i). \tag{9}$$

Then $S_H(n)$ is the time needed to generate (without sharing) n (independent) blocks, and $S'_H(n)$ is the time needed for HMs to generate and share n blocks, one after another.
From (7) we obtain that

$$S'_H = S_H(n) + nD_H,$$

where $S_H(n)$ has the Erlang distribution as the sum of independent identically distributed RVs with exponential distributions:

$$F_{S_H(n)}(t) = P(S_H(n) \leq t) = 1 - e^{-\alpha_H t} \sum_{i=1}^{n} \frac{(\alpha_H t)^k}{k!}.$$

Also let us define RVs $S_M(n)$ in the same way:

$$S_M(n) = \sum_{i=1}^{n} T_M(i). \tag{10}$$

Note that $S_M(n)$ also has the Erlang distribution:

$$F_{S_M(n)}(t) = 1 - e^{-\alpha_M t} \sum_{i=0}^{n-1} \frac{(\alpha_M t)^k}{k!}. \tag{11}$$

Let us also define RV $N_M(t)$ as the number of blocks mined by MMs till the moment t.

Lemma 5.1. *The RV $N_M(t)$ has the Poison distribution with the parameter α_M:*

$$P(N_M(t) = n) = \frac{(\alpha_M t)^n e^{-\alpha_M t}}{n!}.$$

Proof. The event $\{N_M(t) = n\}$ is the same as the event $\{S_M(n) < t\} \cap \{S_M(n+1) > t\}$, where $S_M(n)$ was defined in (10). So we can write the following chain of equalities:

$$\begin{aligned} \{N_M(t) = n\} &= \{S_M(n) < t \cap S_M(n+1) > t\} \\ &= \{S_M(n) < t \ \cap \overline{S_M(n+1) < t\}} \\ &= \{S_M(n) < t\} \backslash \{S_M(n+1) < t\}. \end{aligned}$$

But according to the definition (10), $\{S_M(n+1) < t\} \subset \{S_M(n) < t\}$, then, using (11),

$$\begin{aligned} P\{N_M(t) = n\} &= P\{S_M(n) < t\} - P\{S_M(n+1) < t\} \\ &= F_{S_M(n)}(t) - F_{S_M(n+1)}(t) = \frac{(\alpha_M t)^n e^{-\alpha_M t}}{n!}. \end{aligned}$$

The lemma is proved. □

Note 2. From the properties of the Poison process (independent increments, absence of aftereffects), we get that for any $t_1, t_2 > 0$: the distribution law of $N_M(t_2)$ is the same as the distribution law of

$$N_M(t_1 + t_2) - N_M(t_1).$$

Note that the number of events happening during the period $[0, t_1 + t_2]$ is the sum of the number of events happening during $[0, t_1]$ and $[t_1, t_1 + t_2]$. The number of events happening during $[t_1, t_1 + t_2]$ has the same distribution law as the number of events happening during $[0, t_2]$.

We will use this property in the lemma bellow.

Let us define RVs

$$X'_M(n) = N_M(S'_H(n)), \tag{12}$$

that are the numbers of blocks that MMs generate till the time when HMs generate and share n blocks (in other words: the number of blocks mined by MMs by the time when HMs have just shared their n-th block).

Also, let us define RVs

$$X_M(n) = N_M(S_H(n)) \tag{13}$$

in the same way.

Now we are going to find the distribution function of RV $X'_M(n)$.

