Multidimensional Blockchain: Construction and Security Analysis

Ilya Shilov and Danil Zakoldaev

1 Introduction

Cryptocurrencies and related technologies have appeared not a long time ago but have already gained a significant role in the sphere of information technologies. Among the most important features of blockchain technology, which emerged in 2008 with the invention of Bitcoin, was a consensus mechanism with resistance against 50% of adversarial users. This technology predefined the success of Bitcoin and its sustainable position [1].

On its basis a large number of related technologies had emerged. They applied various changes to the technology and its inner protocols. At the top of this development was the Ethereum project, which significantly transformed the principles of building decentralized systems. As mentioned by its authors, blockchain could be used not only for the construction of cryptocurrencies but for a wider range of spheres. The main advantage of Ethereum was a Turing-complete language, which had enriched blockchain with the possibility of programming.

In general, the following advantages of blockchain technology can be named:

1. Decentralization [6].
2. Reaching consensus in the presence of faults and adversarial actions.
3. The possibility of building decentralized applications.

These advantages attract business, technology companies, and financial corporations to blockchain and related systems. However, it is necessary to admit that blockchain is not deprived of disadvantages. Of those following have a special meaning:

I. Shilov (✉) · D. Zakoldaev
ITMO University, St. Petersburg, Russia
e-mail: ilia.shilov@itmo.ru; d.zakoldaev@itmo.ru

K. Daimi et al. (eds.), *Principles and Practice of Blockchains*,
https://doi.org/10.1007/978-3-031-10507-4_3

1. Unconstrained growth of blockchain size, which complicated its storage and leads to partial centralization due to high requirements for maintaining nodes.
2. Significant expenses for maintaining some consensus mechanisms.
3. Uncontrolled economical processes (applying to cryptocurrencies).
4. Presence of intermediaries in intersystem exchange.

A large part of research has been taken recently to overcome these problems. In particular, the IOHK company has proven the security (in probabilistic sense) of a system based on proof-of-stake consensus mechanism, which is by far less expensive than proof of work [2, 11, 17]. Other approaches to building less power-consuming consensus mechanisms have also been presented [5]. Uncontrolled economical processes are partially handled by the KYC and AML policies, which allow to partially deanonymize operations with cryptocurrencies and increase the trust to cryptocurrencies from business.

However, until the present day, the size of blockchain remains a complicated problem. Blockchain implies replication of the complete database of blocks on all nodes in the network. Moreover, it is necessary to have access to complete block history to verify the chain correctness. Although the attempts to solve this problem have been undertaken, the provided solutions either have not been presented or solve the problem partially. The problem of blockchain size is shown in Fig. 1.

The problem of intersystem exchange is of great importance. Now such operations require using intermediate parties or sidechains. In practice these approaches do not solve the problem but move it to another level of abstraction.

This chapter summarizes the main advances in the direction of solving these problems, which have been achieved by the author in recent years. The concept of multidimensional blockchain is shown, and its protocols and components are

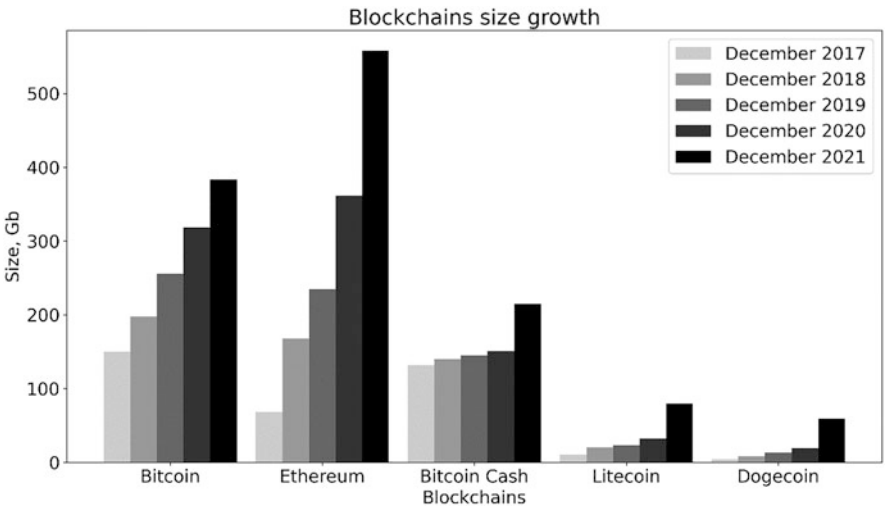

Fig. 1 Volume size growth for some famous systems based on the blockchain technology

described. Next, a brief overview of security analysis is given. On the basis of existing solutions, a novel search and verification protocol for blocks and transactions is presented, and its security is briefly examined. Finally, the experimental results and theoretical comparison for multidimensional blockchain and alternative systems are given.

2 Robust Distributed Ledgers

Before proceeding to the description and analysis of multidimensional blockchain, it is necessary to present several important terms used throughout the research. One of the most important applications of distributed systems is the *ledger*. In literature dedicated to research on consensus mechanisms, this term was created not a long time ago, and it is relatively rarely used in the sphere of cryptocurrencies.

In cryptocurrencies and database management systems, ledger is an ordered set of transactions. In practice ledgers are implicitly used in almost all database management systems. Moreover, a critical component of any automated banking system (ABS) is an ordered sequence of transactions, which also implies using a ledger. Finally, versioning systems, domain name systems, and many other distributed applications in some way use ordered sequence of transactions, which provides perspectives of using distributed ledgers. Distributed ledger is an evolution of ledger concept. It is a ledger maintained by two and more machines.

Robust distributed ledgers must comply with a set of requirements, of which the most important are persistence and liveness. These terms are based on the term of honest node – a node that acts in compliance with the protocol.

Persistence means that when an honest node declares some transaction as stable, all the other honest nodes also declare it as stable when queried. Persistence is presented as a predicate with parameter k. At first, persistence was created for distributed ledger based on blockchain. It meant that reaching depth of k blocks by transaction in a local copy of blockchain for honest node means that it occupies the same position in the same block in a local copy of blockchain for any other honest node. As robust distributed ledgers could be built on systems without the concept of block, later on this term has been generalized [13].

Liveness is a second feature of a robust distributed ledger, which preserves its robustness. It means guaranteed inclusion of honest user's transaction into ledger in an acceptable period (predefined number of time slots). In other words, liveness implies security against denial of including correct transactions into system by adversary's will. In application to blockchain, liveness implies guaranteed reaching of depth more than k blocks in a certain number of rounds by honest transaction.

For ledger controlled by one node, persistence and liveness are fulfilled by default. It is way more difficult to achieve these qualities for distributed ledgers functioning in an unreliable environment in the presence of adversaries operating against the protocol. Blockchain solves this exact problem.

3 Multidimensional Blockchain

Multidimensional blockchain is a system based on the concept of sidechains. Sidechain solutions have been created a while ago and are used mainly to transfer funds between independent cryptocurrencies. At first such operations were performed by observing the complete chain history in foreign blockchain. One of the first approaches to speed up this procedure was the use of nested hash chains (interlink) [15]. Instead of checking the complete chain history, only a chain of blocks with hash-sums less than $T/2^i$ is checked (T is a target hash-sum value for consensus mechanism). Thus, the complexity of verification is significantly reduced. In [16] the approach has been improved. The main advantage of the proposed solution was the possibility to verify a transaction with only one request to the target ledger. Moreover, several additional predicates that generalize the concept of verification have been proposed.

