Paper

Blockchain Networks – Security Aspects and Consensus Models

Andrzej Wilczyński[1,2] and Adrian Widłak[2]

[1] *AGH University of Science and Technology, Cracow, Poland*
[2] *Tadeusz Kosciuszko Cracow University of Technology, Cracow, Poland*

https://doi.org/10.26636/jtit.2019.132019

Abstract—Data integration and fast effective data processing are the primary challenges in today's high-performance computing systems used for Big Data processing and analysis in practical scenarios. Blockchain (BC) is a hot, modern technology that ensures high security of data processes stored in highly distributed networks and ICT infrastructures. BC enables secure data transfers in distributed systems without the need for all operations and processes in the network to be initiated and monitored by any central authority (system manager). This paper presents the background of a generic architectural model of a BC system and explains the concept behind the consensus models used in BC transactions. Security is the main aspect of all defined operations and BC nodes. The paper presents also specific BC use cases to illustrate the performance of the system in practical scenarios.

Keywords—blocks, cryptography, ledger, proof of work.

1. Introduction

Over the few past years, the Blockchain (BC) became the topic of interest for many engineers and companies, especially from financial and ICT sectors. This makes BC one of the most popular technologies used in ICT infrastructures developed for the needs of public institutions, financial markets, cloud storage systems and many other domains [1]. BC may be defined as a decentralized computer network without a central management unit. Data stored in BC blocks within such a system may be efficiently protected against external attacks. Data in a given block cannot be modified without an additional, significant power supply for the ICT infrastructure, which is usually not provided (it would rapidly increase the cost of energy used in BC nodes). In BC networks, the extra supervised transactions are not necessary (no central authority), each node is autonomous and may take decisions about transactions based on consensus procedures. This prevents any data manipulation and intrusions aimed at impersonating entities and performing unauthorized operations. Such consensus models define crucial procedures of the process of creating the chain of BC blocks and data transactions.

In this paper, the backgrounds of the BC architectural model and the consensus procedures are presented, and security-related issues affecting the entire BC system and the users' actions are illustrated. Unlike in existing papers and other publications concerned with BC essentials [2], [3], the BC system is shown from the ICT and engineering perspective, where the BC network may be applied as a potential supportive technology used for data and task processing in HPC computing environments (such as clouds, grids, fogs, etc.) Based on the authors' experience with BC technology, the practical scenario BC use cases are demonstrated.

The rest of the paper is organized as follows. In Section 2, the background of BC architecture is defined, and some most important security issues are presented. Section 3 presents the proof of work, proof of stake and round robin consensus models. In Section 4 the main use cases of BC are specified. The paper ends with a short summary given in Section 5.

2. Blockchain Backgrounds and Security Aspects

Pursuant to the most popular definition of a BC system, Blockchain is a distributed ledger of records in which data transactions and other system information may be specified. From the technological point of view, BC may be defined as a technological protocol that allows the exchange of data between different users in a network (usually external- or end-users) without the need for intermediaries [4]. The following characteristic properties of BC technology may be distinguished:

- **no central authority** – there is no need for a central system manager that decides whether any operation in BC is performed in accordance with the accepted rules or regulations,
- **immutability** – the transactions saved in a chain of blocks cannot be modified, which guarantees the immutability of data stored,
- **security** – cryptographic methods are used in the consensus models and for the protection of transactions,
- **transparency** – resources and transactions of each public address are available for viewing by anyone with access to BC,

- **efficiency and higher speed** – traditional processes of concluding transactions confirmed by a central system manager are time-consuming and may fail easily because of human errors – in BC, the confirmation of a transaction is automatic, which makes the whole process much faster,
- **cost reduction** – Blockchain excludes the involvement of external parties or intermediaries in the process to provide guarantees, which may lead to a reduction in maintenance costs, for instance no need to employ personnel operating and controlling the processes of accounting for money in banks.