Lemma 5.2. *Let us define*

$$P_n(k) = P\left(X'_M(n) = k\right).$$

Then the following statements are true:

(1) *RV $X'_M(n)$ is the sum of two RVs:*

$$X'_M(n) = X_M(n) + N_M(nD_H) = N_M(S_H(n)) + N_M(nD_H), \tag{14}$$

where $S_H(n)$ and $X_M(n)$ were defined in (8) *and* (13)*, respectively.*

(2) *RV $N_M(S_H(n))$ has the negative binomial distribution with parameters (n, p_H) and RV $N_M(nD_H)$ has the Poison distribution with the parameter $\alpha_M nD_H$:*

$$P(N_M(S_H(n))) = C^k_{n+k-1} p^n_H p^k_M, \tag{15}$$

$$P(N_M(nD_H) = k) = \frac{e^{-\alpha_M nD_H} \cdot (\alpha_M nD_H)^k}{k!}. \tag{16}$$

(3) *Probability distribution for RV $X'_M(n)$ is:*

$$P_n(k) = \frac{p^n_H}{(n-1)!} \cdot \frac{e^{-\alpha_M nD_H} \cdot (\alpha_M nD_H)^k}{k!} \cdot \sum_{i=0}^{k} \frac{(n-i+1)! \cdot C^i_k}{(\alpha nD_H)^i}, \tag{17}$$

where

$$\alpha = \alpha_M + \alpha_H.$$

Proof. (1) According to the definitions (12), (7) and (9) we get

$$\begin{aligned} X'_M(n) &= N_M(S'_H(n)) = N_M(S_H(n) + nD_H) \\ &= N_M(S_H(n)) + N_M(nD_H), \end{aligned}$$

where the latter equality was explained in Note 2, and (14) is proved.

(2) To prove (15), note that RV $N_M(S_H(n))$ is the same as RV X_n from the paper [3] that has the negative binomial distribution with parameters (n, p_H), as proved in Proposition 5.1 of the mentioned paper. Next, according to Lemma 5.2, $N_M(nD_H)$ has the Poison distribution with the parameter α_M, whence we get (16).

(3) According to the part 2 of this Lemma,

$$N_M(S'_H(n)) = N_M(S_H(n)) + N_M(nD_H),$$

where two RVs in the right-hand side of this equality are independent. So the sum of these values has the distribution that is the convolution of their distributions:

$$\begin{aligned} P_n(k) &= \sum_{i=0}^{k} P(N_M(S_H(n)) = i) \times P(N_M(nD_H)) = k - i) \\ &= \sum_{i=0}^{k} \left\{ C^i_{n+i-1} p_H^n p_M^i \frac{e^{-\alpha_M n D_H} \cdot (\alpha_M n D_H)^{(k-i)}}{(k-i)!} \right\} \\ &= \frac{P_H^n}{(n-1)!} \cdot \frac{e^{-\alpha_M n D_H} \cdot (\alpha_M n D_H)^{(k)}}{(k)!} \cdot \sum_{i=0}^{k} \frac{(n+i-1)! C^i_k}{(\alpha n D_H)^i}, \end{aligned}$$

where $\alpha = \alpha_M + \alpha_H$. The Lemma is completely proved. □

Note that in the case when $D_H = 0$, we have only one non-zero term in $P_n(k)$, when $i = k$ (because of $0! = 1$ and $0^0 = 1$), so in this case we get the negative binomial probability, as in the "classical" case with the zero synchronization time. This fact shows that (17) is a generalization of the corresponding result from [3].

Let's define the event E_n as "there exists some moment $t > 0$, that at this moment the adversary will manage to catch up from n blocks behind".

In other words, E_n means that at some moment $t > 0$ the adversary's branch will be of the same length as the honest miners' branch which is visible for all honest miners.

Lemma 5.3. *Let's define* $q_n = P(E_n)$.
Then

$$q_n = \begin{cases} 1, & if\ p'_M \geq p'_H, \\ \left(\frac{p'_M}{p'_H}\right)^n = \left(\frac{\alpha_M + \alpha_M(1 - e^{\alpha_M D_H})}{\alpha_H e^{-\alpha_H D_H}}\right)^n, & else. \end{cases}$$

