UDC 004.75

COMPARISON OF BLOCK EXPECTATION TIME FOR VARIOUS CONSENSUS ALGORITHMS

Kaidalov D. S. – PhD, Research Fellow at Input Output HK.

Kovalchuk L. V. – Dr. Sc., Professor, Department of Mathematical Methods of Information Security, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Research Fellow at Input Output HK.

Nastenko A. O. – PhD, Research Fellow at Input Output HK.

Rodinko M. Yu. – Post-graduate student, Department of Information Systems and Technologies Security, V. N. Karazin Kharkiv National University, Research Fellow at Input Output HK.

Shevtsov O. V. – PhD, Research Fellow at Input Output HK.

Oliynykov R. V. – Dr. Sc., Professor, Department of Information Systems and Technologies Security, V. N. Karazin Kharkiv National University, Research Fellow at Input Output HK.

ABSTRACT

Context. We consider security properties of decentralized blockchain-based consensus protocols. The object of research is block confirmation time for users to get assurance that their transaction will not be reverted.

Objective. The goal of the paper is to analyze double-spend attacks on the different blockchain-based systems and compare resulting probabilities of attacker’s success.

Method. We presented two models for two types of attacks on the Ouroboros protocol (for the general and covert adversaries). The models allow calculating the exact number of slots needed to achieve the required level of security. It was shown that the Ouroboros protocol allows achieving the required security level with significantly shorter confirmation period in comparison with Bitcoin. We estimated minimal number of confirmation blocks and compare estimation time for Bitcoin, GHOST and Ouroboros protocols. As a measure of comparison, we considered transaction confirmation time for which the probability of a double-spend attack is less than 0.1%. We use different standard probability distribution and different properties of Markov chains and Random Walks to get comparison of estimated security properties of Bitcoin blockchain against three different models of Bitcoin double spend attack. The splitting attack based on the model where resources of honest participants are divided to compete different chains is applied to Bitcoin and GHOST consensus protocols. Properties of Markov chains and Random Walks are also applied to obtain security estimations for the Ouroboros protocol.

Results. We developed methods to get specific numbers for average block confirmation time for Ouroboros protocol. We compared minimal number of confirmation blocks needed to ensure a high security for considered protocols: Bitcoin, GHOST and Ouroboros.

Conclusions. The obtained results allow determination of security bounds for the Bitcoin, GHOST and Ouroboros consensus protocols. Users of the practically deployed blockchain systems may get specific parameters for a given assurance level.

KEYWORDS: blockchain, Bitcoin, proof-of-work, GHOST, proof-of-stake, Ouroboros.

ABBREAVIATIONS

GHOST is a Greedy Heaviest-Observed Sub-Tree;
HM is an honest miner;
MM is a malicious miner;
PoW is a Proof of Work;
PVSS is a Publicly Verifiable Secret Sharing.

NOMENCLATURE

α is some small probability;

α_z the probability that an adversary would be able to catch up when he is z blocks behind;

m is the number of blocks in the honest chain;

$m(w) = (\lambda, \mu)$ is a state of the string w represented by two variables λ and μ;

$m(\varepsilon) = (0,0)$ is the initial state of the algorithm;

n is the number of blocks in the adversarial chain;

p is the fraction of hashing power that is possessed by honest nodes (equivalent to the probability that an honest node finds the next block);

q is the fraction of adversarial hashing power (equivalent to the probability that an adversary finds next block);

q_K is the probability that an adversary would ever catch up with the deficit of K blocks;

t is the time advantage of an adversary towards fraudulent block production;

w is a characteristic string.

INTRODUCTION

The Bitcoin is a payment system where digitally signed transactions are grouped into blocks and stored securely in a structure called blockchain. A blockchain is a sequence of blocks linked via hash pointers where each new block contains a hash of the previous block. This structure preserves an ordered list of transactions that uniquely determines the state of the system.

Unlike other centralized payment systems, in Bitcoin, once a transaction is added to the blockchain, it could not be considered as confirmed immediately. A user needs to wait some time to be sure that the transaction is set in stone in the blockchain. This is because of decentralized

DOI 10.15588/1607-3274-2018-4-15

nature of the system where everyone can add blocks to the blockchain. To provide consistency among different users and to preserve inability to revert previously added blocks, a special mechanism is used called proof-of-work. The following idea underlies a proof-of-work system: a computational effort (calculation of a hash value below some target) should be applied to produce a block. Only a chain of blocks with the most computations would be considered valid.

As the blockchain technology evolves, the alternatives to the computationally heavy proof-of-work mechanism appear. The most promising one is called proof-of-stake: it does not require heavy computations to produce blocks, instead, a block producer is chosen through a fair procedure among all stakeholders in the system. The Ouroboros is a good example of such a system [1]. To the best of our knowledge, it is the first provably secure proof-of-stake protocol with rigorous security guarantees.

The concept of a blockchain could be undermined if someone would have a possibility to revert blocks by submitting a chain that would substitute the one currently accepted. For example, such possibility can result in the following attack: some buyer pays to a merchant with bitcoins, after the corresponding transaction is included into the blockchain, the merchant accepts a payment and sends a product to the buyer; upon receiving the product the buyer issues a conflicting chain of blocks which does not contain the payment to the merchant but instead sends coins back to the buyer. So as long as the merchant cannot be sure that the payment is irreversible, it would not be secure to deliver the product.

S. Nakamoto argues [2] that the system is secure (with some probability) against such attacks, unless 50% or more of the total computational power possessed by an adversary.

The described double-spend attack is relevant not only for Bitcoin, but also for other proof-of-work systems, for instance, those based on the GHOST algorithm [3], as well as for proof-of-stake systems, like Ouroboros.

The object of study is to focus on the block confirmation time needed to provide reasonable security guarantees for the users.

The subject of study is comparison of block expectation time for popular proof-of-work and proof-of-stake consensus algorithms.

The purpose of the work is to analyze known double-spend models for Bitcoin and evaluate how effective an adversary can be in terms of probability of successful attack. For that purpose we present new mathematical models for the Ouroboros protocol that allows calculating the security bounds for different types of adversaries. We also provide the results of splitting attack simulations for Bitcoin and GHOST algorithms.

1 PROBLEM STATEMENT

In this paper we describe known double-spend models for Bitcoin and present new mathematical models for the Ouroboros protocol that allow calculating the security bounds for different types of adversaries.

Suppose there is the set of miners which is divided into honest miners and malicious miners.

The input values are p, q and $\alpha = 0{,}001$.

The problem is: given p, q, α, find the minimal number z of confirmation blocks, that the probability of double-spend attack after these blocks is less than α.

We build the estimation for minimal number of confirmation blocks and also compare estimation time for different protocols: Bitcoin, GHOST and Ouroboros.

2 REVIEW OF THE LITERATURE

The existing mathematical models of the Bitcoin double-spend attack are presented in [1, 2, 4, 5, 6].

The first model of double-spend attack was introduced by S. Nakamoto in the original Bitcoin white paper [2]. S. Nakamoto considers the scenario when an adversary tries to generate secretly an alternate chain that would be longer (in terms of computational difficulty) than the honest chain.

M. Rosenfeld improved the Nakamoto's model in [5], but did not give any rigorous justification for it. Mathematically description of the attack was given for the first time in paper [6] by Grunspan and Perez-Marco. We also look into two models proposed by C. Pinzon et al. [4] that introduce a notion of time advantage to the original model that was analyzed by Nakamoto and Rosenfeld.