An approach to building proof-of-stake-based sidechains has been developed in [14]. That approach was compatible with the GHOST approach [22]. A formal definition of sidechain notion independent of consensus mechanism has also been presented.

The review of major modern pegged sidechain solutions has been undertaken in [3]. The solutions described have been based on cryptocurrencies and allowed temporary exchange of tokens. It implied freezing the tokens in one system and creating a corresponding number of tokens in a different chain. Also, almost all solutions under review represented applications and lacked security analysis.

In general, pegged sidechains imply following sequence of operations:

1. A user willing to use funds in sidechain sends them to a special address in his blockchain and proves the fact of sending to sidechain.
2. Exchange of tokens happens in sidechain – the account of the user is credited with a certain number of tokens.
3. If reverse exchange is necessary, it is performed analogously: the user sends tokens to a pre-defined address, where these tokens are frozen, and in an original blockchain, a transaction is created to return funds.

Multidimensional blockchain generalizes the concept of sidechains. It is a system consisting of a set of blockchains, where each blockchain, but the first one, follows the procedure of registration in one of the existing blockchains. Registration means storing information on genesis-block and, in some cases, information about some features of the new blockchain in some other blockchains.

The result system is in fact a tree of blocks. At the same time, many consensus algorithms permit temporary forks that lead to the existence of several syntactically correct chains in one blockchain – when correct and approved chain is only one of them. Taking this fact into account to avoid vagueness of term, the name “multidimensional blockchain” has been selected.

Two ways of building a multidimensional blockchain are possible: block mode and state mode. Figure 2 shows a general view of multidimensional blockchain,

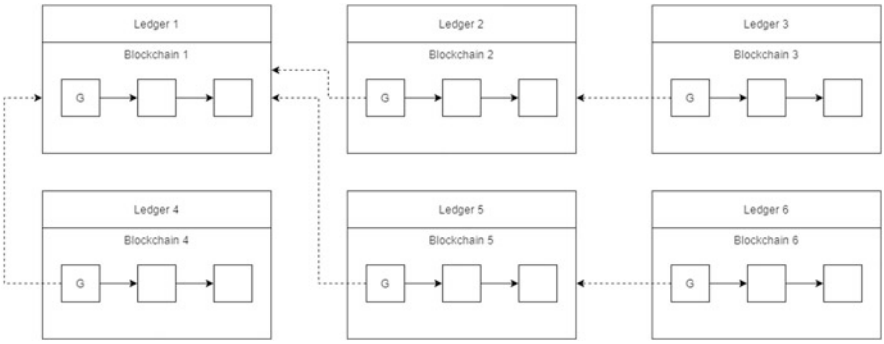

Fig. 2 Multidimensional blockchain

which unites several blockchains into one system. It is supposed that every blockchain implements robust distributed ledger – this assumption allows to disengage from a concrete operation mode. Therefore, it is not explicitly mentioned how exactly blockchain registration is performed: in block of special type or in internal data structure.

Block mode requires the creation of blocks of special type – for registration. State mode is based on the concept of state-transition machine developed in Ethereum white paper by G. Wood. It has been extended to represent a multidimensional blockchain. Consider the multidimensional blockchain mathematical model. As blockchains create new states at different rates, the following ratios assume that transactions were created within a fixed length of time, a slot. For the most correct statement of the mathematical model, the following relation can be taken:

$$\begin{aligned} T^{(k)} &= \left(T^{(k,1)}, \cdots, T^{(k,j)}\right) \left| j = \left[\frac{\text{Time}\left(\sigma_t^{(k)} \to \sigma_{t+1}^{(k)}\right)}{\text{sl}}\right] \right. \\ \text{sl} &\equiv \text{GCD}\ (\text{Time}\ (\sigma_t \to \sigma_{t+1})), T^{(k,j)} = \left(T_0^{(k,j)}, \cdots, T_n^{(k,j)}\right) \end{aligned} \tag{1}$$

where $T^{(k,j)}$ is a transaction tuple in ledger k during slot j, sl is a time slot, GCD is a greater common divider function, Time is a function returning state transition duration, and σ is a state.

In other words, a slot is the largest period of time into which the time intervals necessary for the transition between states in all blockchains are completely divided. As a result, each transition between states in each blockchain occurs once in a fixed (integer) number of slots:

$$\Pi'\left(\sigma^{(k)}, T^{(k,j)}\right) = \begin{cases} \sigma^{(k)} \text{ if } T^{(k,j)} = \varnothing \\ \Omega\left(\text{Y}\left(\ldots \text{Y}\left(\text{Y}\left(\sigma^{(k)}, \text{T}_0^{(k,j)}\right), \ldots\right) T_n^{(k,j)}\right)\right) \text{ otherwise} \end{cases} \tag{2}$$

where Π' is a modified block-level state transition function, Y is a state transition function, and Ω is a finalizing function responsible for consensus mechanism. In general, a multidimensional blockchain can be represented as follows:

$$\Sigma_{i+1} \equiv \Phi\left(\Sigma_i, T\right) \mid \Sigma_i \equiv \left\{\sigma^{(1)}, \ldots, \sigma^{(N)}\right\} \wedge \Phi\left(\Sigma_i, T\right) \equiv \Psi\left(\mathrm{P}\left(\Sigma_i, T\right), T\right) \tag{3}$$

$$\mathrm{P}\left(\Sigma_i, T\right) = \mathrm{E}\left(\mathrm{E}\left(\ldots \mathrm{E}\left(\Sigma_i, T, 1\right), \ldots\right) T, N\right) \mid \mathrm{E}\left(\Sigma_i, T, k\right) = \mathrm{E}'\left(\Pi'^{\left(\sigma^{(k)}, T\right)}\right), \tag{4}$$

where Ψ creates new blockchains, P is a state transition function, E is a state transition function for the k-th blockchain in its composition, and E' is an auxiliary function that returns a multidimensional blockchain for a one-dimensional blockchain and is used to avoid using the universal quantifier in mathematical notation. Finally, let us define the relationship between the state transition functions of ledgers:

$$\Pi\left(\sigma^{(k)}, T^{(k)}\right) = \Pi'\left(\Pi'\left(\ldots \Pi'\left(\sigma^{(k)}, T^{(k,1)}\right), \ldots\right), T^{(k,j)}\right)\right) \tag{5}$$

A key feature of a multidimensional blockchain is addressing, which directly affects the order in which applications are built. Within the framework of a multidimensional blockchain, user accounts, transactions, blocks, blockchains, and nodes supporting the system are subject to addressing. Addressing is performed in hierarchical mode, and a special notation has been developed to distinguish the addressed entities. A deep representation of the addressing has been shown in [18].

It is necessary to outline the fact that every blockchain is still a one-dimensional list that can operate separately from the system and differs from general implementations only, thanks to typing or existence of registration storage – depending on the operation model. At least two ways of addressing exist:

- Absolute – in multidimensional blockchain.
- Relative – in the current blockchain.

Besides, if in any subsystem interaction with a large number of other subsystems is not intended, using full address for every block can become excessive – especially at late functioning stages and in blockchains that are situated deep in the nested structure. In this case, it is possible to use aliases that are known to all participants of current subnet.