2.1. Blockchain Architecture

Blockchain architectural models may be classified into the following network categories: private, public and permissioned.
Private BC networks have an owner (usually a company, the society or a public entity) that decides about the access to BC nodes and data. Private BCs are usually non-decentralized networks, however cryptographic protocols are used to secure the transactions. The most popular example of such systems is Multichain BC [5].
In public BC networks, any external user is capable of reading and easily modifying the ledger of records. The most popular examples of such systems are Bitcoin [6] and Litecoin [7]. In permissioned BC networks, there is a consortium of users or a privileged user who may grant the next node the permission to write or read from the block or blocks after verification of identity. The popular example of such a system are R3 (banks) [8] or EWF (energy) [9].

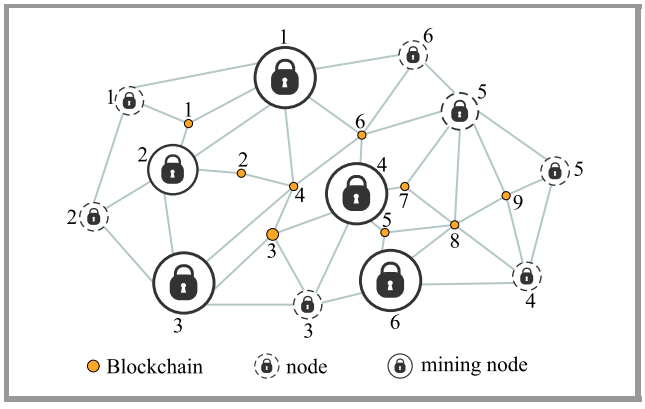

Fig. 1. Blockchain network.

Figure 1 shows an example of a BC network. One may observe that the nodes in a BC network are not always connected with all remaining nodes. There is no central unit and it is possible to connect an external, additional node to the existing network at any time. Hence, the model is very dynamic. The network consists of nodes confirming transactions (establishing the consensus), mining nodes responsible for adding blocks to BC, and users who have addresses and who upload data which are then placed in transactions.

2.2. Blocks and Merkle Tree

BC transactions may be defined by a list of the following attributes:

- transaction identifier – ID,
- transaction sender,
- transaction recipient,
- digital transaction signature,
- transaction data.

Each transaction must be approved by the majority of BC node administrators (users), usually at least 50% of the entire network.

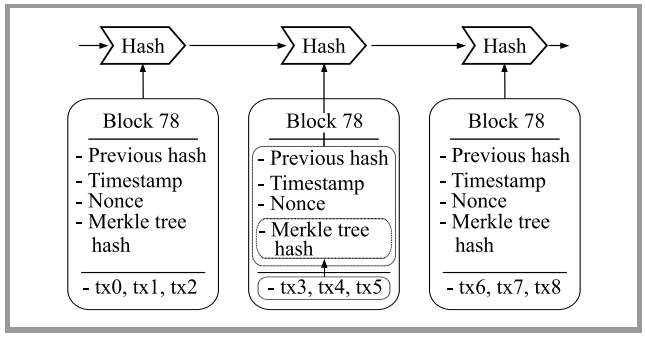

Fig. 2. Abstract model of BC blocks.

All parameters of the approved transaction are added to the block. The block is the main module of any BC node. The number of transactions in a given block (the block volume or block capacity) is defined depending on the standards defined for the entire BC system. An abstract model of a BC block is presented in Fig. 2. Each block is defined by the following components:

- block number,
- hash of current block,
- hash of previous block,
- timestamp,
- nonce – this is the number sought by the mining node, its finding usually consists in the solution of the hash function and makes it possible to add a block to the blockchain,
- the Merkle tree hash (calculations of this value are shown in Fig. 3),
- list of transactions (tx0, tx1, ..., txn) – each tx means the next transaction stored in a given block.