Proof. Using the compound probability formula, we obtain:

$$q_n = P(E_n) = P(E_n/T'_H > T'_M)P(T'_H > T'_M) + P(E_n/T'_H < T'_M)P(T'_H < T'_M) = P(E_{n-1})p'_M + P(E_{n+1})p'_H,$$

where the latter equality was obtained using Lemma 4.1.
We can rewrite this expression as:

$$q_n = q_{n-1}p'_M + q_{n+1}p'_H. \tag{18}$$

To solve (18), we apply the characteristic equation

$$\lambda^2 p'_H - \lambda + p'_M = 0,$$

whose roots are $\lambda_1 = 1$ and $\lambda_2 = \frac{p'_M}{p'_H}$. So the general solution of (18) is

$$q_n = a\lambda_1^n + b\lambda_2^n = a + b\left(\frac{p'_M}{p'_H}\right)^n.$$

If $p'_M > p'_H$, then $\frac{p'_M}{p'_H} > 1$. But $0 \le q_n \le 1$, so in this case the only solution is $q_n = 1$.

Next, in the case when $p'_M = p'_H = \frac{1}{2}$ we get the equality

$$\lambda^2 - 2\lambda + 1 = 0,$$

whence $\lambda_1 = \lambda_2 = 1$ and $q_n = 1$ for $n \ge 1$; also, using the initial condition $q_0 = 1$.

At last, if $p'_M < p'_H$, from the boundary conditions $q_0 = 1$, $q_\infty = 0$ we obtain $a = 0$, $b = 1$ and

$$q_n = \left(\frac{p'_M}{p'_H}\right)^n.$$

The lemma is proved. □

Now we are ready to formulate the main result of this paper.

Theorem 5.4. *The probability of success by the MMs after z confirmation blocks mined by the HMs is:*

$$p(z) = \begin{cases} 1, & if\ p'_M \ge p'_H, \\ 1 - \sum_{k=0}^{z} P_z(k)\left(1 - \left(\frac{p'_M}{p'_H}\right)^{z-k}\right), & else. \end{cases} \tag{19}$$

Proof. For some fixed z, define the event $A_z(k)$ as

$$A_z(k) = \{N_M(S'_H(z) = k)\} = \{X'_M(z) = k\}, k \geq 0,$$

where $X'_M(z)$ was determined in (12).

Also, let us define the event A_z as "MMs succeed after z confirmation blocks have been mined by the HMs." Then

$$A_z = \left\{ \bigcup_{k \geq z} A_z(k) \right\} \cup \left\{ \bigcup_{k=0}^{z-1} (A_z(k) \cap E_{n-k}) \right\},$$

where the event E_{z-k} was introduced in Lemma 5.3.

Note that the events $\{\cup_{k \geq z} A_z(k)\}$ and $\{\cup_{k=0}^{z-1}(A_z(k) \cap E_{n-k})$ are disjoint, and the events $A_z(k)$ and E_{z-k} are independent. Next, according to Lemma 5.2, $P(A_z(k)) = P_n(k)$, and according to Lemma 5.3, $P(E_{z-k}) = \left(\frac{p'_M}{p'_H}\right)^{z-k}$.

Using these two equalities, we obtain:

$$\begin{aligned} P(A_z) &= \sum_{k=z}^{\infty} P(A_z(k)) + \sum_{k=0}^{z-1} P(A_z(k)) \cdot P(E_{z-k}) \\ &= 1 - \sum_{k=0}^{z-1} P(A_z(k)) + \sum_{k=0}^{z-1} P(A_z(k)) \cdot P(E_{z-k}) \\ &= 1 - \sum_{k=0}^{z-1} P(A_z(k)(1 - P(E_{z-k})) = 1 - \sum_{k=0}^{z-1} P_z(k) \left(1 - \left(\frac{p'_M}{p'_H}\right)^{z-k}\right), \end{aligned}$$

and the theorem is proved. □

Note that in the case when $D_H = 0$, the result of Theorem 1 also coincides with corresponding result from [3] and may be considered as its generalization.

6. Numerical results

Here we present the results obtained using Theorem 5.4.