The splitting attack was described in [7] and could be considered as a variation of a double-spend attack since the main goal is to create a fork of the required length. The splitting attack for the GHOST protocol is slightly different compared to Bitcoin [7].

Let's consider a double-spend attack that could happen in a blockchain-based system [8]. As we briefly mentioned before, it does not really matter what type of consensus mechanism underlies the system, a double-spend could happen in both proof-of-work and proof-of-stake systems. Here we describe the main essence of the attack.

As it follows from the name, the whole idea of a double-spend attack is to use the same coins twice. In general, it implies that someone pays for some goods, but after receiving them, he/she reverts the payment so both goods and money are in the hands of the attacker. While it is infeasible to change the transaction with the payment itself (because that would require falsifying of a digital signature), it is possible to reject an entire block which includes the transaction. For doing this, an attacker needs to substitute a valid sub-chain of blocks with a new one that has a bigger score (score calculation depends on the actual blockchain type). Even though this attack requires tremendous resources (computational in the case of a proof-of-work or financial in the case of a proof-of-stake system), it could be profitable.

The attack involves next steps:

1. An adversary A wants to buy some goods from a merchant B. To do this, A creates a transaction tx_1 with a payment to B and sends it to the blockchain (Fig. 1).

©Kaidalov D. S., Kovalchuk L.V., Nastenko A. O., Rodinko M. Yu., Shevtsov O. V., Oliynykov R. V, 2018
DOI 10.15588/1607-3274-2018-4-15

2. B receives the payment from A, he waits for sufficient number of confirmations in the blockchain and then sends goods to A (Fig. 2).

3. A creates a conflicting transaction tx_2 where he redirects coins to his address, and tries to generate a forked block containing this transaction. Given that B waits for additional confirmations on top of the block with the payment, A needs to overcome all those blocks in his chain and create a fork with a higher score (Fig. 3).

4. If A is lucky to produce a fork of the main chain, the transaction tx_1 would be removed from the blockchain. Instead, the transaction tx_2 would be included. The network will continue with the chain of the adversary, so the payment to the merchant B would be lost forever (Fig. 4). At the same time, the adversary A seizes both goods and money.

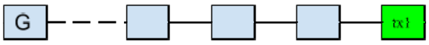

Figure 1 – Initial state of the blockchain from the genesis block G. The transaction tx_1 is included just into the latest block

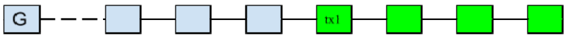

Figure 2 – Merchant B waits for 3 more blocks on top of the block with tx_1 and sends goods to A

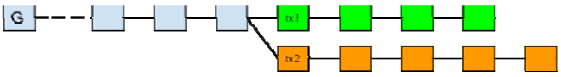

Figure 3 – An adversary creates a fork with a higher score

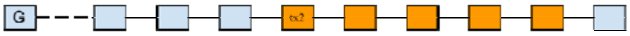

Figure 4 – The network continues with the chain of an adversary. tx_1 is substituted with tx_2

Even though specific techniques of fork creation could vary for different consensus protocols, the essence of the described attack remains the same for all of them.

Now we give an overview of the existing mathematical models of the Bitcoin double-spend attack.

In S. Nakamoto's model [2] given that an adversary starts with some deficit K (the honest chain is longer than adversarial on K blocks), the probability that an adversary would ever catching up with the honest chain is analogous to the Gambler's Ruin problem and could be calculated as follows:

$$q_K = \begin{cases} 1 & \text{if } p \le q; \\ (q/p)^K & \text{if } p > q. \end{cases}$$

Assuming that an adversary starts to work on the malicious fork right after the payment transaction is included into the blockchain (so does not wait for z blocks after which it is confirmed by the merchant), he may have mined some number of blocks so the deficit K is reduced. The adversarial progress will be a Poisson distribution with the expected value $\lambda = z\dfrac{q}{p}$.

The overall probability of the successful double-spend attack can be found by multiplying the Poisson density for each possible amount of progress by the probability of catching up with the remaining deficit:

$$DS_N(q,z) = \\ = \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{z-k}, & \text{if } k \le z; \\ 1, & \text{if } k > z. \end{cases} = \\ = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{z-k}\right). \qquad (1)$$

However, these results 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 generation of the block 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. It is as follows: if the probability of an event is p, then the number of tests in which there will be exactly n events, is exactly $\dfrac{n}{p}$. In fact, this means replacement of the random variable with its mathematical expectation, that is not entirely correct, to say it mildly.

Another well-known mathematical model for the Bitcoin double-spend attack, in addition to those presented by Nakamoto, is the model of M. Rosenfeld. In [5] he clarifies and expands the work of S. Nakamoto. The same basic model is taken: for a successful double-spend attack an adversary needs to catch up with $z = n - m$ blocks where n is the number of confirmations that a user waits to before sending goods, and m is the number of blocks that an adversary is expected to mine during the confirmation period.

M. Rosenfeld considers the catching-up process as a Markov chain, where each step is defined as finding of a block by an honest node or by an adversary:

$$z_{i+1} = \begin{cases} z_i + 1 & \text{with probability } p, \\ z_i - 1 & \text{with probability } q. \end{cases}$$

The attack succeeds if z ever reaches -1. If $z < 0$ then $\alpha_z = 1$, otherwise

$$\alpha_z = p\alpha_{z+1} + q\alpha_{z-1}.$$

DOI 10.15588/1607-3274-2018-4-15

In this case, the probability to catch up with z blocks can be defined as follows:

$$\alpha_z = \min(q/p,1)^{\max(z+1,0)} = \\ = \begin{cases} 1, & \text{if } z<0 \text{ or } q>p\,; \\ (q/p)^{z+1}, & \text{if } z\geq 0 \text{ and } q\leq p. \end{cases} \quad (2)$$

In the paper by Rosenfeld [5], 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 those used by Nakamoto.

M. Rosenfeld models the progress as a negative binomial distribution. The probability that an adversary will mine a given number of blocks m during an honest miner will mine n blocks is

$$P(m) = \binom{m+n-1}{m} p^n q^m. \quad (3)$$

It follows that the probability of a successful double-spend attack, where a merchant waits for n confirmations and an adversary succeeds to find $m+1$ blocks during the confirmation period is equal to

$$DS_R(q,n) = \sum_{m=0}^{\infty} P(m)\alpha_{n-m-1} = \\ = \begin{cases} 1-\sum_{m=0}^{n}\binom{m+n-1}{m}(p^n q^m - p^m q^n), & \text{if } q<p, \\ 1, & \text{if } q\geq p. \end{cases} \quad (4)$$

However, this paper did not provide any justification for this 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 [5], the results were also obtained under the assumption that the propagation time of the block in the network is zero. Regarding Nakamoto's second assumption, it is unclear how far the authors have noticed this fallacy; however, they did not use this 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.

An interested reader could find more rigorous description of this model in the original paper [5].

It is worth to mention two theoretical models that were presented by C. Pinzon et al. [4].

The first one generalizes the model of M. Rosenfeld by adding an extra parameter that represents time-advantage of an adversary.

The second one that is called "a time-based model" is completely different from those described above. In this model, the lengths of the valid and adversarial chains are assumed to be equal. Instead, the authors are focused on the time parameter t that represents the time difference between the n^{th} block in both the adversarial and honest chains.

Wonderful from the mathematical point of view, Grunspan's paper [6] impresses with the mathematical rigor of his 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. 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.

As far as these models are consistent with the model of M. Rosenfeld and give almost the same results, we do not examine them deeply. Short descriptions are given in the Appendices A and B.

