UDC 004.9

INFORMATION TECHNOLOGY FOR DATA AUTHENTICATION BASED ON BLOCKCHAIN TITLE OF MANUSCRIPT

Victoria Vysotska[1], Oleg Prokipchuk[1], Mariia Nazarkevych[2]*, Roman Romanchuk[1]

[1]Department of Information Systems and Networks,
Lviv Polytechnic National University
12 Stepana Bandery St., Lviv, 79013, Ukraine
[2]Department of Radiophysics and Computer Technologies,
Ivan Franko National University of Lviv
50 Drahomanova St., Lviv, 79005, Ukraine

Vysotska, V.A., Prokipchuk, O.A., Nazarkevych, M.A., Romanchuk, R.V. (2025). Information Technology for Data Authentication Based on Blockchain Title of Manuscript. *Electronics and Information Technologies*, 30, 59–74. https://doi.org/10.30970/eli.30.5

ABSTRACT

Background. The article describes the process of developing a blockchain-based technology for information support of the processes of authentication of goods. The paper also investigates the performance of the developed system depending on the adjustment of four parameters: mining complexity, dependence of mining time on technical support capacity, dependence of block creation time on the number of transactions, and dependence of blockchain validation time on the number of blocks.

Materials and Methods. The mining complexity was analyzed in the range from 1 to 7 (with a complexity of 8 or more, mining on a working device takes too long). This parameter determines the number of zeros that must be at the beginning of the hash to consider the work as confirmed. Each subsequent difficulty increases the total time several times. This is due to a significant increase in the number of operations that need to be performed during mining. Of course, mining also depends on the capacity of the technical support, which should be a complex of advanced processors and video cards. The analysis of the dependence of mining time on the processor frequency was conducted for a mining complexity of 5.

Results and Discussion. Additional processor power can significantly reduce mining time. The next study is to analyze the dependence of block creation time on the number of transactions. The main resource burden of this stage is the calculation of hash functions for transactions and for the block, as well as the construction of the hash tree. From the data obtained, we can conclude that the number of transactions in a block is its main resource load. This should be taken into account when choosing the maximum number of transactions per block. The last study was to identify the dependence of blockchain validation time on the number of blocks. The data shows a linear dependence of the blockchain validation time on the number of blocks.

Conclusion. This indicates that the validation process is not overloaded with resource-dependent operations. As the blockchain expands over time, the expected duration of blockchain validation can be calculated according to the following linear relationship.

Keywords: network communication, information system, counterfeit goods, blockchain technology, product life cycle, intelligent information retrieval system

INTRODUCTION

The fight against counterfeiting is ongoing every day, and many methods of identifying authentic goods have already been developed [1-3]. The most popular are the legislative method, the method of developing recommendations, the method of making it difficult to copy, the method of monitoring suspicious ads, and the method of keeping records of each unit of goods [4-6]. The system under development belongs to the latter category. The market for the use of anti-counterfeiting methods is quite developed, as most companies already use at least one of these methods. Among similar systems, Microsoft Aura Ledger stands out, which also uses blockchain technology. It is delivered in the form of software for producers of goods. The advantage of the product under development is that it unites many manufacturers into a single network to ensure greater reliability and transparency of data, as well as the use of two-stage identification, which provides greater accuracy. The main consumers are manufacturers of goods that suffer material damage from counterfeiters [7-11]. Such producers want to minimize the impact of counterfeit goods on their profits, as well as to maintain the reputation and trust of their customers. Competitors include other anti-counterfeiting software providers with the same level of protection.

First, it is necessary to analyze the available tools for solving the problem and select those that are best suited for use in implementing the information system. In order to form a set of tools, first of all, it is necessary to define the tasks to be solved. The main tasks are as follows:

- The developed software needs to be created that allows data and blockchain to be operated in dialogue mode with the user interface.
- The developed software should have high performance of arithmetic and algorithmic operations, be flexible and be able to be customized and scaled.
- The developed software should be able to connect to other applications, in particular to the same software object on other end devices, to combine them into a P2P network.
- Need to create a P2P server to manage information exchange between nodes.
- Need to create a client application in the form of a mobile app or website.

MATERIALS AND METHODS

Once the list of tasks has been compiled, the selection of tools and technologies for their implementation begins. First of all, technical means for implementing the main software application are selected. The technologies used in the creation of the program are as follows: Java, Spring Boot, JavaFX, GSON, and Maven. The same set of technologies was used to implement the P2P server and the server part of the website. The following technology stack was used to create the client part: HTML, CSS, JavaScript. First of all, it is necessary to choose the main programming language. Among the possible options are: Java, Python, C++, C#. To achieve high performance, the language must be compiled. From the available options, the Java programming language was chosen. Java is an object-oriented programming language developed by Sun Microsystems, which was later acquired by Oracle. Programs created using this language are compiled into bytecode, which allows for high performance. Java is compiled into a special code that is recognized by the Java Virtual Machine (JVM). This allows programs to run on any operating system that supports JVM. This multi-platform capability is a big plus for the system being developed, as it allows clients and servers to be developed for different platforms [12]. Java is also object-oriented, which allows the system being developed to be flexible and scalable [13]. The first of the necessary tools should be a framework for centralized object management. Modern enterprise applications are very large and consist of hundreds of classes. With this kind of organization, an application can quickly become highly coupled and non-scalable. To avoid this, the application must adhere to the following rules [14]:

- Application classes should aim for Low Coupling.
- Each class should have only one duty (Single Responsibility Principle).

- When one object creates another object, it performs the duty of creating objects, i.e., according to SRP, it should not do anything else but construct objects. Also, very often objects perform several functions at once, even without considering the creation of other objects.

There are 3 design patterns to help in this situation: strategy, control inversion, and dependency injection.