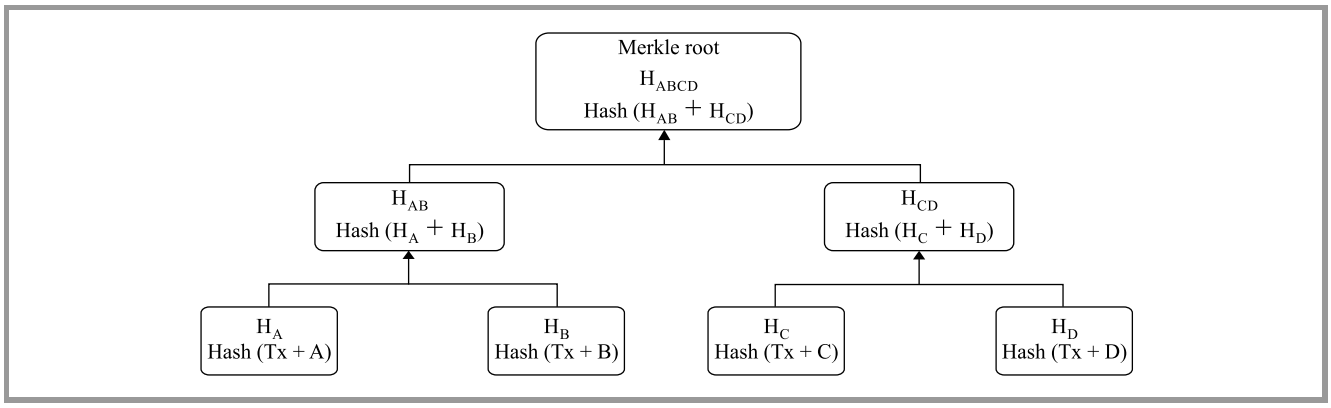

Fig. 3. Merkle tree.

A detailed description of the block components is available in [10].
The data stored in the blocks are protected and encrypted by using the cryptographic methods specified for a given BC network. Usually, these include public and private keys, digital signatures and cryptographic hash methods, such as the Secure Hash Algorithm (SHA) [11].
Each block must be hashed, thus creating a digest ID which represents the block. Any change of data stored in the block will change the hash value, which ensures data immutability [12].
The Merkle tree presented in Fig. 3 is an important component of the block model. It merges the hash values of data in the block until the root of the tree (the top hash value) is generated [10].

2.3. Conflicts and Resolutions

Once the transactions in the block have been completed and the consensus has been reached by the network, the block is added to BC. However, sometimes, when the block is being attached to BC, conflicts arise. Such situations occur if node *A* creates block *n* and distributes it to other nodes and, at the same time, node *B* also creates block *n* and distributes it to the other nodes. The blocks will not be the same in the entire network, because each of them may contain different transactions. These problems generate temporary different versions of blocks (Fig. 4).

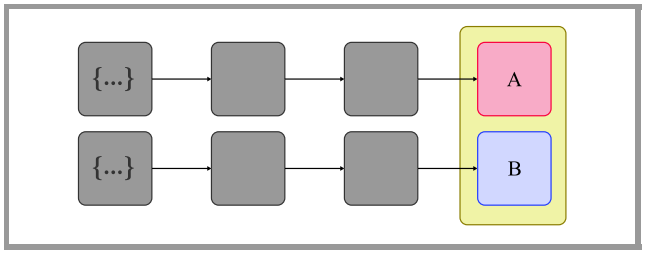

Fig. 4. Blockchain in conflict.

Blockchain systems usually deal with this problem by waiting for the next block to join BC. The longer chain wins and is treated as correct, while the shorter one is removed.

2.4. Security Aspects

Security in BC networks is usually defined as the need to protect transaction- and data-related information in a block. This means that threats and external attacks need to be detected and prevented. Joshi *et al.* in [13] present the main safety procedures in BC:

- **defense in penetration** – a strategy in which many data protection measures are used, based on the fact that many data protection layers are more effective than a single layer,
- **minimum privilege** – access to data is limited to the lowest possible level,
- **manage vulnerabilities** – checking security vulnerabilities and patching them,
- **manage risks** – identification and control of risks in the environment,
- **manage patches** – patching faulty parts of the source code.

BC systems rely on numerous techniques to achieve an adequate security level, mainly for data security purposes, and also for the verification of the nodes' ability to perform specific operations. The concept of accepting the longest chain of blocks as authentic also protects against 51% of attacks and forks problem.