Since all considered models are intended to estimate the probability of the same double-spend attack in Bitcoin, the results are similar except differences between the models of S. Nakamoto and others. The models of C. Grunspan, M. Rosenfeld and C. Pinzon et al. give exactly similar results (assuming that time advantage in the models of C. Pinzon is equal to zero).

The Table 1 shows the values computed for different models. It represents the number of blocks that a user should wait for to be 99.9% sure that his transaction would not be reverted by an adversary.

Table 1 – The number of blocks that a user should wait to be 99.9% sure that his transaction would not be reverted by an adversary with the given hashing power

Adversarial hashing power	The model of S. Nakamoto	The models of Rosenfeld and Grunspan	The model of C. Pinzon (generalized)
0.1	5	6	5
0.15	8	9	8
0.2	11	19	12
0.25	15	20	19
0.3	24	32	32
0.35	42	58	58
0.4	89	133	134
0.45		539	541

It is worth noting that the presented theoretical models for the double-spend attack could also be applied to another Bitcoin-like proof-of-work systems.

Now let's consider the splitting attack [7] which is targeted at the proof-of-work based protocols with a short block generation time that is comparable to the block propagation time in the network.

We will start with a general overview of a splitting attack, and then provide some experimental results showing possibility of its application to different proof-of-work consensus protocols.

In contrast to the classic double-spend attack, where an adversary is supposed to create a fork secretly and publish it after getting goods and only in case if his chain is longer, the splitting attack is public for all nodes from

DOI 10.15588/1607-3274-2018-4-15

the beginning. Moreover, not only an adversary contributes blocks into the forked branches but also honest nodes.

The idea of the attack is the following: when a fork of depth 1 accidentally happens, an adversary splits its hashing power on both branches to keep their lengths equal as long as possible. In this case honest miners would also be split due to their arbitrary choice between branches of equal lengths. When honest miners publish a new block in one of the branches, an adversary publishes block in the other branch to keep the fork running (see Fig. 5). If branches are of the same length, then adversary does nothing so again honest miners are split in half.

So the adversary tries to keep both chains balanced by their lengths. If lengths differ, the adversary extends the chain that is behind by publishing some amount of blocks needed to equalize lengths of both chains. The attack continues till the adversary has sufficient amount of blocks for each chain in his reserves. If he cannot equalize chains' lengths at the end of some round, then the attack is finished.

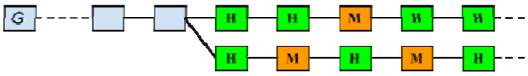

Figure 5 – The fork that keeps running while the adversary is able to equalize lengths of both branches with malicious blocks (marked with M)

A notion of a round was initially taken from [9]; it represents a complete round of information propagation to all nodes in a p2p network. In practice, information propagation is a random variable with an order of tens of seconds. In the described model, it is assumed that one full communication round takes 12.6 seconds (this is the average block propagation time in the Bitcoin network [10]).

A general essence of the splitting attack is the following: when the time of block generation is comparable to the time of block propagation, then the probability of generation of 2 or more blocks in the same round (and at the same block height) becomes non-negligible. In this case, at the beginning of the next round the network would be split into two branches. An adversary leverages such block collisions to keep the fork running.

Thus, an important parameter that facilitates a splitting attack is the number of PoW solutions (mined blocks) per complete round of information propagation. In [7], where this parameter is designated as f, it was shown that when f decreases and gets closer to 0, then the probability of a splitting attack decreases too (an adversary needs almost 50% of the hashing power to make a split). And vice versa, when f increases, the security bound becomes worse (the attack becomes feasible with less than 50% of the hashing power). The splitting attack is the most effective when $f \geq 1$, i.e., at the rate of 1 block per round or more.

It follows from the above that a short block generation time (relative to the block propagation time) creates favorable conditions for a splitting attack to occur. Hence, it becomes interesting to investigate resistance of proof-of-work protocols with different values of the parameter f .

Let's consider the splitting attack on GHOST. GHOST protocol was initially proposed as an improvement of the Bitcoin protocol that allows to reduce time between blocks while preserving the same level of security [3, 11].

The main modification that was suggested is that blocks not included into the main chain can still contribute to the chain's irreversibility. The basic observation behind the protocol is that the blocks that are built on top of some block B add additional weight to block B even if they are not in the main chain. So, in contrast to the Bitcoin protocol, where only the blocks that are in the main chain contribute to the difficulty of this chain, in GHOST a whole sub-tree of blocks is considered (Fig. 6). See for more information [3, 11].

Since it was declared by the authors that the GHOST protocol has a comparable security even with short block generation time (it is stated that even when blocks are issued every second, the security level is the same as in the original Bitcoin protocol, [3]), we found a few serious mistakes in their works that puts to doubt their assertions and results. So it becomes interesting to investigate resistance of the GHOST protocol against a splitting attack.

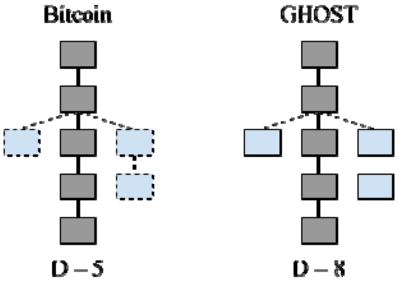

Figure 6 – Calculation of the chain's difficulty D is shown for the Bitcoin and GHOST protocols. In GHOST, even the blocks that are not included into the main chain add weight to it

The splitting attack for the GHOST protocol is slightly different compared to Bitcoin [7]. There are two differences:

– An adversary has to compensate the difference in the total number of honestly mined blocks in both branches at the end of each round, while in Bitcoin-like protocols he has to compensate only the maximal number of honestly mined blocks to keep both chains balanced.

– All blocks produced by an adversary are always valid. This facilitates an attack for adversary, because he can just mine the first nodes after the common prefix of

DOI 10.15588/1607-3274-2018-4-15

the two branches. In contrast, in Bitcoin an adversary has to extend only the head of diverging chains, so all blocks must be recent.

Now let's consider the double-spend attacks on Ouroboros. As stated in [1], it is the first provably secure proof-of-stake blockchain protocol with rigorous security guarantees, comparable to those achieved by the Bitcoin blockchain protocol. First we briefly discuss the protocol itself, and then present two models for different types of adversaries.

As previously stated, the Ouroboros is a proof-of-stake protocol, thus it does not require heavy computations for block production. While in the proof-of-work protocols like Bitcoin the blocks are produced by the miners (which do not necessarily have a stake in the system), in Ouroboros only the stakeholders can produce blocks. Given that the stakeholders are well incentivized to keep the overall stability of the system (as it would consequently keep the value of their coins), it creates an additional incentive for block producers to act honestly, thus making a system more secure in general.

The main idea behind the protocol is that the time is divided into so called epochs, and each epoch consists of a predefined number of slots. Each slot has an associated stakeholder that should produce a block during the time of that slot. The model requires synchrony among stakeholders, and the blocks that are produced in the incorrect timeslots are considered invalid. At most one block could be produced in the given slot (Fig. 7).

The owners of the slots are chosen randomly before the beginning of the epoch. Randomness for a selection procedure is generated collectively by a set of stakeholders by means of a special cryptographic protocol based on the PVSS scheme [12].