Tables 1 and 2 provide the minimal numbers of confirmation blocks guaranteeing that probability (19) of a successful double spend attack is less than 10^{-3}, for various malicious hashrates p_M, synchronization time D_H, and for fixed block generation intensity α.

For Table 1, we take $\alpha = 1/600 = 0.00167$, as in BTC-network, and for Table 2 we take $\alpha = 1/60 = 0.0167$ – 10 times larger.

We also give Table 3 with the same values for parameter $\alpha = 1/15 = 0.066$ (as for Ethereum). But here we should note that consensus protocol for Ethereum isn't pure PoW [10], so the results obtained can't be directly used for ETH blockchain.

p_M	$D_H = 0$	$D_H = 15$	$D_H = 30$	$D_H = 60$	$D_H = 120$	$D_H = 180$
	z					
0.1	6	6	6	6	7	7
0.15	9	9	9	9	10	11
0.2	13	13	13	14	15	16
0.25	20	20	21	22	24	27
0.3	32	33	34	36	42	49
0.35	58	61	64	70	86	109
0.4	133	143	153	179	255	412
0.45	517	621	724	>800	>900	$P_{success} = 1$

Table 1. The results for $\alpha = 0.00167$ and different values of malicious hashrate and synchronization time

p_M	$D_H = 0$	$D_H = 5$	$D_H = 15$	$D_H = 30$	$D_H = 60$
	z				
0.1	6	6	7	8	10
0.15	9	9	10	12	17
0.2	13	14	16	20	32
0.25	20	21	25	34	73
0.3	32	36	45	69	401
0.35	58	68	97	203	$P_{success} = 1$
0.4	133	170	317	$P_{success} = 1$	$P_{success} = 1$

Table 2. The results for $\alpha = 0.0167$ and different values of malicious hashrate and synchronization time

p_M	$D_H = 0$	$D_H = 5$	$D_H = 8$
	z		
0.1	6	7	8
0.15	9	11	12
0.2	13	17	20
0.25	20	28	36
0.3	32	52	74
0.35	58	119	236
0.4	133	519	$P_{success} = 1$

Table 3. The results for $\alpha = 0.067$ and different values of malicious hashrate and synchronization time

As we can see from Table 1, when the malicious hashrate is e.g., $p_M = 0.1$ (or 10%), then for $0 \leq D_H \leq 60$ sec it is sufficient to wait 6 confirmation blocks to be sure that the probability of attack is less than 10^{-3}. Note that 6 confirmation blocks is the value which is widely used now in the BTC-network, but only from some empirical considerations and without any justification. In the case when the malicious hashrate is $p_M = 0.3$ (or 30%), then the minimal number of confirmation blocks is 32 for $D_H = 0$, 33 for $D_H = 15$ sec and 34 for $D_H = 30$ sec. But in the case when $p_M = 0.45$ (or 45%) and $D_H = 180$ sec, any number of confirmation blocks cannot prevent the attack, its probability is 1 anyway. In terms of [5] it means that for these parameters, the security threshold is smaller than 45%. For $p_M = 0.45$ and $D_H = 60; 120$ calculation of z requires much time, so we give the next estimations: $z > 800$ for $D_H = 60$ and $z > 900$ for $D_H = 120$.

For a large parameter $\alpha = 1/60 = 0.0167$ in Table 2, the probability of attack is equal to 1 for smaller values of malicious hashrate and of synchronization time, e.g., for $p_M = 0.35$, $D_H = 60$ sec, or for $p_M = 0.4$, $D_H = 30$ sec. Thus, as shown in [5], the security threshold decreases when block intensity generation increases. In this paper we also show that the probability of attack increases with block intensity generation, hence for large intensity we need more confirmation blocks to prevent the attack.