2.5. Cryptographic Methods Used in BC Systems

A asymmetric key cryptography is usually used in BC systems for the authorization of processed transactions [14]. A private key is used to sign transactions, a public key to identify addresses assigned to the user and to verify the signatures generated with the use of private keys. Due to asymmetric cryptography, it is possible to determine whether the user who sends a message to another user has a private key with which the message has been signed, and thus whether he has the right to send it.
In Fig. 5 the process of signing and verifying transactions in BC systems is presented. The transaction is signed with

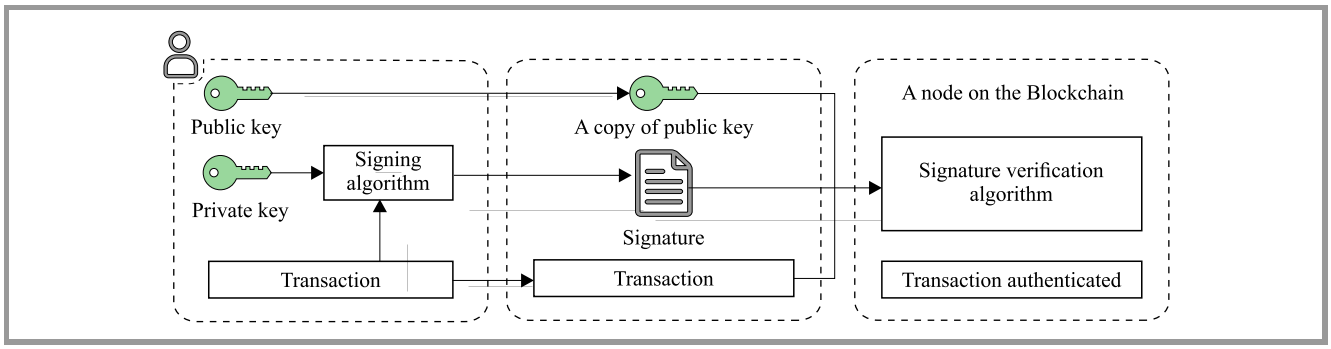

Fig. 5. Asymmetric key cryptography.

a private key, then it is forwarded, together with the signature and the public key, to the recipient. Based on this information, using the verification algorithm, a node in the network may authorize the received transaction.

3. Consensus Model

The acceptance of BC system joining procedure by an external user results in the user's adaptation to the initial state of such a system. The initial BC state is recorded in the genesis block [10], which is always a "head" component in the chain of the blocks. This means that every block must be added to BC after the genesis block, based on consensus method that has been agreed upon. Regardless of the method, each block may be validated independently by each external user (the block is valid). Having the initial state and the ability to verify every block, the external users can agree on the system's current state.
The following procedure should be implemented in the process of defining of the chain of blocks [10]:

- the initial state is defined globally and must be accepted by all external users,
- the external users agree to the consensus method by means of which blocks are added to the BC system,
- each block is linked to the previous block with a specified hash value,
- users may verify each block.

Note that the genesis block is the initial block in the chain and its hash value is set to 0. Through block validations, external users may easily verify the integrity of the BC system. This renders the system distributed and there is no need to have any third-party authority for setting/defining the system's current state. The agreement (consensus) of the active nodes and users in the system is necessary for adding new blocks into the system. The consensus method must work even in the presence of potential malicious users attempting to disrupt or take over BC. The major consensus models are presented later in this section.

3.1. Proof of Work Consensus Model

The proof of work (PoW) consensus model is the most popular agreement method in BC. Here, each external user may add a new block to the existing chain after solving a computationally intensive puzzle. The solution to this puzzle is called the "proof" of the work the user has performed. The puzzle should be defined based on the following conditions:

- the process of solving the puzzle should be complex – the puzzle should be non-trivial and difficult to solve,
- verification and validation of the solution should be easy to process.

Simple validation of the puzzle solution enables the proposed blocks to be validated by other mining system nodes and users. Negative validation of the proposed block automatically rejects the blocks from the chain. The process of solving puzzles conducted by a node does not increase its probability of solving the puzzles faster in the future. Below, we present a simple example of such a puzzle, where a node using the SHA-256 [11] algorithm must find a hash value meeting the following criteria:

$$\text{SHA256}(\text{“test”} + \text{nonce}) = \text{hash value starting with “00”}$$

The string "test" is appended to the value of nonce, and hash value is calculated. Nonce is a numerical value that changes after each hash calculation. This operation is repeated until the result has the form of a hash starting with "00". Some results are presented below:

SHA256("test1") = 1B4F0E9851971998E732078
544C96B36C3D01CEDF7CAA332359D6F1D83567014
1B means "not solved"