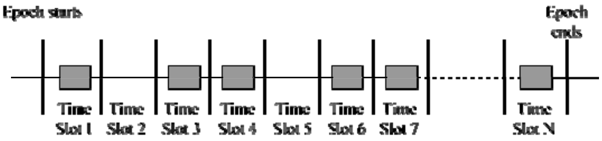

Figure 7 – A general scheme of the Ouroboros protocol

The time is divided into slots, each slot has an associated stakeholder who should produce a block in this slot. It is not necessary that the block in the given slot will be produced (for instance, a corresponding stakeholder could be offline at the moment), but there is a strict rule that only one block can be produced in the slot.

Following the terminology given in [1], an attack that consists in a fork creation is called an attack on a common prefix. There are two possible models for an adversary that is going to create a fork: the one that immediately demonstrates an adversarial behavior and the one that leaves an adversary covert. We will briefly describe both of them.

Despite of the rule that a slot winner can produce only one block per slot in the given chain of blocks, nothing can prevent him from creating several blocks in the same slot but in different chains, thus creating a fork (see Fig. 8). An adversary can facilitate an attack by publishing blocks in both chains forcing honest slot winners to be split between them. In what follows, we will call such adversary a general adversary.

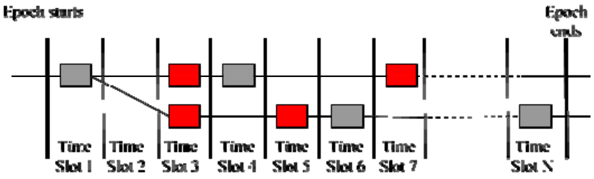

Figure 8 – An adversary that possesses some slots (shown in red) tries to split honest slot winners into two chains, thus facilitating an attack

While the described attack provides an adversary with significant opportunities, it leaves a suspicious "audit trail" – multiple signed blocks at the same slot that immediately signals malicious behavior. That motivates to consider a restricted class of covert adversaries, who produce not more than one block per slot (though not necessarily in the expected slot [1]).

An interested reader could find more details in [1, 13].

3 MATERIALS AND METHODS

Let's consider Ouroboros general adversary model. A central point of the security arguments given in [1] is the notions of the characteristic and forkable strings. A characteristic string is a binary string $\{0,1\}^n$ where each element indicates a slot that is assigned either to an adversary (denoted with 1) or to an honest user (denoted with 0). A forkable string is a characteristic string with such disposition of adversarial slots that allows fork creation.

Understanding density of the forkable strings among all characteristic strings will help to determine the probability of an attack. The paper [1] gives an upper bound on the probability of a string being forkable. In our research, we are interested in the exact probabilities of forks. To obtain such probabilities, we utilize a recursive algorithm that detects a forkable string (see lemma 4.18 in [1] for more details):

$$m(w0)=\begin{cases}(\lambda(w)-1,0), & \text{if } \lambda(w)>\mu(w)=0;\\ (0,\mu(w)-1), & \text{if } \lambda(w)=0;\\ (\lambda(w)-1,\mu(w)-1), & \text{otherwise.}\end{cases} \qquad (5)$$

Given a characteristic string w and the initial state $m(\varepsilon)$, the state is updated sequentially with each element of the string. Finally, when all elements from w are processed, the variable μ is checked: if $\mu \ge 0$ then the string w is forkable, otherwise it is not.

DOI 10.15588/1607-3274-2018-4-15

Having such an algorithm, it is possible to calculate the overall probability of a fork for a string of particular length. It could be done by constructing of a matrix of probabilities for all possible states (Fig. 9).

The matrix could be calculated iteratively using the following rules (based upon the algorithm (5)):

$$p^0_{0,0} = 1 \text{ and } p^0_{i,j} = 0, \text{ for } i \neq 0 \text{ or } j = 0,$$

$$p^n_{i,j} = \text{lam1}\cdot q\cdot p^{n-1}_{i-1,j-1} + \text{mu1}(1-q)\,p^{n-1}_{i+1,j+1} + \\ +\text{mu2}(1-q)\,p^{n-1}_{i+1,0} + \text{lam2}(1-q)\,p^{n-1}_{0,j+1},$$

$$\text{mu1} = \begin{cases} 1 & \text{if } j \neq 0, \\ 0 & \text{otherwise;} \end{cases} \qquad \text{mu2} = \begin{cases} 1 & \text{if } j = 0, \\ 0 & \text{otherwise.} \end{cases}$$

$$\text{lam1} = \begin{cases} 1 & \text{if } i > 0, \\ 0 & \text{otherwise;} \end{cases} \qquad \text{lam2} = \begin{cases} 1 & \text{if } i = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the probability that an adversary with the fraction of stake q would be able to create a fork of n slots could be defined as follows:

$$DS(q,n) = \sum_{i=0}^{n}\sum_{j=0}^{n} p^{(n)}_{i,j}. \qquad (6)$$

Note that it is also possible to estimate the probability of a fork by simulating an attack directly. It could be done by generating of random binary strings (taking into account the probability of an adversarial slot) and checking them with the algorithm (5). The results conform with those obtained analytically with the equality (6).

Now let's consider Ouroboros covert adversary model. As stated previously, a covert adversary tries to keep an attack in secret, until he creates a branch of sufficient length. In this case, an adversarial behavior would be to refrain from publishing of blocks in the honest chain (Fig. 10).

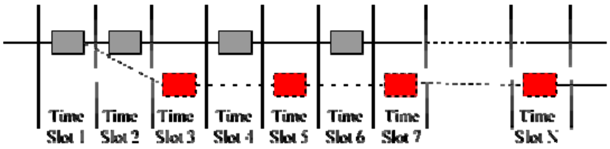

Figure 10 – A covert adversary tries to accumulate sufficient amount of slots (shown in red) to overcome an honest chain at some moment in future

In the classical double-spend attack it is assumed that an adversary has to create a fork of at least n blocks, where n is the number of confirmations that a user waits for before sending of goods or providing of a service. In this formulation, the attack with a covert adversary is basically close to the Bitcoin double-spend attack. Therefore, the probability of a fork after n blocks could be easily calculated using, for instance, the model of S. Nakamoto (see section 2, eq. 1).

Because of the deterministic nature of the block creation process in the Ouroboros protocol, it is more convenient to consider security bounds as the number of slots that a user should wait for to be sure (to some degree) that a fork cannot be created (opposite to the number of blocks in the classical model).

In our model, for a successful attack an adversary needs to create a fork of l slots (or longer). To do this, he needs to possess at least half of the slots at some point after the slot l. The probability of this event consists of two components: the ability of the adversary to accumulate some slots before the slot l, and the ability to catch up with the deficit (if any) after the slot l. We assume that neither honest users nor the adversary do not skip their slots, so there are no gaps.