In block mode, addressing of child blockchain is possible using a number or a hash sum of block that performs registration of child blockchain. Number means height of block in parent blockchain where the current blockchain is registered. Both approaches are identical, but using hash sum allows to avoid reading all the blockchain and works faster in general. Also, if several blockchains are registered

in one block, it is necessary to specify the blockchain registration number inside the block. Special designation is to be used in this case.

In state model, every blockchain is registered by placing genesis-block or its hash sum into another blockchain. Child blockchain can be referenced by its genesis-block hash sum. The uniqueness of hash sums is provided by the hashing algorithm in use – and it is chosen while designing a concrete implementation. Cryptographic hash algorithms guarantee the existence of collisions with negligible probability, which leads to practically guaranteed uniqueness of genesis-block hash sum throughout the multidimensional blockchain provided that the genesis-blocks are unique. Theoretical addressing can be built using non-unique hash sums, but this might lead to double-spent vulnerability. In case double-spent attacks are not actual, using non-unique hash sums is permitted.

The main feature of a multidimensional blockchain is the presence of external transactions. An external transaction is an ordered sequence of logically related write-and-read operations in two or more ledgers. The ledger in which the external transaction starts is called the initiator, and the ledgers that accept the transaction are called recipients or acceptors. An external transaction, respectively, consists of two phases – initiation and acceptance (reception). It is worth noting that any external transaction always has one initiator, but there can be several recipients.

Consider the algorithm for conducting an external transaction in a multidimensional blockchain (Fig. 3). For the correct acceptance of a transaction in the acceptor ledger, it must be present in the initiating ledger, and the transaction must not have been accepted before.

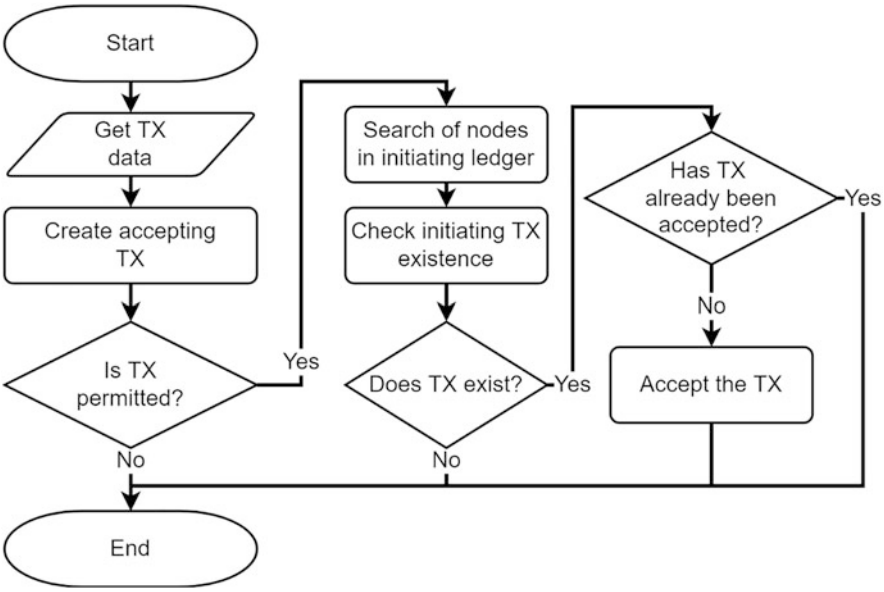

Fig. 3 Algorithm of accepting phase for external transaction

To sum up, multidimensional blockchain is a system based on the concept of one-dimensional blockchain and is meant to perform secure intersystem exchange and scaling of the systems based on distributed ledgers. The security is provided by the underlying data structure and a set of protocols for search and verification of external transactions.

4 Multidimensional Blockchain Security Analysis

The security analysis of multidimensional blockchain has been divided into several directions. First, it is necessary to show how the security of separate robust distributed ledgers is affected in case of scaling with multidimensional blockchain. This analysis is important as some security parameters might change, thanks to the change in relation between honest and adversarial nodes. Second, it is required to examine security of intersystem exchange organized with multidimensional blockchain. Finally, an analysis of scaling security must be performed to show that multidimensional blockchain implements robust distributed ledger.

4.1 Security Analysis of Underlying Robust Distributed Ledgers

The first security assessment of blockchain technology (e.g., internal consensus mechanism – proof of work) was presented in the first work on the first widespread cryptocurrency by Satoshi Nakamoto. When scaling using a multidimensional blockchain, the nodes that support the system are split into groups. As a result, the relation between the number of honest and adversarial nodes changes. Consequently, the parameters of the system change, which leads to a change in the probability of an attack. For a multidimensional blockchain, this probability takes the following form (modified version of probability calculated by Satoshi Nakamoto):

$$P_a = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - \left(\frac{q}{\frac{p}{N}}\right)^{(z-k)}\right), \tag{6}$$

where p and q are the probabilities of an adversarial and honest creating a block, respectively; N is the number of blockchains in a multidimensional blockchain; and z is the block depth for which the probability is calculated. An example of the probability of an attack on the last six blocks from the end of the chain is shown in Fig. 4.

The GHOST (Greedy Heaviest-Observed Subtree) approach has been developed by Zohar and Sompolinsky during their security assessment of Bitcoin and its

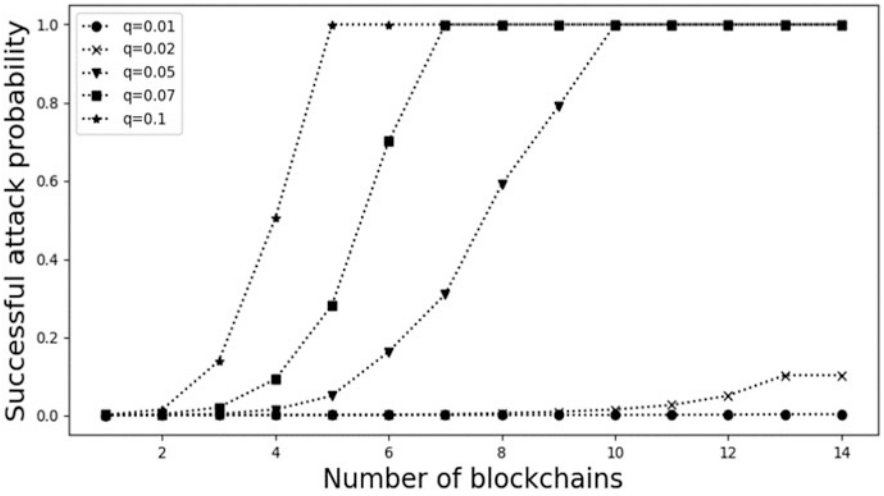

Fig. 4 Probability of a successful attack when the number of nodes is insufficient

underlying protocols [22]. It has been shown that standard chain selection rule is vulnerable to a potential attack of 25% of adversarial power. The novel approach implies placing in each block not only a hash-sum of a previous block but hash-sums of last blocks in recent forks. As a result, the discovered attack on Bitcoin and similar systems becomes impossible. The general safety condition is as follows:

$$\beta\left(\lambda_{\text{rep}}\right) \geq \frac{q}{1-q}\lambda_{\text{rep}} = \frac{q}{p}\lambda_{\text{rep}}, \tag{7}$$

where β is the block inclusion rate, λ_{rep} is the observed block creation rate, q and p are the probabilities that the next block is created by the attacker or honest node, respectively. Dividing miners into groups when creating a multidimensional blockchain (and when creating blockchains within a multidimensional blockchain) entails a change in the ratio of p and q, i.e., the probabilities of creating the next block by honest and attacking nodes. This leads to strengthening of the security requirement.