SHA256("test2") = 60303AE22B998861BCE3B28
F33EEC1BE758A213C86C93C076DBE9F558C11C752
60 means "not solved"

SHA256("test304") = 009FA371CD0B736AB80E8D
55C5741944DD0E740BBD92C97808F740A03722576B
00 means here "solved"

The above puzzle is not difficult to solve, but with each additional "0" in the expected hash value, i.e. "000", "0000", "0000…", the degree of its complexity increases. The higher the computing power of the mining node, the greater the probability that it will find the solution faster. After finding the solution, the mining node sends the block with the correct hash to other nodes. The recipient's nodes verify that this operation has been carried out correctly. If the verification renders a correct result, they add the block to their chain of blocks and they continue to distribute it further over the network. The PoW has been designed for networks where there is no trust. Both high performance and low performance computing units are capable of solving the puzzle correctly.
However, the main disadvantage of this approach is the consumption of considerable amounts of electricity. Due to the growing difficulty with proofs of work, nodes combine into "pools" or "collectives", where they solve puzzles together and then share the reward. Sharing the problem, each of the nodes may attempt to solve the puzzle at equal intervals:

- node 1: check "test1" to "test100",
- node 2: check "test101" to "test200",
- node 3: check "test201" to "test300",
- node 4: check ""test301" to "test400".

This strategy allows to find the solution more quickly thanks to the cooperation of several nodes. The most popular systems in which PoW is applied include Bitcoin, Litecoin and Ethereum Dogecoin.

3.2. Proof of Stake Consensus Model

In the proof of stake model, the consensus between network blocks is not achieved by mining nodes, but through the minters having stake/tokens. The higher the stake of a given user, the more likely they are to join the block to BC.

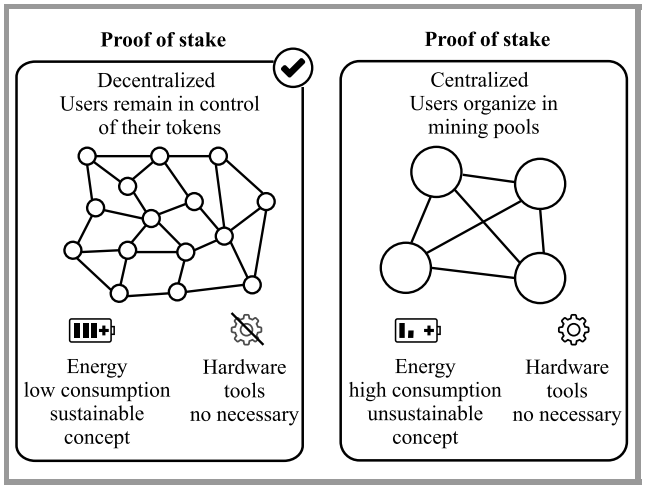

Fig. 6. PoW vs. PoS models.

Let's assume a simple network with 100 tokens, without specific minimum resources needed to participate in the mining process. With 20 tokens, we get a 20% chance to "mine" another block.
Systems using this consensus include, for example, Decred [15] or Peercoin [16]. In some implementations, older tokens have more purchasing power when mining, which may lead to monopolization of the network, i.e. a situation in which users with large resources are getting rich faster than others, and their advantage is growing continuously. There are methods to prevent such situations, which involve the introduction of limited life-time resources, for instance the user must wait, after a successful block check, a certain amount of time before proceeding to confirm the next one. This system is safe until one of the nodes takes over 51% of tokens. In Fig. 6, a simple comparison of PoW and PoS models is presented [17].

3.3. Round Robin Consensus Model

In some systems with a certain level of trust between mining nodes, there is no need of using complicated algorithms to reach the consensus, and the determination of which node will add the next block to BC may be performed alternately. This method is known as the round robin model and is usually used in private BC networks. The publishing of successive blocks is carried out alternately by nodes within the network. If a given node has the right to join the block (its turn has come), but for some reasons it does not join it or is not available, an element of randomness is introduced. This approach does not require high computational power, because there are no cryptographic puzzles to solve here. Nevertheless, a certain level of trust is required, and this model does not work well in open networks (public Blockchains).