The number of slots that an adversary would get during the period of l slots is a random variable that follows a binomial distribution. The probability to get exactly m slots is the following:

$$P(m) = \binom{l}{m} q^m p^{l-m}. \qquad (7)$$

The probability of catching up with $z = n - m$ slots (where $n = l - m$ is the number of honest slots) could be

$p^{(n)}_{0,-n}$	...	$p^{(n)}_{0,-2}$	$p^{(n)}_{0,-1}$	$p^{(n)}_{0,0}$	$p^{(n)}_{0,1}$	$p^{(n)}_{0,2}$	...	$p^{(n)}_{0,n}$
$p^{(n)}_{1,-n}$	...	$p^{(n)}_{1,-2}$	$p^{(n)}_{1,-1}$	$p^{(n)}_{1,0}$	$p^{(n)}_{1,1}$	$p^{(n)}_{1,2}$	...	$p^{(n)}_{1,n}$
$p^{(n)}_{2,-n}$	...	$p^{(n)}_{2,-2}$	$p^{(n)}_{2,-1}$	$p^{(n)}_{2,0}$	$p^{(n)}_{2,1}$	$p^{(n)}_{2,2}$	...	$p^{(n)}_{2,n}$
...	...	...	...	...	...	...	...	...
$p^{(n)}_{n,-n}$	...	$p^{(n)}_{n,-2}$	$p^{(n)}_{n,-1}$	$p^{(n)}_{n,0}$	$p^{(n)}_{n,1}$	$p^{(n)}_{n,2}$	...	$p^{(n)}_{n,n}$

Figure 9 – The matrix shows the probabilities of a random characteristic string w of length n being in the state $m(w) = (i, j)$. It is indexed by all possible values λ and μ that could be reached by the string of length n

©Kaidalov D. S., Kovalchuk L.V., Nastenko A. O., Rodinko M. Yu., Shevtsov O. V., Oliynykov R. V, 2018
DOI 10.15588/1607-3274-2018-4-15

defined as a particular case of the Gambler Ruin problem [14] as:

$$(\frac{q}{1-q})^{l-2m}.$$

It follows that the probability of a successful attack where an adversary creates a fork of l slots is equal to:

$$S(q,l)=\sum_{m=0}^{l}P(m)C(l-2m)= \\ =\sum_{m=0}^{\lfloor l/2\rfloor}\binom{l}{m}q^m(1-q)^{l-m}(\frac{q}{1-q})^{l-2m}+ \\ +\sum_{m=\lfloor l/2\rfloor+1}^{l}\binom{l}{m}q^m(1-q)^{l-m}. \qquad (8)$$

In order to get insights on the density of forks produced by different types of adversaries and to compare them with other consensus protocols, we made a calculation using the expressions above. The results are shown in Table 2.

Because synchrony between time slots is assumed in the Ouroboros protocol, it does not make sense to consider the parameter k (time between blocks) as we did for other consensus protocols.

Table 2 – The number of slots that a user should wait for to be 99.9% sure that his transaction would not be reverted by an adversary with the given stake

Adversarial stake	General Adversary	Covert Adversary
0.1	15	11
0.15	23	17
0.2	35	25
0.25	55	39
0.3	94	63
0.35	181	115
0.4	443	265
0.45	1990	1077

4 EXPERIMENTS

Firstly, in our experiments, we took two most widespread protocols: Bitcoin and GHOST and obtained experimental results during the computational modeling for both protocols.

As it is known, the average block generation time in Bitcoin is equal to 10 minutes [2]. Given that the average block propagation time is 12.6 seconds [2], the parameter $f=\frac{12.6}{10\cdot 60}=0.021$. In what follows, it is more suitable to use the parameter k instead of f that shows an average amount of communication rounds between 2 consecutive blocks: $k=\frac{1}{f}$. It is interesting to estimate the possibility of a successful splitting attack for the original choice $k=47.6$ made in Bitcoin, and see how security degrades in the case when k decreases. To accomplish this, we perform an experimental analysis of the described attack.

The next experiments included comparison among different consensus protocols and adversarial models described in the previous sections. As a unified measure, we took the number of block confirmations (or time slots in the case of Ouroboros) needed to be sure that a given block cannot be removed from the blockchain with the probability of at least 99.9% (in other words, the longest fork that an adversary with a certain hashing power/stake can create with the probability of at least 0.1%).

The chosen measure appears to be relevant for a real-world application because it shows how long a user should wait before accepting a payment transaction, thus decreasing the possibility of the considered attacks to a sufficient level.

To get further insights on the usability of the considered protocols, it is helpful to compare them by the average confirmation time. As long as different protocols have different time between blocks, this would give us more accurate picture of the security guarantees provided by protocols against different types of attacks.

The time between two consecutive slots in the Ouroboros system is expected to be 20 seconds. The average time to mine a Bitcoin block is 10 minutes [2]. During the analysis of the splitting attack, we also estimated the security bounds for the Bitcoin with reduced block generation time (12.6 seconds per block). The GHOST values of block generation time is the same as for Bitcoin.

5 RESULTS

Let's consider experimental results during the computational modeling for Bitcoin and GHOST protocols.

The results of the simulations for Bitcoin are summarized in Fig. 11. It is shown what fork length an adversary can maintain with the probability of success of at least 0.1%. It is easy to see that when the time between blocks decreases, an adversary gets a chance to create a longer fork.

Our simulation shows that for the choice of $k=47.6$ (like in Bitcoin) 6 confirmations are needed to be sure that the probability of a splitting attack is less than 0.1% (considering an adversary that possesses 35% of the hashing power). If we assume that the average block generation time is equal to the block propagation time (so that $k=1$) then 9 confirmation is needed for the same level of security.

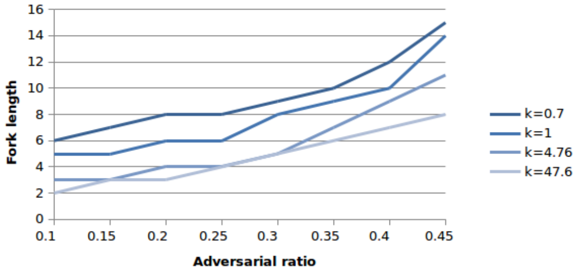

Figure 11 – The fork length that an adversary with a given hashing power can create with the probability of success of at least 0.1%. Different lines represents different choice of the parameter k

©Kaidalov D. S., Kovalchuk L.V., Nastenko A. O., Rodinko M. Yu., Shevtsov O. V., Oliynykov R. V, 2018
DOI 10.15588/1607-3274-2018-4-15

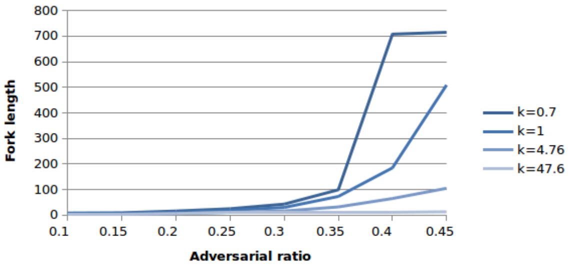

Figure 12 – The fork length that an adversary with a given hashing power can create for the GHOST protocol with the probability of success at least 0.1%. Different lines represents different choice of the parameter k

The results of the simulation (Fig. 12) for GHOST show that the attack is extremely effective when the parameter k is near to 1.

The summarized results of protocols' comparison are presented in Table 3. It includes two models for Ouroboros (with general and covert adversaries), classic Bitcoin double-spend attack, Bitcoin splitting attack (including hypothetical Fast Bitcoin with reduced block generation time to one per communication round, e.g. 12.6 sec) and GHOST splitting attack (both with 10 min and 12.6 sec blocks).

The Table 4 and Figure 13 show how long (in minutes) a confirmation period should be to reduce the probability of an attack to less than 0.1%.