- Strategy is a design pattern that allows a class to extend its functionality by delegating additional work to auxiliary objects. Thanks to this pattern, all classes will have one main duty and delegate all others.
- Dependency injection is a design pattern for changing the order in which auxiliary objects are assigned to the main object. The class does not create auxiliary objects itself but only declares containers for them. The controlling program injects the auxiliary objects into the main one. In this way, the object avoids the obligation to construct objects.
- Control Inversion is a design pattern that is a module that registers and constructs application objects. This pattern allows you to automate the process of dependency injection, thereby significantly reducing the complexity of the program [15].

The use of these design patterns allows for high flexibility and scalability of the application. For the system under development, 2 possible frameworks were chosen: Google Guice and Spring Boot. I chose the latter because of its greater functionality [16]. The next step is to choose a tool for creating the user interface. The choice of such tools for the Java programming language is small and consists of two tools: Swing and JavaFX [17].

Both tools allow the creation of high-quality user interfaces and were developed under the leadership of Oracle. Starting with Java version 9, JavaFX is no longer included in the core JDK development package and is being developed separately from Oracle. JavaFX was chosen for this project because of its convenience and superior functionality, in particular the ability to create designs in FXML files with XML markup, whereas in Swing, components can only be created and populated programmatically. One of the processes that requires a lot of extra time to implement manually is serialization and deserialization. Serialization is the conversion of an object into a byte or character format that is convenient for transfer to non-Java environments. Deserialization is the reverse process, i.e. converting a character or byte format back into a software object. For this program, the JavaScript Object Notation (JSON) character data transfer format was chosen because it is simple to implement and easy to read. This is necessary for tracking and demonstrating the intermediate results of the programme's work. Possible implementations of this technology are Google GSON [19] and Jackson. GSON was chosen because of its higher popularity, which ensures more active work on the software product.

For convenient development and ensuring the correct interaction of all software components, it is advisable to use one of the building systems. These systems automatically compile and link all modules into a single software package and provide the ability to control the versions of all components and the main product. There are 3 such systems for Java: Ant, Maven, and Gradle. Today, Ant is an outdated system. Of the two modern systems, Maven [20] was chosen because of the widespread use of its repositories. Although Gradle is a more modern and advanced system, not all components are compatible with it yet.

One of the key points in developing the main program is network communications. A method of data exchange using the TCP protocol is required. Unlike UDP, TCP [21] checks sent data for possible losses, which is very necessary in a secure and accurate system. One way to achieve this is to use the HTTP application layer protocol, as it is based on TCP and there are many frameworks for Java that allow this functionality to be easily implemented. Despite this, standard Java functionality was chosen to create Socket/Server connections for all P2P communications. This method allows a channel to be established between the client and the server, both of which can listen and edit. This approach is necessary for implementing a P2P network because the communication of such a connection is not regulated by higher-level protocols and can therefore be arbitrary for the developed product. Although the HTTP protocol was rejected for the implementation of a P2P network, it is an integral part of a web server. For the client application, the option of a website that communicates with the server using the HTTP protocol was chosen. HTTP is an application-level protocol based on the TCP/IP protocol stack. Communication between the server and the client is performed using a request-response pattern. The client forms a request, fills it with data and sends it to the server. In turn, the server processes the request and sends a response to the client. Communication takes place using one of the following methods: GET, POST, PUT, DELETE, OPTIONS, HEAD, PATCH, TRACE, CONNECT [22]. In most cases, the first four methods are used. There are several basic ways to implement such a server, but they are all based on the use of a Servlet container. A servlet is a software unit (object) that processes a client request in a separate thread. The most popular implementation of a Servlet container is Tomcat [23]. However, Spring Boot, which is already used in the program under development, contains extensions compared to Tomcat, which increase its functionality and simplify interaction with other Spring Boot components, which is why this tool is chosen for the implementation of the web server.

In turn, the client side is a web application. Any website is based on three main technologies: HTML, CSS, and JavaScript. This technology stack has no current analogues, so the choice is quite simple. At the beginning of the Internet's development, various tools could be used to create a website, but over time, they have all been ousted from the market.

JavaScript is a scripting language that is executed every time a website is launched. It is used to execute all the logic of a website, as well as to provide user interaction with the website and to asynchronously exchange data with the server. The language's runtime environment, such as a browser, usually restricts the language's access to the resources of the owner's device, so visiting websites is completely safe, as the execution of dangerous scripts has been prevented. Since the language is scripted, the user could call new commands directly while using the website [24]. When the technology stack has already been formed, it is time to choose software development and testing environments. The first such environment should be a tool for developing, compiling, and running Java code. Today, the main competitors on the market are Eclipse and JetBrains Intellij Idea. NetBeans used to be an active participant in this competition, but now its market share is very small. Based on a general review of previous products, it can be said that these are very powerful and functional tools. A more detailed comparison is shown in Table 1. To perform this work, we obtained a student license for the Intellij Idea Ultimate Edition [25].

The second necessary tool is a means of debugging and testing the client web application. Most browsers today have built-in tools for this purpose. The most popular among them are Google Chrome, Firefox, and Opera. Among the above-listed options, Google Chrome was chosen due to its functionality and ease of use. All hardware and software tools are used to perform this work in a specific configuration. Each of them contains its own unique parameters that characterize a particular application. Most tools contain such characteristics as product version, developer, system requirements, etc. For selected software products, here are their detailed specifications.

To date, blockchain has become widespread and is used in many areas of everyday life, especially in cryptocurrencies such as Bitcoin, Ethereum, Binance Coin, Tether, Bitcoin cash, Litecoin, and others [26]. The principle of blockchain operation is shown in Fig. 1.