Figure 1 shows dependency graph of the number of confirmation blocks on malicious hashrate for different values of D_H and fixed block generation intensity α. One can see that as the number of the attackers increases, the proportion of the attackers sufficient to carry out an attack with probability 1 decreases, and the number of confirmation blocks required to prevent the attack increases.

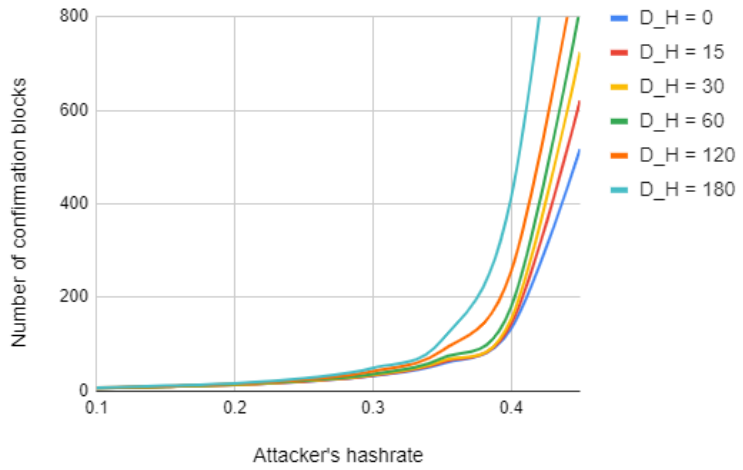

Figure 1. Number of confirmation blocks needed to provide the probability of successful attack 1/1000 (block generation time of 600 sec)

Figures 2 and 3 shows the attack success probability depending on p_M and for different α.

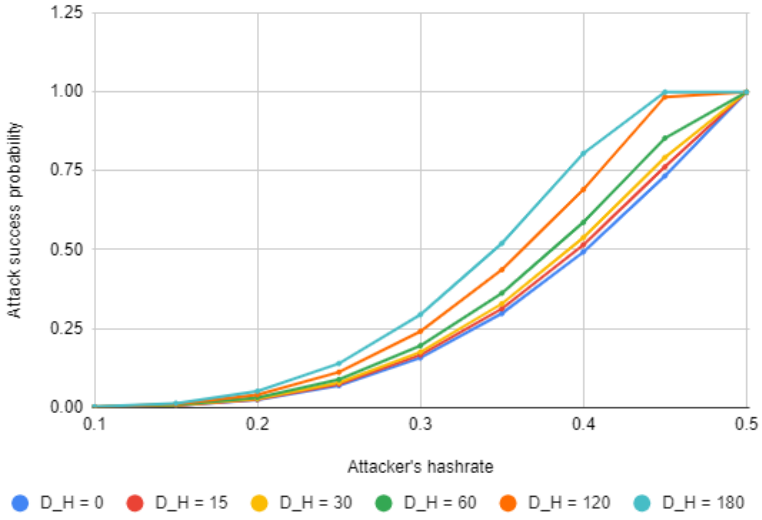

Figure 2. Attack success probability (block generation time of 600 sec, 6 confirmation blocks)