Table 3 – The number of slots that a user should wait to be 99.9% sure that his transaction would not be reverted by an adversary with the given hashing power (or stake in the case of Ouroboros protocol). Note that for Ouroboros the values in the table represent the number of slots, while for other protocols they represent the number of blocks

Adversarial stake (hashing power)	Ouroboros General Adversary	Ouroboros Covert Adversary	Bitcoin (Rosenfeld)	Bitcoin splitting	Fast Bitcoin splitting	GHOST splitting	Fast GHOST splitting
0.1	15	11	6	3	6	3	6
0.15	23	17	9	4	7	4	8
0.2	35	25	13	4	8	6	11
0.25	55	39	20	5	9	9	19
0.3	94	63	32	6	10	9	30
0.35	181	115	58	8	12	11	73
0.4	443	265	133	9	14	12	185
0.45	1990	1077	539	14	18	13	509

Table 4 – An average confirmation time (in minutes) that guarantees, with the probability of more than 99.9% that a block would not be reverted from the blockchain

Adversarial stake (hashing power)	Ouroboros General Adversary	Ouroboros Covert Adversary	Bitcoin (Rosenfeld)	Bitcoin splitting	Fast Bitcoin splitting	GHOST splitting	Fast GHOST splitting
Block generation time	20 sec	20 sec	10 min	10 min	12.6 sec	10 min	12.6 sec
0.1	5	3.6	60	30	1.2	30	1.2
0.15	7.6	5.6	90	40	1.4	40	1.6
0.2	11.6	8.3	130	40	1.6	60	2.3
0.25	18.3	13	200	50	1.8	90	4
0.3	31.3	21	320	60	2.1	90	6.3
0.35	60.3	38.3	580	80	2.5	110	15.3
0.4	147.3	88.3	1330	90	2.9	120	38.8
0.45	663.3	359	5390	140	3.7	130	106.9

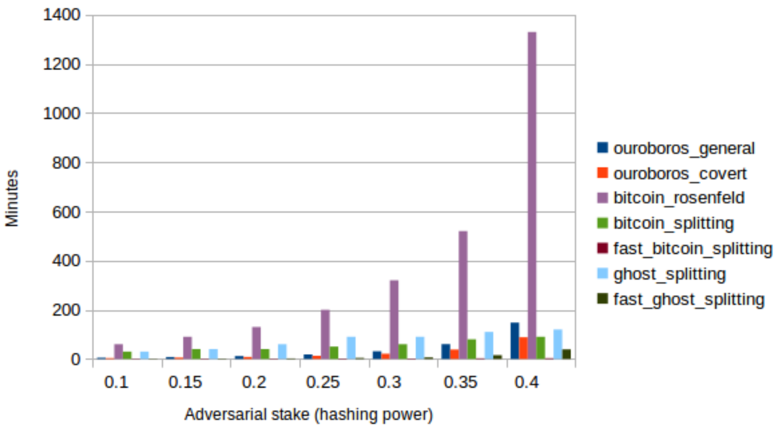

Figure 13 – Comparison of the expected confirmation periods (in minutes) for different protocols and adversarial models

6 DISCISSION

From the Table 4 and Figure 13 we can note that the Ouroboros protocol allows to confirm the block in 5 minutes in the worst case (considering an adversary with 10% of the total resources) while Bitcoin needs almost 60 minutes to provide the same level of security.

The splitting attack is more effective for the systems with short block generation time, but in general case, it is not better than the classical double-spend attack. Our simulations showed possibility of the attack for the Bitcoin and GHOST protocols with 10 min and 12.6 sec blocks. Not surprising that shorter blocks increase the required number of blocks to confirm a transaction but, despite this, the overall confirmation time is significantly reduced due to fast blocks.

CONCLUSIONS

In this paper we presented an analysis of the different consensus protocols and adversarial models. The main goal was to compare the well-known proof-of-work protocol that underlies Bitcoin with the new proof-of-stake algorithm that was introduced in Ouroboros. We also had a look at the GHOST algorithm that is initially intended to improve Bitcoin consensus. As a measure of comparison, we considered transaction confirmation time that allows to be sure that the probability of a double-spend attack is less than 0.1%.

The scientific novelty of obtained results: we presented two models for two types of attacks on the Ouroboros protocol (for the general and covert adversaries). The models allow calculation of the exact number of slots needed to achieve the required level of security. It was shown that the Ouroboros protocol allows achieving of the required security level with significantly shorter confirmation period compared to Bitcoin.

The practical significance consists in the fact that obtained results allow determination of the security bounds for the Ouroboros system. It becomes extremely important for a real-world application because it will help users to figure out how long they should wait before accepting the transaction.

APPENDIX A. THE GENERALIZED MODEL OF C. PINZON ET AL.

The model proposed by C. Pinzon et al. [4] generalizes the model of M. Rosenfeld by adding of an extra parameter that represents time-advantage of an adversary.

As in the previous models, a successful double-spend attack consists of two constituents: the progress of an adversary during the confirmation period of m blocks and his ability to catch up with the deficit $z = m - n$. The catch-up function is the same as originally used by S. Nakamoto (which occurs in Gambler's Ruin Problem). The improvement of this model lies in the modified progress function. It is represented as follows:

$$P(q, m, n, t).$$

Basically, the function P represents the probability of an adversary mining exactly n blocks once the honest network mines m blocks, assuming that an adversary has been additionally mining secretly for t time units. While the first three parameters (q, m, n) are well-known from the previous models, the time-advantage t is the new one. It represents an amount of time since the n^{th} block is found by an adversary until the m^{th} block is found by the honest network. This time period t potentially increases the probability of an adversary to find the next block faster than the honest network thus giving him an advantage.

In order to define the function P, it is necessary to define the function $a(q, t, k)$ that represents the probability to mine exactly k blocks during the time period t with a fraction q of hashing power (the proof could be found in the original paper [4]):

$$a(q,t,k) = \begin{cases} 1, & \text{if } t = n = 0, \\ 0, & \text{if } t \le 0, \\ \dfrac{(qt)^k}{k!} e^{-qt}, & \text{otherwise.} \end{cases}$$

The function P can be defined as follows:

$$P(q,m,n,t) = \sum_{z=0}^{n} a(q,t,z) P_R(q,m,n-z). \quad (9)$$

Note that in the case of $t = 0$ the progress function $P(q, m, n, t)$ is equivalent to the progress function presented by M. Rosenfeld [5].

Let $C_R(x, y)$ be the catch-up function as defined by M. Rosenfeld (eq. 2) and K – the number of blocks in the honest chain. It follows that the probability of a successful double-spend attack is equal to:

$$DS_G(q,K,n,t) = \\ = 1 - \sum_{z=0}^{K-n} P(q,K,z,t)(1 - C_R(q, K-n-z)). \quad (10)$$

Note that if the parameters $t = 0$ and $n = 1$ then this model is equivalent to the one proposed by M. Rosenfeld [5]. More information can be found in the original paper [4].

APPENDIX B. THE TIME-BASED MODEL OF C. PINZON ET AL.

The second model presented by C. Pinzon et al. is completely different from those described in section 2. In the time-based model, the lengths of the valid and fraudulent chains are assumed to be equal. Instead,

©Kaidalov D. S., Kovalchuk L.V., Nastenko A. O., Rodinko M. Yu., Shevtsov O. V., Oliynykov R. V, 2018
DOI 10.15588/1607-3274-2018-4-15

authors are focused on the time parameter t that represents time difference between the n^{th} block in adversarial and honest chains.

We will not go deep into the details of this model, instead we will only present the final equation for calculation of the probability of a double-spend attack. We refer an interested reader to the original paper [4] to find more details about this model.

Let P be the progress function from the generalized model (eq. (9)) and C_T is the catch up function for the time-based model that is defined as follows:

$$C_T(q,t) = \begin{cases} \frac{q}{p} e^{-(p-q)t}, & \text{if } t > 0, \\ 1, & \text{otherwise.} \end{cases}$$

The double-spend attack probability can be defined as the probability of having a time disadvantage t once the $K+1'th$ block is mined, multiplied by the probability of catching up with that disadvantage:

$$DS_T(q,K,n_0,t_0) = \\ = \int_{-\infty}^{\infty} P(q,K+1,K-n_0+1,t) C_T(q,t-t_0)dt. \tag{11}$$

The parameters in (11) are the same as in (10).