4. Blockchain Use Cases

There are many applications that rely, to a lesser or higher degree, on the basic BC principles. Initially, BC was used in digital currency systems. Currently, it is also implemented in voting systems, identity management, smart cities and many other types of applications. Those that deserve particular attention include the following:

- Guardiam – is a token for a new global safety response network that provides a framework for distributed emergency response systems for places in the world where no emergency numbers are available [18],
- Blockchain Charity Foundation – it is a non-profit foundation whose task is to transform philanthropy by building a decentralized charity foundation, supporting sustainable development and ensuring that no one is left behind [19],
- Power Ledger – a system that allows customers to choose a source of electricity, enabling trading elec-

tricity with their neighbors and ensuring a fair return on investment, where energy is stable and affordable for everyone [20],

- EthicHub – a system whose aim is to provide all customers, with individual investors included, with the same access to traditional financial services by democratizing finances and making available investment opportunities around the world [21],
- Grassroots Economics & Bancor – decentralized BC-based community currencies in Kenya, aiming to combat poverty by encouraging local and regional trade [22],
- VeChain – decentralized platform in which companies may easily establish contacts and make transactions without intermediation [23].

The above examples show that the use of this technology not only ensures high security and quick execution of transactions, but also enables to solve problems that have not been solved in any other ways. First of all, it fosters development in areas where technological progress is very slow and where access to technology is very limited. Smart city use cases need to be taken into consideration as well, involving for instance car navigation systems, where the protection of personal data is important [24]. Current solutions, such as Google Traffic or Waze, are a specific type of a black box solution and do not offer sufficient guarantees to those concerned with their privacy.

5. Conclusions

Blockchain technology is the direction in which the industry will be heading over the coming years. The use of cryptographic algorithms ensures appropriate level of security that is required by most ITC environments. Full transparency and data integrity make it suitable for use in many data processing-related domains. Decentralization and the lack of a central supervisor makes the processes where a verification unit is needed faster and more efficient, due to the lack of the human factor and full automation. The trust built by nodes within the network ensures that all operations are carried out in accordance with the rules defined for a given network. Many systems based on Blockchain technology are already in existence. They are subject to continuous improvement and their number may be expected to grow.

References

[1] S. Ølnes and A. Jansen, "Blockchain technology as infrastructure in public sector: an analytical framework", in *Proc. of the 19th Ann. Int. Conf. on Digit. Government Res.: Governance in the Data Age DG.O 2018*, Delft, The Netherlands, 2018, Article no. 77 (doi: 10.1145/3209281.3209293).

[2] B. Marr, "A Complete Beginner's Guide To Blockchain" [Online]. Available: https://www.forbes.com/sites/bernardmarr/2017/01/24/a-complete-beginners-guide-to-blockchain/#6affecba6e60 (accessed 10 Feb. 2019).

[3] Ch. Lafaille, "What Is Blockchain Technology? A Beginner's Guide" [Online]. Available: https://www.investinblockchain.com/what-is-blockchain-technology (accessed 10 Feb. 2019).

[4] J. Seffinga, L. Lyons, and A. Bachmann, "The Blockchain (R)evolution – The Swiss Perspective", Deloitte, Feb. 2017 [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/ch/Documents/innovation/ch-de-innovation-blockchain-revolution.pdf

[5] G. Greenspan, "MultiChain Private Blockchain", Coin Sciences Ltd [Online]. Available: https://www.multichain.com/download/MultiChain-White-Paper.pdf

[6] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: a technical survey on decentralized digital currencies", *IEEE Commun. Surv. & Tutor.*, vol. 18, no. 3, pp. 2084–2123, 2016 (doi: 10.1109/COMST.2016.2535718).

[7] Litecoin – Open source P2P digital currency 2019 [Online]. Available: https://litecoin.org (accessed 10 Feb. 2019).

[8] r3.com [Online]. Available: https://www.r3.com (accessed 10 Feb. 2019).

[9] Energy Web Foundation, [Online]. Available: http://energyweb.org (accessed 10 Feb. 2019).

[10] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview", Draft NISTIR 8202, National Institute of Standards and Technology, 2018 [Online]. Available: https://csrc.nist.gov/CSRC/media/Publications/nistir/8202/draft/documents/nistir8202-draft.pdf (accessed 5 Feb. 2019).