A more complete analysis of the proof of work has been presented by the IOHK company. The main requirement for the model is to comply with the requirements of honest majority:

$$t \leq (1-\delta)(n-t), \quad \delta \geq 2f + 2\epsilon, \tag{8}$$

where t is the number of compromised nodes, n is the total number of nodes, f is the expected number of new blocks created in each round, and ϵ is a negligible number. When the blockchain is split into independent blockchains inside multidimensional blockchain, the number of honest nodes decreases, which leads to a decrease in the

parameter f. Consequently, the lower bound of the parameter δ decreases, which entails the strengthening of the requirement for an honest majority.

The security of a proof-of-stake system does not depend on the number of nodes maintaining it and is determined by the number of accounts in the system and the ratio between the shares of honest and attacker accounts. Therefore, the theorems introduced in the articles about Ouroboros remain correct for a multidimensional blockchain under the only condition – the creation of a genesis block with a fair majority when registering a new blockchain.

A more deep and thorough analysis has been presented in [19].

4.2 Intersystem Exchange Security Analysis

To prove the security of a multidimensional blockchain, it is required to show that it does not break the security of internal robust distributed ledgers with the novel functionality of external transactions. In other words, it is necessary to show that the intersystem exchange is secure. To achieve this goal, a generalized universal composition framework (GUC-framework) is used. It involves representing the system in the form of a set of interacting interactive Turing machines and proving either of the following:

1. For any adversarial node attacking the target system (ideal functionality), there exists a simulator attaching the constructed protocol such that it is impossible for an environment (external observer) to distinguish the executions (in probabilistic sense).
2. There exists a sequence of equivalent hybrid models from the target system model to the constructed protocol model. A hybrid model incorporates parts of both target and constructed systems.
3. The probability of bad events that might break some security requirements is negligible in probabilistic sense (in this case, the GUC model is used to perform formalization).

A deep description of the GUC framework has been presented in [7, 8] and some other papers in which the security analysis has been performed with the help of it [9, 10]. It is worth mentioning that the models of robust distributed ledgers have been presented in the literature before [2, 4, 12]. However, these functionalities have not implemented external transactions. Thus, a novel robust distributed ledger model has been built to prove the security of multidimensional blockchain. It supports external transaction functionality and is constructed as close to the pre-invented models as possible. Also, a model for a protocol implementing robust distributed ledger has been created. In addition, an auxiliary ideal functionality for searching and verifying external transactions was proposed.

Both models – of ideal functionality implementing robust distributed ledger and corresponding protocol – are presented in Fig. 5. These models include the following parties: G_{CLOCK} is a timing ideal functionality, G_{VERIFY} is an ideal func-

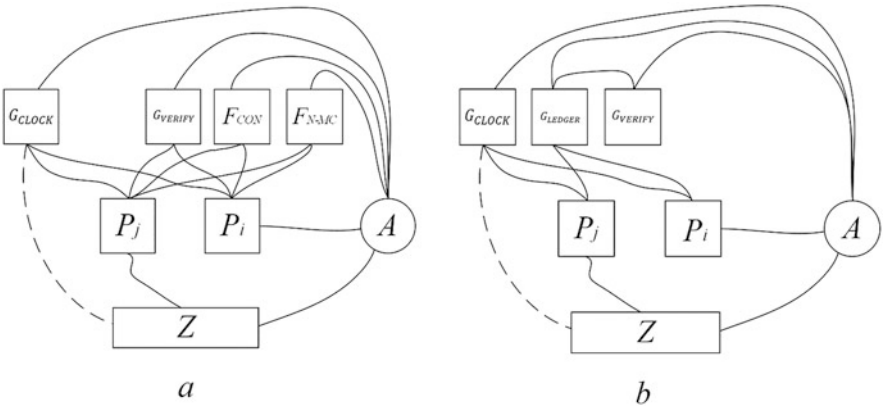

Fig. 5 GUC-model of protocol implementing robust distributed ledger (**a**) and GUC-model of robust distributed ledger (**b**)

tionality used for search and verification, F_{CON} is consensus mechanism, $F_{N-\text{MC}}$ is multicast medium, G_{LEDGER} is robust distributed ledger (which implements external transactions), A is adversary, Z is environment, and P_i are the parties running the model.

With these models, the following propositions have been proven:

1. The robust distributed ledger model is compatible with previously proposed models.
2. The properties of persistence and liveness are not violated when using the ideal search and verification functionality for blocks and transactions with a probability proportional to the probability of a fork at depth k.

The first proposition has been proven by comparison of the execution models. It is sufficient to show that these have no differences but external transactions [21]. The proof leads to the proof of the second proposition. The second proof is based on the examination of possible "bad" events that might break the properties of persistence and liveness (the proof is shown as it has not been presented before):

- *BAD1* – breaking liveness of the initiating ledger. This event is impossible, thanks to the way the external transactions are performed (the initiating phase is equal to ordinary transaction).
- *BAD2* – breaking liveness of accepting ledger. This event is impossible because ideal functionality provides guaranteed verification in case the verification period exceeds the provided time window. For this to happen, the following relation must hold:

$$\text{window} \times t_{\text{sl}} - 1 \geq \max\{d\} \times 2 \times \max\{t_v\}, \tag{9}$$

where window is a window size in slots, t_{sl} is a slot duration, d is the maximum ledger depth, and t_v is the time of interaction with ledger during search or verification.

- *BAD3* – breaking persistence of the initiating ledger. This event is impossible, thanks to the way the external transactions are performed (the initiating phase is equal to ordinary transaction).
- *BAD4* – breaking persistence of the accepting ledger. As the response on verification of external transactions is delayed, the only way to break persistence is to apply transaction to the ledger and to revert it in the initiating ledger. All the ledgers in multidimensional blockchain are robust by assumption. Thus this situation is possible only when the transaction is reverted before going deep enough in the chain of blocks. Let $p^{(k)}$ be the probability of fork at depth k. Then the probability of acknowledgement is as follows:

$$p = p^{(k)} \times \frac{\sum \theta_i^H + \gamma \sum \theta_i^A}{|H| + |\ A\ |} \tag{10}$$

In the worst case, the nodes are split into two equally sized groups such that in one of them, there are all the adversarial nodes and sufficient number of honest nodes. Then the adversary has a maximum chance of reverting transaction:

$$\begin{cases} \sum \theta_i^A = |\ A\ | \\ \sum \theta_i^H = 0,5 \times (|H| + |A|) - |A| = 0,5\,(|H| - |A|) \end{cases} \Rightarrow p = p^{(k)} \times 0,5 \tag{11}$$

Finally, for a protocol that is executed by nodes supporting a multidimensional blockchain, the GUC-model has also been proposed:

A proposition has been proven that this protocol GUC-implements the ideal functionality of a multidimensional blockchain. To prove this, it is enough to show that the execution of this protocol is equivalent to the execution of the multidimensional blockchain GUC-model, because in this case, the universal composition theorem will be applicable. The proof is based on hybrid models, when each next model differs from the previous one but remains equivalent to it (the proof is shown as it has not been presented before):

- HYB0 is a multidimensional blockchain model. All nodes use queries to the multidimensional blockchain to work. In fact they act like "dummy" parties that only pass queries in an appropriate format to an ideal functionality.
- HYB1 is a model in which nodes independently handle addressing actions, i.e., determine source and destination ledgers for each external transaction. Instead of one external transaction, they redirect two internal transactions (outgoing and incoming) to the multidimensional blockchain. HYB1 is equivalent to HYB0, because the way the model uses multidimensional blockchain does not change: external transactions are simply divided in advance.