REFERENCES

1. Kiayias A. Ouroboros: A provably secure proof-of-stake blockchain protocol [Electronic resource], *Cryptology ePrint Archive*. Electronic data. [International Association for Cryptologic Research, 2016]. Mode of access: http://eprint.iacr.org/2016/889 (viewed on May 13, 2018). Title from the screen.
2. Nakamoto S. A. peer-to-peer electronic cash system" [Electronic resource], *Bitcoin*. Electronic data, 2008. Mode of access: https: //bitcoin.org/bitcoin.pdf (viewed on May 13, 2018). Title from the screen.
3. Sompolinsky Y., Zohar Aviv Accelerating bitcoin as transaction processing. Fast money grows on trees, not chains [Electronic resource], Cryptology ePrint Archive. Electronic data. [International Association for Cryptologic Research, 2013]. Mode of access: http://eprint.iacr.org/2013/881 (viewed on May 13, 2018). Title from the screen.
4. Pinzon C., Rocha C. Double-Spend Attack Models with Time Advantage for Bitcoin, Electronic Notes in Theoretical Computer Science, 2016, Vol. 329, pp. 79–103.
5. Rosenfeld M. Analysis of hashrate-based double-spending races [Electronic resource], Preprint arXiv. Electronic data, [Cornell: Cornell University, 2017], Mode of access: https://arxiv.org/abs/1402.2009 (viewed on May 13, 2018). Title from the screen.
6. Grunspan C., Pérez-Marco R. Double spend races [Electronic resource], Preprint arXiv. Electronic data, [Cornell: Cornell University, 2017]. Mode of access: https://arxiv.org/abs/1702.02867.pdf (viewed on May 13, 2018). Title from the screen.
7. Kiayias A., Panagiotakos G. Speed-security tradeoffs in blockchain protocols [Electronic resource] / A. Kiayias, // Electronic data. – [International Association for Cryptologic Research, Cryptology ePrint Archive, 2015]. – Mode of access: http://eprint.iacr.org/2015/1019 (viewed on May 13, 2018). Title from the screen.
8. Double-spending [Electronic resource], BitcoinWiki. Mode of access: https://en.bitcoin.it/wiki/Double-spending (viewed on May 13, 2018). Title from the screen.
9. Garay J. A. Kiayias Aggelos, and Leonardos Nikos The Bitcoin Backbone Protocol: Analysis and Applications", Advances in Cryptology, EUROCRYPT 2015, 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26–30, 2015: proceedings. Berlin, Springer, 2017. Part II, pp. 281–310. DOI: 10.1007/978-3-662-46803-6_10.
10. Decker C., Wattenhofer R. Information Propagation in the Bitcoin Net-work, Peer-to-Peer Computing: IEEE International Conference on Peer-to-Peer Computing (P2P), Trento, Italy, September 9–11, 2013, proceedings. Trento, IEEE Xplore, 2013, pp. 1–10. DOI: 10.1109/P2P.2013.6688704.
11. Sompolinsky Y., Zohar A. Secure high-rate transaction processing in Bitcoin, Financial Cryptography and Data Security – 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26–30, 2015: proceedings. Berlin, Springer, Lecture Notes in Computer Science, 2004, Vol. 8975, pp. 507–527. DOI: 0.1007/978-3-662-47854-7_32.
12. Schoenmakers B. A simple publicly verifiable secret sharing scheme and its application to electronic voting, Advances in Cryptology – CRYPTO 99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19, 1999: proceedings. – Berlin: Springer, 1999, Volume 1666 of Lecture Notes in Computer Science, pp. 148–164.
13. Russel A. Forkable Strings are Rare [Electronic resource], Cryptology ePrint Archive. Electronic data. [International Association for Cryptologic Research, 2017]. Mode of access: http://eprint.iacr.org/2017/241 (viewed on May 13, 2018). Title from the screen.
14. Feller W. An Introduction to Probability Theory and its Applications. New York: John Wiley & Sons, 1970, 700 p. DOI: 10.1137/1014119.

Received 15.05.2018.
Accepted 25.06.2018.

УДК 004.75

ПОРІВНЯННЯ ЧАСУ ПІДТВЕРДЖЕННЯ БЛОКУ ДЛЯ РІЗНИХ АЛГОРИТМІВ КОНСЕНСУСУ

Кайдалов Д. С. – канд. техн. наук, науковий співробітник в Input Output HK.

Ковальчук Л. В. – д-р техн. наук, професор, професор кафедри математичних методів захисту інформації Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», науковий співробітник в Input Output HK.

Настенко А. О. – канд. техн. наук, науковий співробітник в Input Output HK.

DOI 10.15588/1607-3274-2018-4-15

Родінко М. Ю. – аспірант кафедри безпеки інформаційних систем і технологій Харківського національного університету ім.В.Н.Каразіна, науковий співробітник в Input Output HK.

Шевцов О. В. – кандидат технічних наук, науковий співробітник в Input Output HK.

Олійников Р. В. – д-р техн. наук, доцент, професор кафедри безпеки інформаційних систем і технологій Харківського національного університету ім. В. Н. Каразіна, науковий співробітник в Input Output HK.

АНОТАЦІЯ

Актуальність. Проведений аналіз децентрализованих протоколів консенсусу, які базуються на блокчейні, з точки зору властивостей безпеки системи. Об’єктом дослідження є час підтвердження блоку із відповідним рівнем гарантій відсутності відміни транзакції для користувачів. Метою роботи є порівняння кінцевих ймовірностей успіху атаки подвійної витрати для різних систем на базі блокчейну.

Метод. Представлено дві моделі для двох атак на протокол Уроборос (для загального та прихованого зловмисників). Представлені моделі дозволяють обчислити точне значення числа слотів, необхідних для досягнення необхідного рівня безпеки. Показано, що протокол Уроборос дозволяє досягнути необхідного рівня безпеки за значно коротший період підтвердження у порівнянні з протоколом Біткоін. Зроблена оцінка та порівняння мінімального числа блоків підтвердження для протоколів Біткоін, GHOST та Уроборос. В якості міри порівняння було прийнято час підтвердження транзакції для якого ймовірність атаки подвійної трати менше, ніж 0,1%. Різні типи стандартних імовірнісних розподілів, а також властивості ланцюгів Маркова та випадкових блукань застосовується для отримання порівняння і оцінок властивостей безпеки блокчейна Біткоіна до трьох різних моделей атаки подвійної трати. Атака розгалуження, що заснована на моделі, де ресурси чесних учасників поділені для конкурування різних ланцюгів, застосована до протоколів консенсусу Біткоін і GHOST. Для оцінок безпеки протоколу Уроборос також використовуються властивості ланцюгів Маркова та випадкових блукань.

Результати. Розроблено методи для отримання точних значень середнього часу підтвердження блока для протоколу Уроборос. Зроблено порівняння мінімального числа блоків підтвердження для забезпечення високого рівня безпеки для протоколів Біткоін, GHOST та Уроборос.

Висновки. Отримані результати дозволяють визначити безпечні межі застосування протоколів консенсусу Біткоін, GHOST і Уроборос. Користувачі можуть отримати конкретні параметри для заданого рівня гарантій безпеки.

КЛЮЧОВІ СЛОВА: блокчейн, Біткоін, консенсус із доказом роботи, GHOST, консенсус із доказом володіння, Уроборос.