The main qualitative characteristic of this technology is the security and transparency of this data structure, as new blocks can only be added to the blockchain without the possibility of editing them. This means that any data entered into the blockchain becomes visible to all its participants and remains there in its original format for the entire duration of the blockchain's existence. There are many blockchain implementations, so its behavior

Table 1. **Comparison Intellij Idea and Eclipse**

Parameters	Intellij Idea	Eclipse
System requirements	Minimum 2 GB of RAM	Minimum 0.5 GB of RAM
Distribution method	Free with paid version	Free
Debugging	An extended set of debugging tools	Standard debugging tools
Plugins	750+ plugins	1250+ plugins
Auto-complete	Automaton	Use the key combination Ctrl + Space
Productivity	Optimized for indexed transactions	Faster under heavy loads
Refactoring	An extended set of tools	Standard set of tools
Design	Modern design, easy to use	Outdated and overloaded design
Focus	Small and medium-sized projects	Large projects

and characteristics can often vary from one product to another. The first successful and most well-known blockchain implementation, Bitcoin, was taken as a basis for this paper. A block is the basic structural unit of a blockchain. The only operation to change the blockchain is to add a new block to an existing chain. Blocks are created by the nodes of the blockchain network. In addition to being a structural unit of the chain, the most important function of a block is to be a receptacle for data that must be unchanged. A block consists of a header and basic data. The header of a block is also called its metadata. This division was created due to the fact that the blockchain can reach very large sizes with prolonged use. Thus, the Bitcoin cryptocurrency today occupies more than 200 GB. This is not a problem for specialized desktop devices, but for mobile devices, downloading 200 GB of data can be a challenge. For optimization purposes, the block was divided into a header containing the minimum required data set and the main body. This makes it possible to develop mobile clients that work only with block headers. Such mobile clients have one drawback. If necessary, they have to download the necessary data, and when using such

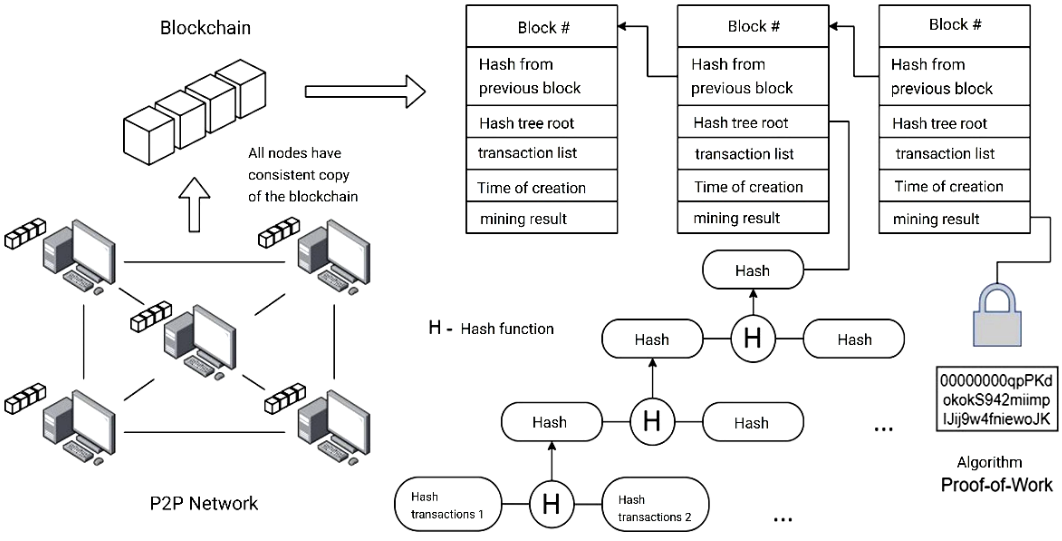


Fig. 1. Principles of blockchain robots.

Table 2. **Block structure**

Field	Size
Caption	
Hash of the previous block	32 Byte
Hash block	32 Byte
Hash root	32 Byte
Hour of creation	8 Bytes
Appendix (nonsense)	8 Bytes
Main part	
Hash tree	Depending on the number of transactions
List of transactions	Limited to implementation

a client, it is very important to make sure that this download is carried out from a reliable source. The structure of the block is shown in Table 2. Thus, the size of the block header is 112 Bytes. Let's take a closer look at each of the components [27].

A Block Hash is a unique block identifier obtained by passing the block header through a hash function. The SHA-256 hashing algorithm was used to perform this work. The algorithm generates 256-bit hash for any data. To form a chain, the blocks must be connected in a certain way. Each block contains a reference to the previous block in the form of a hash of the previous block (Previous Block Hash). As a result, having the last block, it is possible to trace the initial block by moving through the previous hashes. The hash of the previous block is included in the data used to calculate the hash of the current block. This provides protection against data tampering. If the data of any block of the blockchain is changed, all subsequent blocks become invalid. The Merkle root is the root of the hash tree. A Merkle tree is a tree built based on hash values of transactions. The tree is built using the following algorithm:

- The construction starts with the tree leaves, which are the hashes of each of the submitted transactions.
- Transaction hashes form a queue.
- 2 elements are selected from the queue, and a parent node is created on their basis.
- The parent node is created as a result of passing the concatenation of children's hashes through the hash function.
- The resulting node is added to the queue of the next stage.
- The operation is repeated until the current queue is over, after which the queue of the next stage becomes the current one.
- The general algorithm is repeated until there is only 1 element left in the final queue, which is the root of the tree.