- HYB2 is a model in which nodes perform notifications on all external transactions: when a new transaction is created, a notification is sent to the nodes that maintain the target ledger. Then they independently send a request to the multidimensional blockchain. The difference from HYB1 is only in the origin of the second (incoming) transaction, because the transmission of the notification takes negligible time in the scale of the slot time.
- HYB3 is a model in which nodes independently verify an external transaction and send a request to add an incoming transaction only if it is correct. For this, the ideal search and verification functionality is used, which is guaranteed to carry out the verification correctly. The same functionality is used inside multidimensional blockchain ideal functionality. Because no changes have been made to the functionality of the multidimensional blockchain, this model is equivalent to HYB2.
- HYB4 is a model in which nodes independently carry out verification of incoming external transactions using a search and verification protocol, which must GUC-implement an ideal search and verification protocol. According to the universal composition theorem, this model is equivalent to HYB3 with a probability determined by the probability of successful verification.
- HYB5 is a model in which the multidimensional blockchain is replaced by many one-dimensional blockchains. Because verification is guaranteed (subject to the constraints of the GUC-implementation), this model is equivalent to HYB4.

It can be seen that HYB5 is the same model as that given in Fig. 6. In other words, this model is actually a simulated protocol that implements a multidimensional blockchain (MBC-Protocol). Thus the MBC-Protocol GUC-implements ideal multidimensional blockchain functionality and can be used in models instead of this functionality and vice versa.

4.3 Scaling Security Analysis

Yet multidimensional blockchain has been invented to perform intersystem exchange; it also solves the problem of scaling robust distributed ledgers. To prove the security of scaling, it is sufficient to show that multidimensional blockchain GUC-implements robust distributed ledger. In this case a one-dimensional blockchain GUC-implementing robust distributed ledger can be replaced by multidimensional blockchain.

To build such a proof, a simulating approach was used: it has been proven that for any node attacking a multidimensional blockchain, there exists a simulator attacking the ideal functionality of a robust distributed ledger that, from the side of the environment, the two executions are identical. A schematic representation of the simulation model is shown in Fig. 7.

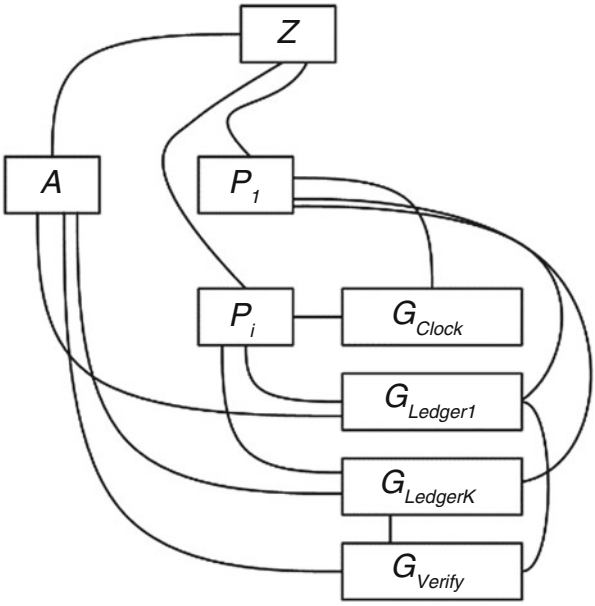

Fig. 6 GUC-model of a protocol implementing multidimensional blockchain (MBC-Protocol)

5 Search and Verification Protocol

In [20] several approaches to building search and verification protocol for blocks and transactions have been presented. The following conclusions were obtained:

1. A centralized search and verification protocol is equivalent to ideal functionality, provided that the node supporting the protocol is honest.
2. The search and verification protocol for blocks and transactions built on the basis of a fully connected network interaction graph GUC-implements an ideal search and verification protocol for blocks and transactions with the probability specified in Relation (12).
3. The search and verification protocol for blocks and transactions, built on the basis of a fully connected graph of network interaction with the parent blockchain, GUC-implements an ideal search and verification protocol for blocks and transactions with the probability indicated in Relation (12).
4. The 1-to-1 connection approach is not secure and should not be used when building a search and verification protocol for blocks and transactions.
5. The approach with connecting subsets of neighboring ledgers is not secure and should not be used when building a search and verification protocol for blocks and transactions.

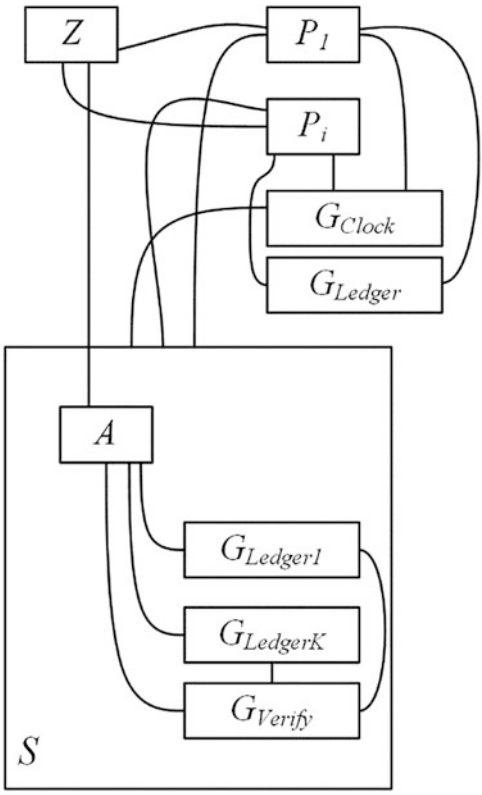

Fig. 7 GUC-model of a system with a simulator

$$P = \begin{cases} \sum_{i=0}^{\left[\frac{k}{2}\right]-1} C_k^i \times q^i \times p^{k-i}, \text{if } z = \left[\frac{k}{2}\right] - 1 \leq N_A \\ 1, \text{if } \left[\frac{k}{2}\right] - 1 > N_A \end{cases}, \tag{12}$$

where k is the number of polled nodes, p is the proportion of honest nodes, q is the proportion of attacking nodes, C is the number of combinations, and N_A is the number of attackers.