УДК 004.75

СРАВНЕНИЕ ВРЕМЕНИ ОЖИДАНИЯ БЛОКА ДЛЯ РАЗЛИЧНЫХ АЛГОРИТМОВ КОНСЕНСУСА

Кайдалов Д. С. – канд. техн. наук, научный сотрудник в Input Output HK.

Ковальчук Л. В. – д-р техн. наук, профессор, профессор кафедры математических методов защиты информации Национального технического университета Украины «Киевский политехнический институт имени Игоря Сикорского», научный сотрудник в Input Output HK.

Настенко А. О. – канд. техн. наук, научный сотрудник в Input Output HK.

Родинко М. Ю. – аспирант кафедры безопасности информационных систем и технологий Харьковского национального университетаим В. Н. Каразина, научный сотрудник в Input Output HK.

Шевцов О. В. – канд. техн. наук, научный сотрудник в Input Output HK.

Олейников Р. В. – д-р техн. наук, доцент, профессор кафедры безопасности информационных систем и технологий Харьковского национального университета им. В. Н. Каразина, научный сотрудник в Input Output HK.

АННОТАЦИЯ

Актуальность. Выполнен анализ децентрализованных протоколов консенсуса, основанных на блокчейне, с точки зрения свойств безопасности системы. Объектом исследования является время подтверждения блока с соответствующим уровнем гарантий отсутствия отмены транзакции для пользователей. Целью работы является сравнение результирующих вероятностей успеха атаки двойной траты для различных систем на базе блокчейна.

Метод. Представлены две модели для двух атак на протокол Уроборос (для общего и скрытого злоумышленников). Представленные модели позволяют вычислить точное значение числа слотов, необходимых для достижения необходимого уровня безопасности. Показано, что протокол Уроборос позволяет достичь необходимого уровня безопасности за более короткий период подтверждения по сравнению с протоколом Биткоин. Произведена оценка и сравнение минимального числа блоков подтверждения для протоколов Биткоин, GHOST и Уроборос. В качестве меры сравнения было принято время подтверждения транзакции, для которого вероятность атаки двойной траты меньше, чем 0,1%. Различные типы стандартных вероятностных распределений, а также свойства марковськых цепей и случайных блужданий используются для получения сравнения и оценок свойств безопасности блокчейна Биткоина к трем различным моделям атаки двойной траты. Атака разветвления, основанная на модели, где ресурсы честных участников разделены для конкуренции различных цепочек, применена к протоколам консенсуса Биткоин и GHOST. Для оценок безопасности протокола Уроборос также используются свойства марковских цепей и случайного блуждания.

Результаты. Разработаны методы для получения точных значений среднего времени подтверждения блока для протокола Уроборос. Сделано сравнение минимального числа блоков подтверждения для обеспечения высокого уровня безопасности для протоколов Биткоин, GHOST и Уроборос.

DOI 10.15588/1607-3274-2018-4-15

Выводы. Полученные результаты позволяют определить безопасные границы для применения протоколов консенсуса Биткоин, GHOST и Уроборос. Пользователи могут получить конкретные параметры для заданного уровня гарантий безопасности.

КЛЮЧЕВЫЕ СЛОВА: блокчейн, Биткоин, консенсус с доказательством проделанной работы, GHOST, консенсус с доказательством владения, Уроборос.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Kiayias A. Ouroboros: A provably secure proof-of-stake blockchain protocol [Electronic resource] / A. Kiayias // Cryptology ePrint Archive. – Electronic data. – [International Association for Cryptologic Research, 2016]. – Mode of access: http://eprint.iacr.org/2016/889 (viewed on May 13, 2018). – Title from the screen.
2. Nakamoto S. A. peer-to-peer electronic cash system” [Electronic resource] / Satoshi Nakamoto // Bitcoin. – Electronic data. – 2008. – Mode of access: https: //bitcoin.org/bitcoin.pdf (viewed on May 13, 2018). – Title from the screen.
3. Sompolinsky Y. Accelerating bitcoin as transaction processing. Fast money grows on trees, not chains [Electronic resource] / Yonatan Sompolinsky, Aviv Zohar // Cryptology ePrint Archive. – Electronic data. – [International Association for Cryptologic Research, 2013]. – Mode of access: http://eprint.iacr.org/2013/881 (viewed on May 13, 2018). – Title from the screen.
4. Pinzon C. Double-Spend Attack Models with Time Advantage for Bitcoin / C. Pinzon, C. Rocha // Electronic Notes in Theoretical Computer Science. – 2016. – Vol. 329. – P. 79–103.
5. Rosenfeld M. Analysis of hashrate-based double-spending races [Electronic resource] / M. Rosenfeld // Preprint arXiv. – Electronic data. – [Cornell: Cornell University, 2017]. – Mode of access: https://arxiv.org/abs/1402.2009 (viewed on May 13, 2018). – Title from the screen.
6. Grunspan C. Double spend races [Electronic resource] / C. Grunspan, R. Pérez-Marco // Preprint arXiv. – Electronic data. – [Cornell: Cornell University, 2017]. – Mode of access: https://arxiv.org/abs/1702.02867.pdf (viewed on May 13, 2018). – Title from the screen.
7. Kiayias A. Speed-security tradeoffs in blockchain protocols [Electronic resource] / A. Kiayias, G. Panagiotakos // Electronic data. – [International Association for Cryptologic Research, Cryptology ePrint Archive, 2015]. – Mode of access: http://eprint.iacr.org/2015/1019 (viewed on May 13, 2018). – Title from the screen.
8. Double-spending [Electronic resource] // BitcoinWiki. – Mode of access: https://en.bitcoin.it/wiki/Double-spending (viewed on May 13, 2018). – Title from the screen.
9. Garay J. A. The Bitcoin Backbone Protocol: Analysis and Applications” / Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos // Advances in Cryptology – EUROCRYPT 2015 – 34 th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26–30, 2015: proceedings. – Berlin : Springer, 2017. Part II. – P. 281–310. DOI: 10.1007/978-3-662-46803-6_10.
10. Decker C. Information Propagation in the Bitcoin Net- work / C. Decker, R. Wattenhofer // Peer-to-Peer Computing: IEEE International Conference on Peer-to-Peer Computing (P2P), Trento, Italy, September 9–11, 2013: proceedings. – Trento: IEEE Xplore, 2013. – P. 1–10. DOI: 10.1109/P2P.2013.6688704.
11. Sompolinsky Y. Secure high-rate transaction processing in Bitcoin / Y. Sompolinsky, A. Zohar // Financial Cryptography and Data Security – 19th International Conference, FC 2015, San Juan, Puerto Rico, January 26–30, 2015: proceedings. – Berlin : Springer, Lecture Notes in Computer Science, 2004. – Vol. 8975. – P. 507–527. DOI: 0.1007/978-3-662-47854-7_32.
12. Schoenmakers B. A simple publicly verifiable secret sharing scheme and its application to electronic voting / Schoenmakers B. // Advances in Cryptology – CRYPTO 99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15–19, 1999: proceedings. – Berlin : Springer, 1999, Volume 1666 of Lecture Notes in Computer Science. – P. 148–164.
13. Russel A. Forkable Strings are Rare [Electronic resource] / A. Russel // Cryptology ePrint Archive. – Electronic data. – [International Association for Cryptologic Research, 2017]. – Mode of access: http://eprint.iacr.org/2017/241 (viewed on May 13, 2018). – Title from the screen.
14. Feller W. An Introduction to Probability Theory and its Applications / W. Feller. – New York : John Wiley & Sons, 1970. – 700 p. DOI: 10.1137/1014119.

DOI 10.15588/1607-3274-2018-4-15