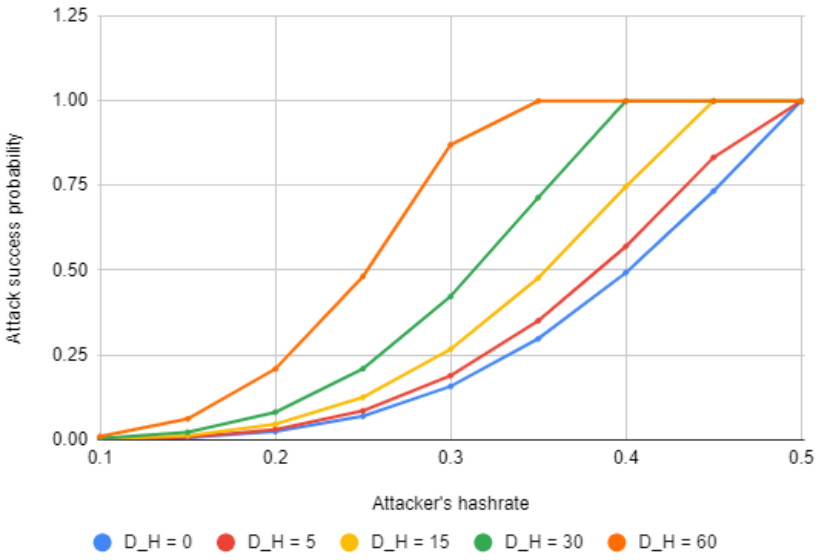

Figure 3. Attack success probability (block generation time of 60 sec, 6 confirmation blocks)

7. Conclusions

In this paper we obtained and rigorously substantiated analytical expressions for probabilities of a double spend attack on the blockchain. It is shown that the probability of an attack depends not only on the attacker's share, but also on the network synchronization time and intensity of block generation.

Network synchronization time is a critical parameter that needs to be taken into account when analyzing the resistance of the network to a double spend attack. It is shown that this parameter affects not only the required number of confirmation blocks allowing to defend against the attack, but also the attacker's share allowing realization of an attack with probability 1. For example, with a sufficiently long synchronization time, the attacker may have significantly less than 50% of the hash rate for guaranteed success in attack.

Our results generalize and complement the results obtained by Grunspan and Ricardo Perez-Marco in [3]. The main result of [3] may be obtained from Theorem 5.4 and (19) as a particular case when time delay is equal to zero.

References

[1] J. Garay, A. Kiayias and N. Leonardos, The bitcoin backbone protocol: Analysis and applications, In: Annual international conference on the theory and applications of cryptographic techniques, 2015, 281–310.

[2] P. Gaži, A. Kiayias and A. Russell, Tight consistency bounds for bitcoin, In: Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, 2020, 819–838.

[3] C. Grunspan and R. Pérez-Marco, Double spend races, *International Journal of Theoretical and Applied Finance* **21 8** (2018), 1850053.

[4] L. Kovalchuk, D. Kaidalov, A. Nastenko, M. Rodinko, O. Shevtsov and R. Oliynykov, Number of Confirmation Blocks for Bitcoin and GHOST Consensus Protocols on Networks, *Theoretical and Applied Cybersecurity* **1 1** (2019).

[5] L. Kovalchuk, D. Kaidalov, A. Nastenko, M. Rodinko, O. Shevtsov and R. Oliynykov, Decreasing security threshold against double spend attack in networks with slow synchronization, *Computer Communications* **154** (2020), 75–81.

[6] S. Nakamoto, A peer-to-peer electronic cash system, 2008, https://bitcoin.org/bitcoin.pdf.

[7] C. Pinzón and C. Rocha, Double-spend attack models with time advantange for bitcoin, *Electronic Notes in Theoretical Computer Science* **329** (2016), 79–103.

[8] M. Rosenfeld, Analysis of hashrate-based double spending, *arXiv preprint arXiv:1402.2009* (2014).

[9] Y. Sompolinsky and A. Zohar, Secure high-rate transaction processing in bitcoin, In: International Conference on Financial Cryptography and Data Security, 2015, 507–527.

[10] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, *Ethereum project yellow paper* **151 2014** (2014), 1–32.

LYUDMILA KOVALCHUK†, MARIIA RODINKO‡, ROMAN OLIYNYKOV‡,
DMYTRO KAIDALOV*, ANDRII NASTENKO*
† ‡ *INPUT OUTPUT HK

†NATIONAL TECHNICAL UNIVERSITY
OF UKRAINE "IGOR SIKORSKY KYIV
POLYTECHNIC INSTITUTE"

‡V.N. KARAZIN KHARKIV
NATIONAL UNIVERSITY

E-mail: {lyudmila.kovalchuk,mariia.rodinko,roman.oliynykov,
dmytro.kaidalov,andrii.nastenko}@iohk.io