A robust search and verification protocol for blocks and transactions has been invented. This protocol makes it possible to guarantee the search for nodes of the target registry and, under certain conditions, to carry out verification faster than using a sidechain-based solution. The following version of the protocol is proposed:

1. Each blockchain node keeps track of the last $l + 2k$ blocks from each neighboring blockchain. k blocks are used to provide the common prefix property. l blocks correspond to the chain quality.
2. They are asked for the headers of $l + 2k$ blocks in the next blockchain and the addresses of the nodes that created them (identifiers and a network entry point for

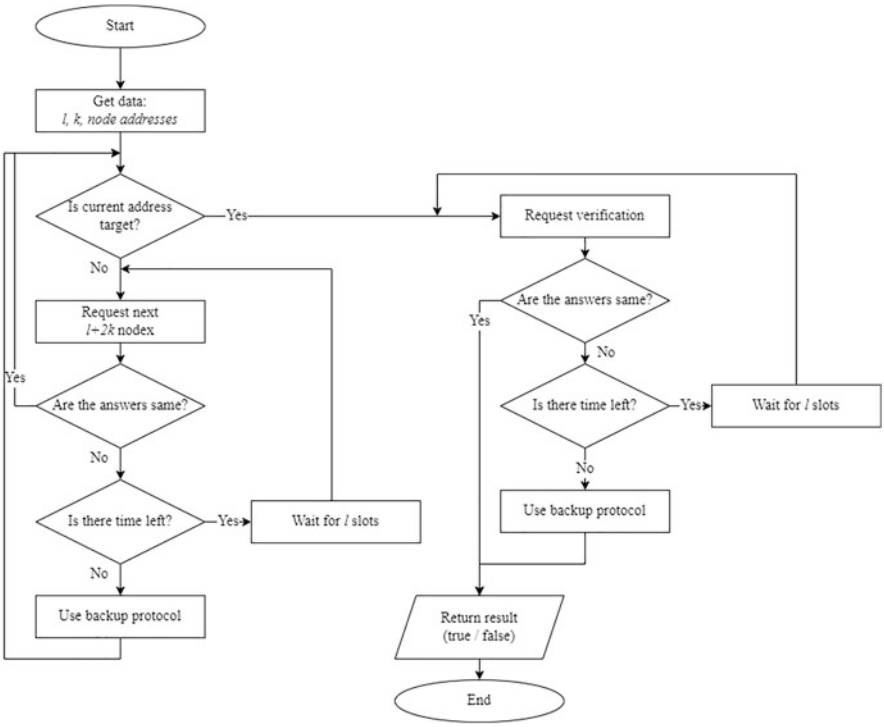

Fig. 8 Robust search and verification protocol for blocks and transactions

searching are allowed). The last k blocks are used to comply with the common prefix property, the next l blocks are used to enforce the purity property of the chain, and the last k blocks are needed for verification. Because there is at least one honest node among the nodes (by the CQP, because the length is greater than l), such a chain is guaranteed to exist. For the obtained blocks, the correctness of their construction is checked.

3. If the ledger is targeted, then go to the next step. Otherwise, select l nodes in the middle of the resulting chain, and go to step 1.
4. If the combined search and verification time exceeds the maximum allowable time in terms of liveness, perform verification using the backup functionality.
5. To verify the l found nodes (which created l blocks deeper than the last k) in the block chain, a chain of $l + 2k$ blocks is requested.
6. If the $l + k$ first blocks are the same among all the received results, perform verification by requesting the block containing the outgoing transaction and the chain of headers from this block to the first among $l + k$ received earlier. Otherwise, skip l slots and go to step 1.

The algorithm is shown in Fig. 8.

Using the universal composition framework, the following propositions have been proven:

1. The robust search and verification protocol makes it possible to correctly verify an external transaction with a probability close to 1, with the right choice of the set of polled nodes.
2. The robust search and verification protocol based on the chain quality property implements the ideal search and verification functionality with the probability of maintaining the chain quality by the blockchains included in the multidimensional blockchain.

By the definition of a multidimensional blockchain, all ledgers within it are stable. A ledger is stable if it meets the Chain Growth (CGP), Common Prefix (CPP), and Chain Purity (CQP) requirements. Consider possible attacks on the protocol by an attacker.

Event 1: The attacker forms a completely fabricated chain of length $l + 2k$. This event can only occur if the purity property of the chain is violated. Therefore, the probability of this event in the worst case is as follows:

1. $p_1 = 1 - (1 - e^{-\Omega(\kappa)}) = e^{-\Omega(\kappa)}$ for proof of work (this upper bound is less than other from analogous works, e.g., [12]).
2. $p_1 = 1 - \left(1 - e^{-\Omega(\sqrt{l+k})+\ln R}\right) = e^{-\Omega(\sqrt{l+k})+\ln R}$ for proof-of-stake (by Theorem 4.13 of [17]).

Event 2: All polled nodes are attackers. This event can only occur if the chain quality property of the chain is violated. Therefore, the probability of this event in the worst case is as follows:

1. $p_2 = 1 - (1 - e^{-\Omega(\kappa)}) = e^{-\Omega(\kappa)}$ for proof of work (this upper bound is less than other from analogous works, e.g., [12]).
2. $p_2 = 1 - \left(1 - e^{-\Omega(\sqrt{l+k})+\ln R}\right) = e^{-\Omega(\sqrt{l+k})+\ln R}$ or proof of stake (by Theorem 4.13 of [17]).

Event 3: The attacker generates an alternative chain of blocks when requesting information from block N to block N-l-$2k$. This event is determined by the probability of finding the first preimage for the hash sum used in the blockchain. In the worst case, this probability is equal to $p_3 = \left(\frac{1}{2^\kappa}\right)^{l+2k}$.

Event 4: Two honest nodes provide different responses to the query. This situation can only occur if the common prefix property is violated. Because nodes that created blocks at depth k are used for interaction, the probability of this event is as follows:

1. $p_4 = 1 - \left(1 - e^{-\Omega^\kappa}\right) = e^{-\Omega(\kappa)}$ for proof of work (this upper bound is less than other from analogous works, e.g., [12]).
2. $p_4 = 1 - (1 - e^{-\Omega(\kappa) + \ln R}) = e^{-\Omega(\kappa) + \ln R}$ for proof of stake.

Event 5: The attacker does not provide information when requested. This event causes the fallback protocol to be used and therefore does not compromise system security. In this case, in the worst case, the probability of an event depends on the probability of the presence of at least one attacker (in the worst case, *0.5l*) and is therefore not negligible.

Therefore, the probability of violating the information security and robustness properties of a distributed ledger for proof of work and proof of stake, respectively, is as follows:

$$P_{\text{POW}} = p_1 + p_2 + p_3 + p_4 = 3 \times \mathrm{e}^{-\Omega(\kappa)} + \left(\frac{1}{2^{\kappa}}\right)^{l+2k} \approx \varepsilon \tag{13}$$

$$P_{\text{POS}} = p_1 + p_2 + p_3 + p_4 = 2 \times \mathrm{e}^{-\Omega\left(\sqrt{l+k}\right)+\ln R} + \left(\frac{1}{2^{\kappa}}\right)^{l+2k} + e^{-\Omega(\kappa)+\ln R} \approx \varepsilon \tag{14}$$

The second proposition is proven with the help of hybrid models:

- HYB0 is the original model; external interactions are carried out using the ideal functionality to validate external transactions.
- HYB1 is a model in which the nodes themselves provide work with the search for ledgers for interaction. However, all ledgers still notify the ideal functionality about external transactions. As a result, each node using the search protocol can be guaranteed to discover a subset of the initiating ledger nodes, i.e., the search is performed independently, while verification is still performed using ideal functionality. Since the search is carried out correctly with a probability close to 1, for an external observer, this model is equivalent to HYB0.
- HYB2 – separation of transaction validation logic. Instead of ideal functionality, a wrapper is used that executes a set of ideal functionalities within itself, each of which is passed requests related to only one ledger. External interfaces do not change, so the model is equivalent to HYB1.
- HYB3 – wrapper elimination. Ledgers interact independently with ideal functionalities. Information about which ideal functionality to request verification from is requested from the nodes found through the search protocol. The search and verification protocol searches with the probability of respecting the chain quality and common prefix properties. According to the previous proposition, this probability is close to 1.
- HYB4 – requesting information directly from nodes. Similarly to HYB3, information is requested from the nodes; however, it is information about the correctness of the transaction that is requested. The probability of correct verification remains the same, because honest nodes follow the protocol and correctly verify the transaction. This model is equivalent to a search and verification protocol for blocks and transactions.