Thus, each parent node is a hash of the hashes of the child nodes [28]. Blockchain nodes always accept only the longest chain of blocks, all others are considered irrelevant and discarded. The blockchain is already protected from data tampering by hashing it, but what prevents an attacker from creating a longer chain of blocks so that other nodes will accept the fake blockchain as the real one? The Proof-of-Work algorithm was developed to prevent such a situation. This algorithm is based on the fact that the creation of each block should be accompanied by certain resources and time costs of the processor. The process that provides such costs is Mining. Mining a block means selecting a Nonce value such that the block's hash starts with a certain number of zeros. The number of zeros that should be at the beginning of the hash is determined by the complexity of mining. Difficulty is a configuration parameter that can be used to control the duration of mining. There are two main algorithms for selecting a Nonce value: iteration and the use of random numbers. The use of random numbers allows this algorithm to be executed in parallel in multithreaded mode to increase mining performance. Thus, in order to create a fake chain that is longer than the actual one, it is necessary to perform Proof-of-work for each block of the new chain. If the mining time is 10 minutes and an attempt is made to compromise a blockchain with a length of 1,000 blocks, this will take 10,000 minutes. This makes such attacks virtually impossible. The main data of a block is its transactions. A transaction stores secure data that can interact with each other. The person who creates the block (miner) is not necessarily the owner of the transaction. A node receives transactions from other nodes, combines them into a single block, and adds them to the blockchain. The content of a transaction can be anything. In the Bitcoin system and in the application of this paper, the filling is used in the form of transferring a numerical balance from one wallet to another. A transaction consists of: transaction id, sender's address, recipient's address, transferred balance, inputs, outputs, additional data, and an electronic digital signature. Let's take a closer look at each of the elements:

- The transaction Id is its unique identifier assigned after its creation.
- The sender/receiver addresses are the public keys of the wallets.
- A wallet is a set of public and private cryptographic keys. The balance of a wallet can be calculated. A wallet is used by the sender to create a transaction to transfer funds to another wallet. The public key of a wallet is used as its address and to verify digital signatures. The private key of a wallet is used to create a digital signature for a transaction. Public and private keys are generated using the RSA algorithm. The generation of such keys depends on random numbers. In turn, the algorithm for generating such numbers must not have any patterns.
- RSA is an asymmetric cryptographic encryption algorithm based on the properties of large prime numbers. The purpose of the algorithm is to encrypt data and create a digital signature. The algorithm is widespread today and is used in many applications [29].

To start the system, it is necessary to combine the nodes into a single network, as the reliability of the blockchain directly depends on the number of users. When implementing the standard, it was decided to move away from the standard centralized client-server model in favor of the decentralized P2P model. P2P is a network topology in which each participant can simultaneously act as a client, making requests to one group of nodes, and be a server, responding to requests from another group of nodes. Such a network is not hierarchical, and all its endpoints are on the same level. If one of the nodes fails, the network can still operate correctly. The nodes of such a network are called peers [30]. There are several ways to organize such a network: decentralized, centralized, and mixed. Decentralized P2P consists only of nodes located on the 1st level, and neighboring peers are used to find nodes. In a centralized P2P, there is a server that knows the addresses of the nodes, and when establishing communication, peers contact the server to get the addresses of other peers. The mixed method uses both approaches.

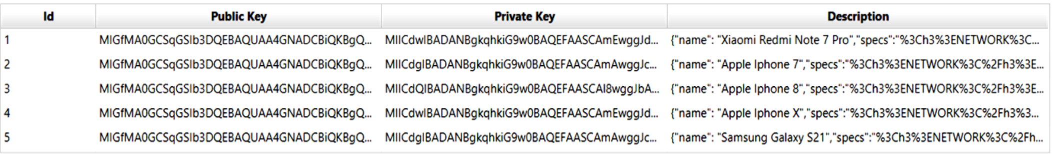
In a blockchain, each pir has its own local copy of the blockchain. When a node starts working, it scans the network for the longest chain of blocks, and if it finds one, it updates its local copy. When a node creates a block, it sends it to all available peers. Both processes have one thing in common: validation. Among the network nodes, there can be both reliable peers and malicious actors that promote fake data. To avoid distortion of the blockchain, each node checks all data received in the network for validity. The validation algorithm sequentially goes through each block and recalculates all the data in search of inconsistencies, such as transactions created by someone other than the wallet owner, distorted block data, etc. Such verification may be slow if there are a large number of blocks. When planning the implementation of a blockchain, it is necessary to carefully choose the means of its creation. It was decided to create all the logic in the Java programming language. To demonstrate the operation and structure of the blockchain, it was decided not to use auxiliary libraries to create it, but to implement it manually. Manual implementation also provides more flexibility in development and control. The next step is to select auxiliary algorithms and tools for their implementation. The hashing algorithm used was SHA256, as it is quite common and is used in the Bitcoin product, which became the basis for the developed IP. The algorithm can be conveniently implemented using the Java package java.security. RSA was chosen as a cryptographic algorithm primarily because it can be easily implemented using the java.security package. Its main analogue was the ECDSA algorithm, which is based on the properties of elliptic curves, but its implementation in Java is complicated. To generate random numbers, we chose the SecureRandom tool from the java.security package. When choosing the type of network topology, the main criterion was the ability to manage nodes for commercialization. That is why a centralized type was chosen.

Summary of the main material

The first step in using the application is to go to the Products menu. After switching, a blank table of product types and possible options will be displayed. To create a new product type, click the ‘Units’ button. This will open a modal window where you can enter data for the new product. A special feature of this window is the ability to manually enter product type keys or automatically generate them using the ‘Generate Keys’ button. As a result of creating several product types, the window for viewing these types will look as shown in Fig. 2 below. At this stage, you need to select a specific type of product and press the "Units" button to go to specific units of goods (Fig. 3). Each physical item is a product unit. Although different units may have the same product type, each product unit has unique identifiers to distinguish the required product. Create several product units using the "Add" button (Fig. 4). Both the product type creation window and the product unit creation window have a defined key for automatic field generation. In this case, the "Public Info" and "Private Info" fields must be saved, they will be needed later (Fig. 5).