[11] D. Rachmawati, J. T. Tarigan, and A. B. C. Ginting, "A comparative study of Message Digest 5(MD5) and SHA256 algorithm", *J. of Physics: Conference Series, Conf. Series*, vol. 978, 012116, 2018 (doi: 10.1088/1742-6596/978/1/012116).

[12] J. Kołodziej, A. Wilczyński, D. Fernandez-Cerero, and A. Fernandez-Montes, "Blockchain secure cloud: a new generation integrated cloud and blockchain platforms – general concepts and challenges", *Eur. Cybersecur. J.*, vol. 4, no. 2, pp. 28–35, 2018.

[13] A. P. Joshi, M. Han, and Y. Wang, "A survey on security and privacy issues of Blockchain technology", *Mathem. Foundat. of Comput.*, vol. 1, no. 2, pp. 121–147, 2018 (doi: 10.3934/mfc.2018007).

[14] D. Pointcheval, "Asymmetric cryptography and practical security", *J. of Telecommun. and Inform. Technol.*, no. 4, pp. 41–56, 2002.

[15] B. Garner, "What Is Decred (DCR)? A Guide on Decentralized Blockchain Governance" [Online]. Available: https://coincentral.com/decred-lowdown-decentralized-blockchain-governance/ (accessed 9 Feb. 2019).

[16] Peercoin [Online]. Available: https://peercoin.net (accessed 9 Feb. 2019).

[17] "An Introduction to consensus algorithms: Proof of Stake and Proof of Work" [Online]. Available: https://cryptocurrencyhub.io/an-introduction-to-consensus-algorithms-proof-of-stake-and-proof-of-work-cd0e1e6baf52 (accessed 9 Feb. 2019).

[18] Guard Global Decentralized Emergency Response Network, [Online]. Available: https://guardtoken.net (accessed 5 Feb. 2019).

[19] Blockchain Charity Foundation [Online]. Available: https://www.binance.charity (accessed 5 Feb. 2019).

[20] Power Ledger [Online]. Available: https://www.powerledger.io (accessed 5 Feb. 2019).

[21] EthicHub [Online]. Available: https://ethichub.com (accessed 5 Feb. 2019).

[22] "Bancor To Launch First Blockchain-Based Community Currencies in Kenya", [Online]. Available: https://www.businesswire.com/news/home/20180621005727/en/Bancor-Launch-Blockchain-Based-Community-Currencies-Kenya (accessed 5 Feb. 2019).

[23] J. Zwanenburg, "What Is VeChain (VEN)?" [Online]. Available: https://www.investinblockchain.com/what-is-vechain (accessed 5 Feb. 2019).

[24] H. Liviu-Adrian, C. Dobre, "Blockchain privacy-preservation in intelligent transportation systems", in *Proc. IEEE Int. Conf. on Comput. Sci. and Engin. CSE 2018*, Bucharest, Romania, 2018 (doi: 10.1109/CSE.2018.00032).

Andrzej Wilczyński is an Assistant Professor at the Cracow University of Technology and a Ph.D. student at AGH University of Science and Technology. The topics of his research include Blockchain-based modeling in distributed computing, cloud computing and, in particular, data and resource virtualization, tasks scheduling in cloud computing and broadly defined security issues in these domains.

https://orcid.org/0000-0001-6774-3667
E-mail: and.wilczynski@gmail.com
AGH University of Science and Technology
Mickiewicza 30, 30-059 Cracow, Poland

Tadeusz Kosciuszko Cracow University of Technology
Warszawska 24, 31-155 Cracow, Poland

Adrian Widłak majored in Computer Science at Cracow University of Technology, Poland, received his B.A. and M.A. degrees in 2016 and 2017, respectively. He has been a researcher and teacher at Cracow University of Technology, Institute of Computer Science, since 2017. His interests include point cloud processing, artificial intelligence, computational geometry and data visualization.

https://orcid.org/0000-0001-9256-0061
E-mail: adrian.widlak@pk.edu.pl
Tadeusz Kosciuszko Cracow University of Technology
Warszawska 24
31-155 Cracow, Poland