6 Theoretical and Experimental Analysis

Multidimensional blockchain has several advantages over conventional systems. This section covers them. Consider saving memory by a separate node of a computer network when replacing a one-dimensional blockchain with a multidimensional analogue. Let the source ledger be divided into N_L ledgers. If the transaction generation period (frequency) or the block size decreases, then the average value of the amount of information stored by the nodes at any given time is as follows:

$$\overline{\mathrm{LV}} = \frac{\sum_i \mathrm{LV}^{(i)}}{N_L} = \frac{LV \times \sum_i p_i}{N_L} = \frac{LV}{N_L{}'} \tag{15}$$

where N_L is the number of registries, and p_i is the number of accounts transferred to the new blockchain. Figure 9 shows comparison of the volume size growth for the different numbers of ledgers inside the multidimensional blockchain.

Another important feature of any ledger is the number of transactions per second (TPS). Figure 10 shows an estimate of the TPS. In each unit of time, the number of blockchains increases by 1. Graphs 1 and 2 reflect the increase in the number of transactions per unit of time in the system as a whole with a constant increase in the number of transactions and without growth, respectively. Graphs 3 and 4 reflect the reduction in the load on each blockchain separately under the same conditions. A more comprehensive analysis has been presented in [18].

In order to verify the applicability of multidimensional blockchain, an experimental analysis has been conducted. For this, a prototype that implements a multidimensional blockchain was applied. It implemented a simple token-based system with accounts associated with key pairs and balances.

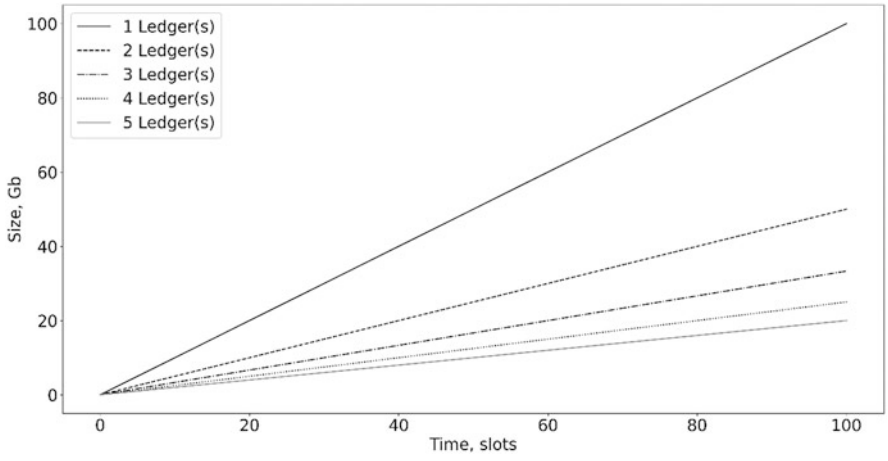

Fig. 9 Volume of one node for the different numbers of blockchains

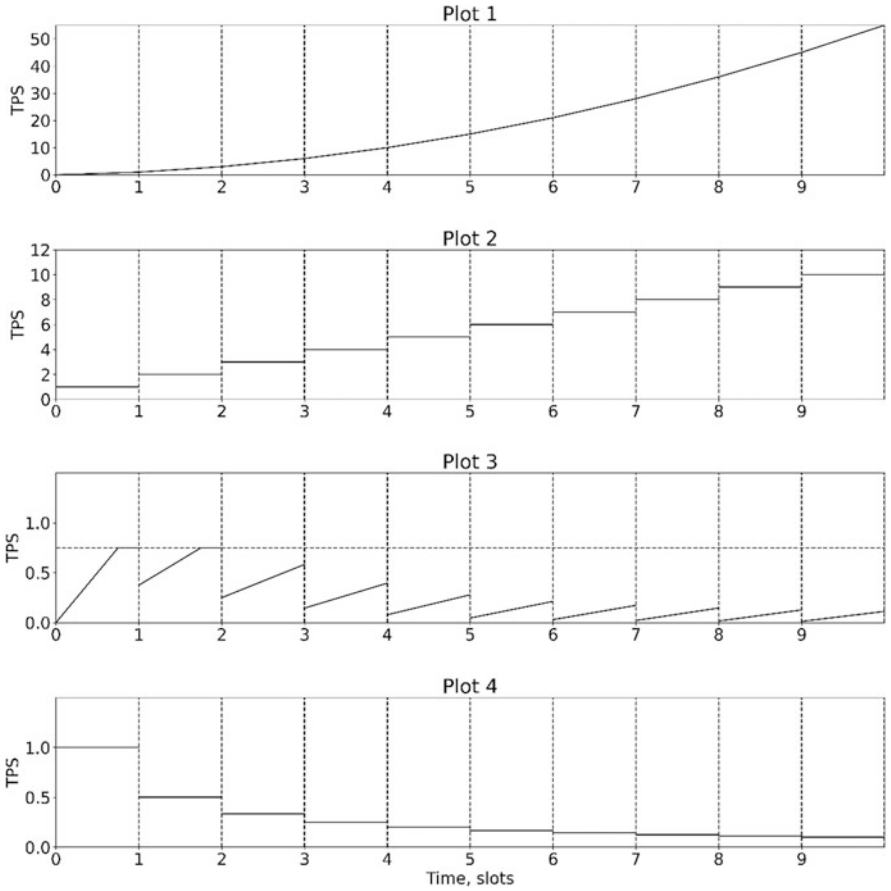

Fig. 10 TPS in multidimensional blockchain

A series of experiments were conducted to test the properties of a multidimensional blockchain. As part of the first experiment, five independent robust ledgers were used, each of which was executed by five nodes in different parts of the world. It is necessary to admit that the system was based on simple centralized consensus mechanism (beacon service) in order to decrease the influence of consensus scheme peculiarities on the analysis. Also search and verification errors have been modeled by a probabilistic approach: a probability of unsuccessful search and verification has been introduced. The sequence of transactions has been generated programmatically, and for each ledger, a period of intensive transaction generation has been introduced.

The purpose of the first experiment was to identify the numerical characteristics of the system operation. All external transactions have been accepted. The average results for all ledgers are presented below (Table 1).