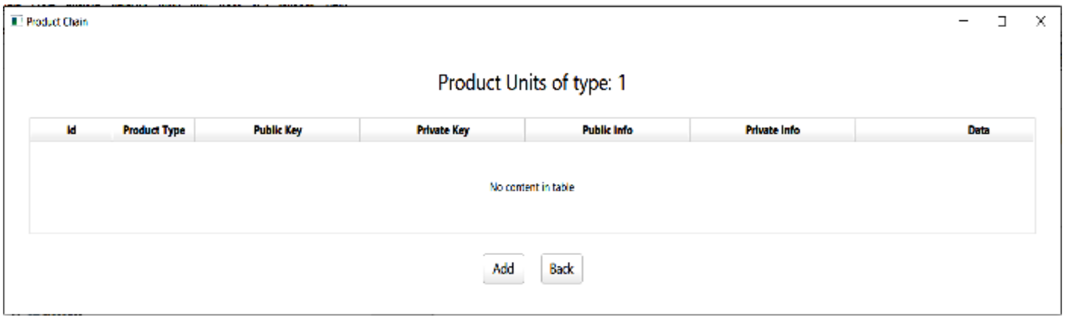
This is what the product units scene looks like after creating several instances. The next step is to add the created data to the blockchain. To do this, return to the main menu and open the transaction scene using the ‘Transactions’ button. The only thing visible in this scene will be an empty transaction table, as no transactions have been created yet. To automatically create the necessary transactions, click the ‘Refresh’ button (Fig. 6).

In Fig. 6, there are four automatically generated transactions. Three of them correspond to the three created units of goods and have a balance of 1, and the last



Id	Public Key	Private Key	Description
1	MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQ...	MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJd...	{"name": "Xiaomi Redmi Note 7 Pro","specs":"%3Ch3%3ENETWORK%3C...
2	MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQ...	MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJc...	{"name": "Apple Iphone 7","specs":"%3Ch3%3ENETWORK%3C%2Fh3%3E...
3	MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQ...	MIICdQIBADANBgkqhkiG9w0BAQEFAASCAl8wggJbA...	{"name": "Apple Iphone 8","specs":"%3Ch3%3ENETWORK%3C%2Fh3%3E...
4	MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQ...	MIICdwIBADANBgkqhkiG9w0BAQEFAASCAmEwggJd...	{"name": "Apple Iphone X","specs":"%3Ch3%3ENETWORK%3C%2Fh3%3...
5	MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQ...	MIICdgIBADANBgkqhkiG9w0BAQEFAASCAmAwggJc...	{"name": "Samsung Galaxy S21","specs":"%3Ch3%3ENETWORK%3C%2Fh...

Fig. 2. Scene of product types.

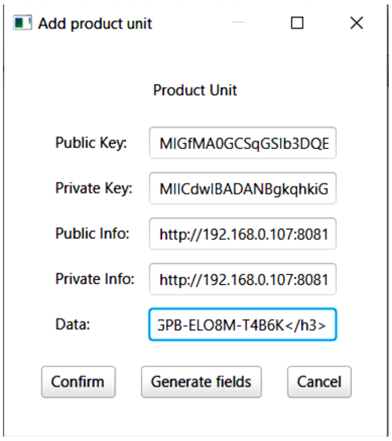


Product Units of type: 1

Id	Product Type	Public Key	Private Key	Public Info	Private Info	Data

No content in table

Fig. 3. Product unit scene.



Product Unit

Public Key: MIGfMA0GCSqGSIb3DQE

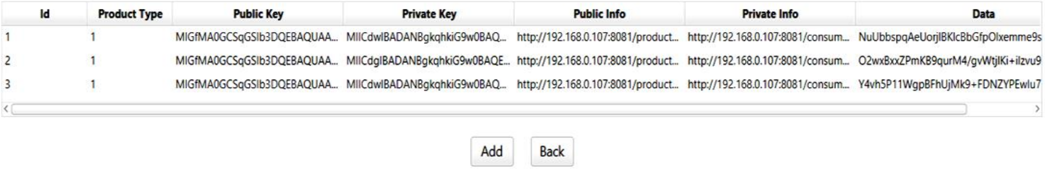
Private Key: MIICdwIBADANBgkqhkiG

Public Info: http://192.168.0.107:8081

Private Info: http://192.168.0.107:8081

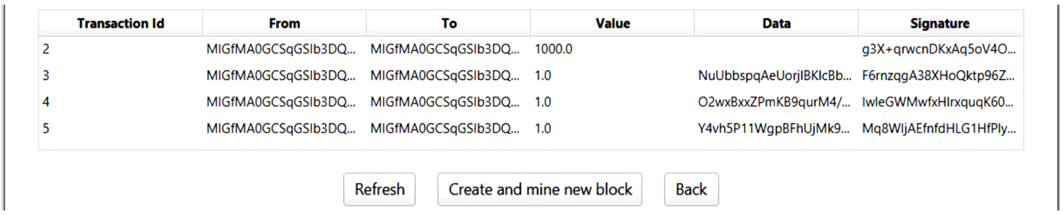
Data: 3PB-ELO8M-T4B6K</h3>

Fig. 4. Product unit creation window.



Id	Product Type	Public Key	Private Key	Public Info	Private Info	Data
1	1	MIGfMA0GCSqGSIb3DQEBAQUAA...	MIICdwIBADANBgkqhkiG9w0BAQ...	http://192.168.0.107:8081/product...	http://192.168.0.107:8081/consum...	NuUbbspqAeUorjlBKlcBbGfpOlxemme9s
2	1	MIGfMA0GCSqGSIb3DQEBAQUAA...	MIICdgIBADANBgkqhkiG9w0BAQE...	http://192.168.0.107:8081/product...	http://192.168.0.107:8081/consum...	O2wxBxxZPmKB9qurM4/gvWtjlKi+ilzvu9
3	1	MIGfMA0GCSqGSIb3DQEBAQUAA...	MIICdwIBADANBgkqhkiG9w0BAQ...	http://192.168.0.107:8081/product...	http://192.168.0.107:8081/consum...	Y4vh5P11WgpBFhUjMk9+FDNZYPEwlu7

Fig. 5. Product unit scene.



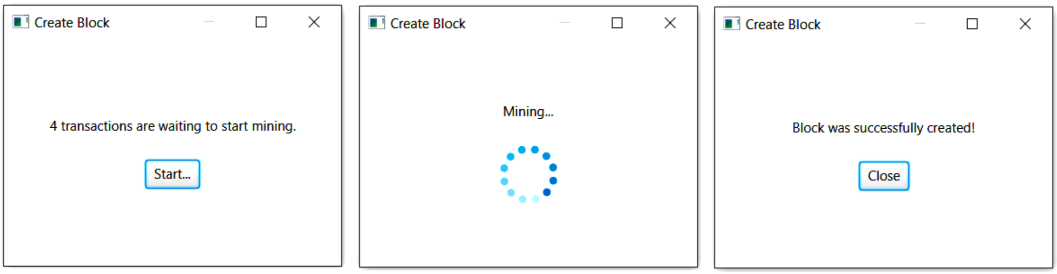
Transaction Id	From	To	Value	Data	Signature
2	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1000.0		g3X+qrwcnDKxAq5oV4O...
3	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1.0	NuUbbspqAeUorjlBKlcBb...	F6rnzqgA38XHoQktp96Z...
4	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1.0	O2wxBxxZPmKB9qurM4/...	IwleGWMwfxHlrxquqK60...
5	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1.0	Y4vh5P11WgpBFhUjMk9...	Mq8WIjAEfnfdHLG1HfPly...

Fig. 6. Transaction viewing scene.

transaction, with a balance of 1000, corresponds to the transfer of funds from the base wallet to a specific type of goods. This is necessary so that units can be created from a type of goods. After such a transfer, it is possible to create 1,000 units of one type of goods. To create a block based on the transaction data and add it to the blockchain, click the ‘Create and mine new block’ button, which will open the ‘Create Block’ modal window (Fig. 7). The creation of a block is visually divided into three stages:

- Notification of how many transactions will be added to the created block. This value depends on the number of transactions created and the maximum possible number of transactions in a block. In this example, the maximum number of transactions is set to 4, so all transactions are included in the block. If there are more transactions, it is necessary to repeat the block creation until all transactions are applied.
- The mining stage is the longest stage, depending on the set complexity.
- Block creation status can be either successful or unsuccessful.

After creating a block, one can proceed to the main menu and the blockchain stage by pressing the ‘Blocks’ button (Fig. 8). On this stage, it is possible to view all blocks and values recorded in the blockchain. By default, this list is organized into a tree structure, sorted by blocks in descending order, and all its values collapsed. Use the ‘Refresh’ button to update the list of blocks to the current one. Now the manufacturer needs to generate QR codes using any generator based on the saved ‘Public info’ and ‘Private Info’ fields and

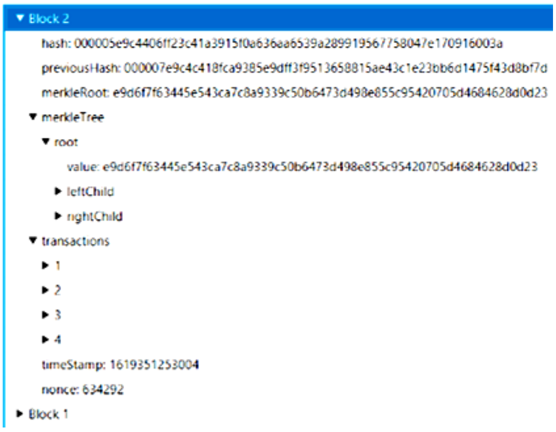


4 transactions are waiting to start mining.

Mining...

Block was successfully created!

Fig. 7. Stages of creating a block.



```
▼ Block 2
    hash: 000005e9c4406ff23c41a3915f0a636aa6539a28991956775804 7e170916003a
    previousHash: 000007e9c4c418fca9385e9dff3f9513658815ae43c1e23bb6d1475f43d8bf7d
    merkleRoot: e9d6f7f63445e543ca7c8a9339c50b6473d498e855c95420705d4684628d0d23
  ▼ merkleTree
    ▼ root
        value: e9d6f7f63445e543ca7c8a9339c50b6473d498e855c95420705d4684628d0d23
      ▶ leftChild
      ▶ rightChild
  ▼ transactions
    ▶ 1
    ▶ 2
    ▶ 3
    ▶ 4
    timeStamp: 1619351253004
    nonce: 634292
▶ Block 1
```

Fig. 8. Blocks viewing scene.

place them on the product packaging (Fig. 9). The control example continues from the buyer's side, and the first step is to scan the public code. Upon receiving the product, before unpacking it, the buyer can scan the external QR code and find out whether the product has already been used and whether the code leads to the correct manufacturer.

Fig. 10a shows what this page looks like on a mobile device. From the information obtained, it is possible to determine the correct manufacturer's website, verify that the characteristics of the actual product match the information on the website, and confirm the status of the product as unused. Therefore, it can be concluded that the product is authentic. Otherwise, the product is determined to be counterfeit, and the buyer can either refuse to purchase the product or contact the appropriate consumer protection authorities. Since this product is authentic, the buyer opens the package and scans the internal QR code. As a result of scanning the internal QR code, the buyer receives information about the specific product, such as activation codes, promo codes, etc. Here comes the second