Table 1 Experimental results for external transaction delay (intersystem exchange)

Experiment	Delay	Delay of internal transactions, sec	Delay of external transactions verification, sec	Delay of external transactions applying, sec
No adversarial actions	Maximum	20.171	140.608	276.836
	Minimum	0.001	6.325	6.997
	Average	6.923	59.904	71.632
Adversarial actions	Maximum	20.009	260.179	339.509
	Minimum	0.001	6.303	6.82
	Average	5.042	65.716	80.989

Table 2 Experimental results for external transaction delay given the transaction rate growth (scaling)

Number of ledgers	Average transaction delay, sec	Maximum transaction delay, sec	Average storage load (number of TX stored)	TPS
1	4.97	10.15	313	0.21667
2	36.08	86.94	425.5	0.42083
3	15.57	80.32	350	0.52847
5	11.97	80.25	331	0.92847

Table 3 Experimental results for external transaction delay given the constant transaction rate (scaling)

Number of ledgers	Average transaction delay, sec	Maximum transaction delay, sec	Average storage load (number of transactions stored)	TPS
1	4.31	10.09	310	0.21806
2	36.14	80.38	229	0.21111
3	16.57	79.22	116.67	0.18194
5	12.02	80.09	67.4	0.1875

The second experiment was aimed at analyzing similar parameters under conditions of a targeted attack on the protocol by 10% of attackers (without using a backup protocol). All the other characteristics were left untouched. The results are presented in Table 2.

For scaling, an experimental test was also carried out using four models consisting of one, two, three, and five blockchains, respectively (the total number of nodes is unchanged). At the same time, a situation was considered in which the load on each independent blockchain is higher (the flow of transactions increases in proportion to the increase in the number of ledgers) or remains unchanged (Table 3).

Based on the experimental results, the following conclusions can be made:

1. Multidimensional blockchain allows to perform secure intersystem exchange with acceptable (under certain circumstances) average transaction delay.
2. Multidimensional blockchain allows to increase the system throughput.
3. Multidimensional blockchain allows to decrease requirements for nodes.
4. The split into multidimensional blockchain containing two ledgers leads to the opposite effect: the size of blockchain volume increases for all nodes thanks to the two-phase structure of external transactions.

7 Conclusion

This chapter covers the recent advances in the sphere of constructing multidimensional blockchain. The technology is briefly described, and its peculiarities are highlighted. Several statements on multidimensional blockchain security have been proven. Some proofs have been presented for the first time. Finally, experimental analysis of multidimensional blockchain functioning has been presented for the first time.

In general, multidimensional blockchain allows solving the problem of scaling robust distributed ledgers and the problem of secure exchange between independent robust distributed ledgers. The research and its results described in this chapter are of interest to developers of decentralized and distributed technologies and applications, as well as researchers involved in the problem of secure intersystem interaction and questions of building distributed technologies.

As prospects for further development, we can point out the improvement of the proposed search and verification protocol for blocks and transactions in order to increase the likelihood of its successful operation in the face of attacks from malicious network nodes. In addition, it is of interest to introduce zero-knowledge cryptographic methods into the process of conducting external transactions to ensure the confidentiality of transactional information. Finally, it is possible to search for new areas for applying the proposed methods and algorithms and adapt them accordingly.

References

1. A. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies* (O'Reilly Media, Inc., Sebastopol, 2014)
2. C. Badertscher, P. Gaži, A. Kiayias, A. Russell, V. Zikas, Ouroboros genesis: composable proof-of-stake blockchains with dynamic availability, in *ACM Conference on Computer and Communications Security – ACM CCS 2018* (2018), pp. 913–930
3. A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timon, P. Wuille, Enabling blockchain innovations with pegged sidechains. https://blockstream.com/sidechains.pdf. Retrieved March, 2022

4. C. Badertscher, U. Maurer, D. Tschudi, V. Zikas, Bitcoin as a transaction ledger: a composable treatment, in *Advances in Cryptology – CRYPTO 2017, LNCS*, vol. 10401, (Springer, Cham, 2017), pp. 324–356
5. I. Bentov, A. Gabizon, A. Mizrahi, Cryptocurrencies without proof of work, in *Financial Cryptography and Data Security, FC 2016, LNCS*, vol. 9604, (Springer, Berlin, Heidelberg, 2016), pp. 142–157
6. C. Cachin, R. Guerraoui, L. Rodrigues, *Introduction to Reliable and Secure Distributed Programming* (Springer-Verlag, Berlin, Heidelberg, 2011)
7. R. Canetti, Universally composable security: a new paradigm for cryptographic protocols, in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, Newport Beach (2001), pp. 136–145
8. R. Canetti, Y. Dodis, R. Pass, S. Walfish, Universally composable security with global setup, in *Theory of Cryptography, TCC 2007, LNCS*, vol. 4392, (Springer, Berlin, Heidelberg, 2007), pp. 61–85
9. R. Canetti, D. Shahaf, M. Vald, Universally composable authentication and key-exchange with global PKI, in *Public-Key Cryptography – PKC 2016, PKC 2016, LNCS*, vol. 9615, (Springer, Berlin, Heidelberg, 2016), pp. 265–296
10. B. David, R. Dowsley, M. Larangeira, ROYALE: a framework for universally composable card games with financial rewards and penalties enforcement, in *Financial Cryptography and Data Security, FC 2019, LNCS*, vol. 11598, (Springer, Cham, 2019), pp. 282–300
11. B. David, P. Gaži, A. Kiayias, A. Russell, Ouroboros Praos: an adaptively-secure, semi-synchronous proof-of-stake blockchain, in *Advances in Cryptology – EUROCRYPT 2018, LNCS*, vol. 10821, (Springer, Berlin, Heidelberg, 2018), pp. 66–98
12. J. Garay, A. Kiayias, N. Leonardos, The bitcoin backbone protocol: analysis and applications, in *Advances in Cryptology - EUROCRYPT 2015, LNCS*, vol. 9057, (Springer, Berlin, Heidelberg, 2015), pp. 281–310
13. J. Garay, A. Kiayias, N. Leonardos, The bitcoin backbone protocol with chains of variable difficulty, in *Advances in Cryptology – CRYPTO 2017, LNCS*, vol. 10401, (Springer, Cham, 2017), pp. 291–323
14. P. Gazi, A. Kiayias, D. Zindros, Proof-of-stake sidechains, in *2019 IEEEE Symposium on Security and Privacy (SP)*, vol. 1 (2019), pp. 677–694
15. A. Kiayias, N. Lamprou, A. Stouka, Proofs of proofs of work with sublinear complexity, in *Financial Cryptography and Data Security*, vol. 9604, (Springer, Cham, 2016), pp. 61–78
16. A. Kiayias, A. Miller, D. Zindros, Non-interactive proofs of proof-of-work, in *Financial Cryptography and Data Security*, vol. 12059, (Springer, Cham, 2020), pp. 505–522
17. A. Kiayias, A. Russell, B. David, R. Oliynykov, Ouroboros: A provably secure proof-of-stake blockchain protocol, in *Advances in Cryptology – CRYPTO 2017, LNCS*, vol. 10401, (Springer, Cham, 2017), pp. 357–388
18. I. Shilov, D. Zakoldaev, Multidimensional blockchain and its advantages. Inf. Technol. **26**(6), 360–367 (2020)
19. I. Shilov, D. Zakoldaev, Multidimensional blockchain security analysis. Lect. Notes Netw. Syst. **235**, 911–924 (2022)
20. I. Shilov, D. Zakoldaev, Security of search and verification protocol in multidimensional blockchain. Inform. Autom. **20**(4), 793–819 (2021)
21. I. Shilov, D. Zakoldaev, The robust distributed ledger model for a multidimensional blockchain security analysis. Sci. Tech. J. Inf. Technol. Mech. Opt. **132**(2), 249–255 (2021)
22. Y. Sompolinsky, A. Zohar, Accelerating bitcoin's transaction processing fast money grows on trees, not chains. IACR Cryptology ePrint Archive (2013)