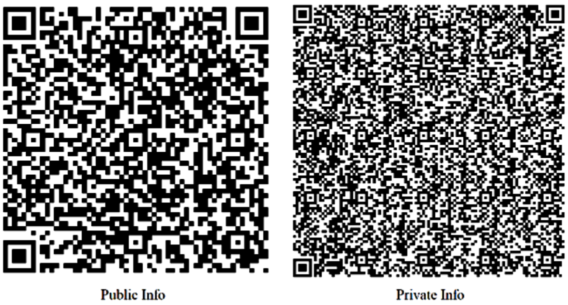
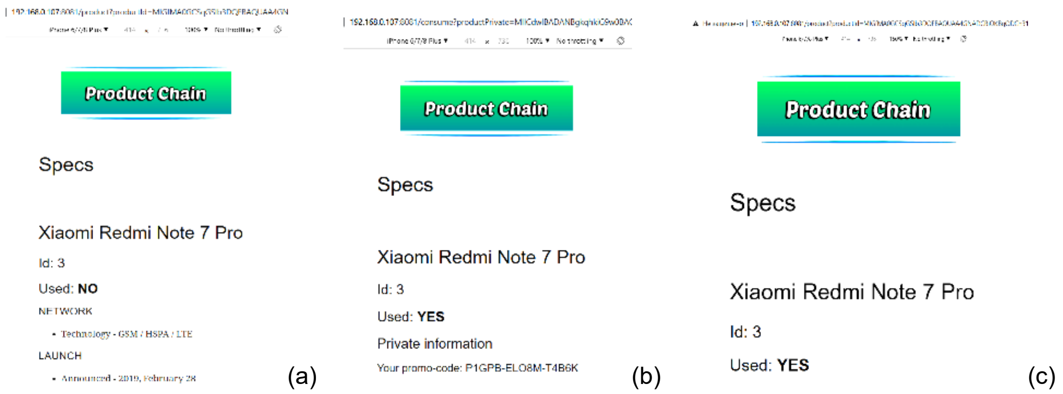


Fig. 9. Generated QR codes.



192.168.0.107:8081/product?productId=MIGfMA0GCSqGSIb3DQEBAQUAA4GN

Product Chain

Specs

Xiaomi Redmi Note 7 Pro

Id: 3

Used: **NO**

NETWORK

- Technology - GSM / HSPA / LTE

LAUNCH

- Announced - 2019, February 28

(a)

192.168.0.107:8081/consume?productPrivate=MIICdwIBADANBgkqhkiG9w0B

Product Chain

Specs

Xiaomi Redmi Note 7 Pro

Id: 3

Used: **YES**

Private information

Your promo-code: P1GPB-ELO8M-T4B6K

(b)

Product Chain

Specs

Xiaomi Redmi Note 7 Pro

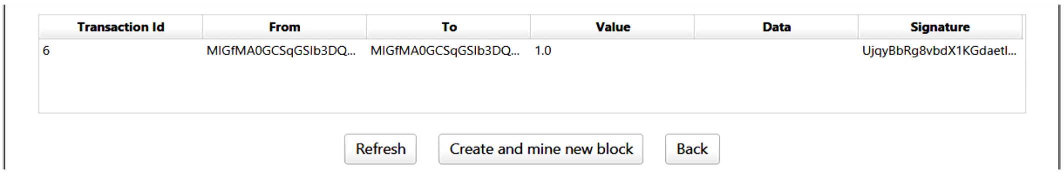
Id: 3

Used: **YES**

(c)

Fig. 10. Transition by a) Public Info, б) Private Info та c) re-transition by Private Info.

authentication factor: the buyer checks whether the product ID from the public and private QR codes match, and then scans the public QR code again, which changes the status of the product to ‘used.’ After scanning the external QR code again, the buyer sees that the product is now used. At this point, the authenticity check is complete, and the buyer can start using the product. The control example continues from the manufacturer's side. When the private link is clicked, a consumption transaction is created in the manufacturer's Product Chain app, as shown in Fig. 11, which must also be added to the block and then to the blockchain.



Transaction Id	From	To	Value	Data	Signature
6	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1.0		UjqyBbRg8vbdX1KGdaetl...

Fig. 11. Transaction of consumption of goods.

Since the application is connected to a P2P server, it can send and receive data from other nodes. Let's start another process of the manufacturer's client application running on other ports using the command: *java -DSERVER_PORT=8001 -DCLIENT_PORT=8102 -DWEB_SERVER_PORT=8082 -jar ProductChain.jar.* The new application will no longer have an empty blockchain but will download the existing blockchain from the first node (Fig. 12).

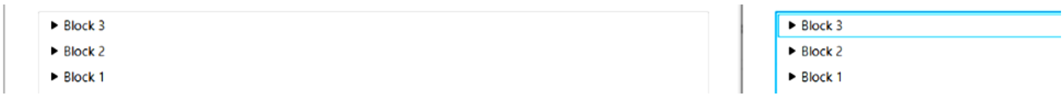
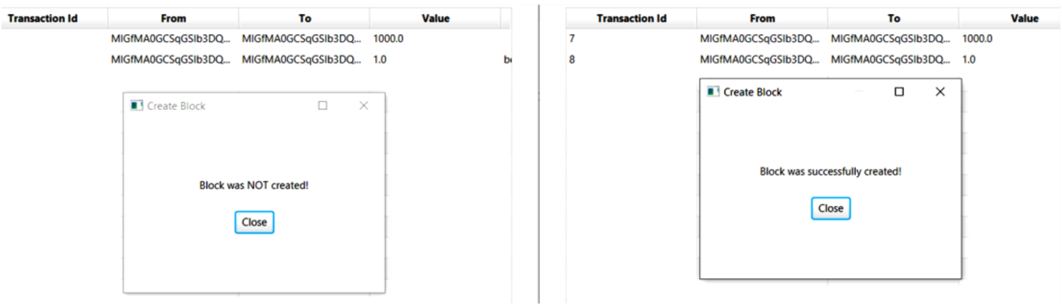


Fig. 12. Two processes of the Product Chain application running simultaneously/

Both processes contain the same number of blocks. From this point on, after each new block is mined, this block is distributed throughout the blockchain network so that all its nodes have the most up-to-date state. However, the same feature is also a difficulty because if a node receives a new block during the mining process, it will have to cancel mining and start creating the block again. To demonstrate this phenomenon, we start mining simultaneously on two processes (Fig. 13). According to the conclusions of each block, the mining of the first process was unsuccessful, while the second was successful. In other words, the first process received a new block during mining and stopped it in order to add the block to the local copy of the blockchain and recreate the transactions.



Transaction Id	From	To	Value
	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1000.0
	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1.0

Create Block

Block was NOT created!

Close

Transaction Id	From	To	Value
7	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1000.0
8	MIGfMA0GCSqGSIb3DQ...	MIGfMA0GCSqGSIb3DQ...	1.0

Create Block

Block was successfully created!

Close

Fig. 13. The result of the simultaneous mining of blocks of two processes

To optimize further work with the program, as well as to search for opportunities to increase performance by changing the working environment, the performance aspects of the system are studied depending on the adjustment of certain parameters. The first such parameter is mining complexity. This parameter determines the number of zeros that must be at the beginning of the hash in order for the work to be considered confirmed. Thus, the complexity range from 1 to 7 is investigated (at complexity 8 and above, mining on the working device takes too long). The results of the analysis are shown in Fig. 14.

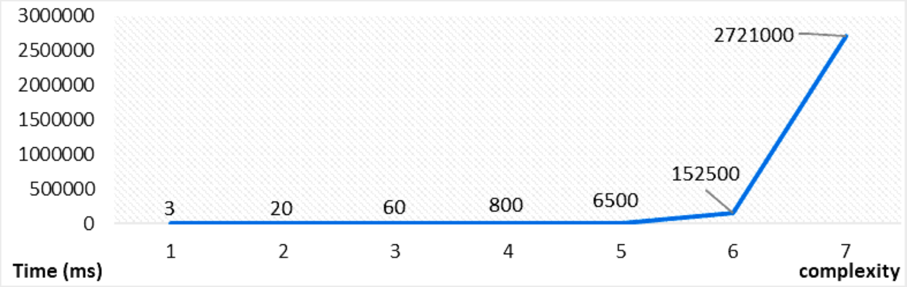


Fig. 14. Graph of the dependence of mining time on its complexity.

The time is shown in milliseconds. This graph shows that each subsequent complexity increases the total time several times. This is due to a significant increase in the number of operations that need to be performed during mining. Of course, mining also depends on the power of the technical equipment. Real devices that mine on a regular basis are a bunch of fancy processors and graphics cards. The dependence of mining time on processor frequency is shown in Figure 15. The analysis was performed for mining complexity - 5. The data shows that additional processor power can significantly reduce mining time. The next subject of research is block creation. The main resource load at this stage is the calculation of hash functions for transactions, for the block, and the construction of a hash tree. The results of the study are shown in Figure 16. From the data obtained, it

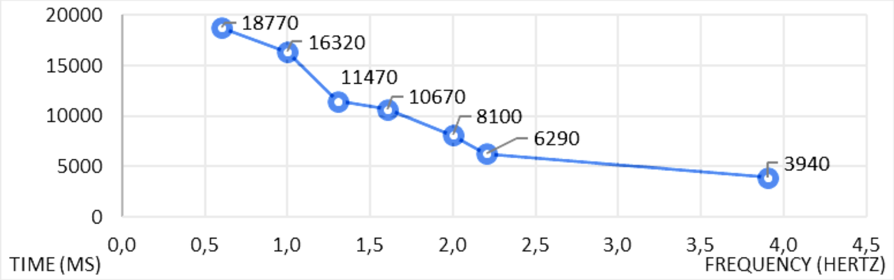


Fig. 15. Graph of mining time versus processor frequency.

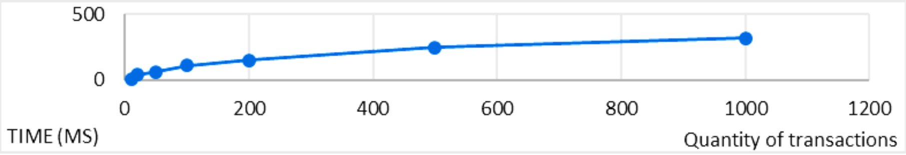


Fig. 16. Graph of block creation time vs. number of transactions.

can be concluded that the number of transactions in a block is its main resource load. This should be taken into account when selecting the maximum number of transactions per block.

The last study was to determine the dependence of blockchain validation time on the number of blocks. The results of this study are shown in Figure 17. The graph shows a linear dependence of the blockchain validation time on the number of blocks. This indicates that the validation process is not overloaded with resource-dependent operations. As the blockchain expands over time, you can expect the expected duration of blockchain validation to be linearly dependent on.

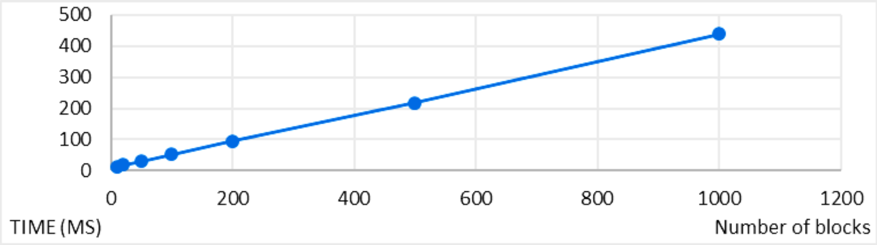


Fig. 17. Graph of the dependence of blockchain validation time on the number of blocks.

RESULTS AND DISCUSSION

As a result of the work, an information system for verifying the authenticity of goods based on blockchain technology was developed. All necessary modules for a complete demonstration of the system were developed, including blockchain modules, a manufacturer's user interface, and a web server. A P2P server was also implemented, allowing the developed software applications to interact. The issues were studied, a system analysis was carried out, and technical means of implementation were selected. After that, the system was developed and its behavior was studied. The work also included a study of the performance aspects of the developed system depending on the adjustment of four parameters: mining complexity, mining time dependence on technical capabilities, block creation time dependence on the number of transactions, and blockchain validation time dependence on the number of blocks. The mining complexity was analyzed on a scale from 1 to 7 (at complexity 8 and above, mining on a working device takes too long). This parameter determines the number of zeros that must be at the beginning of the hash in order for the work to be considered confirmed. Each subsequent complexity increases the total time several times. This is due to a significant increase in the number of operations that must be performed during mining. Of course, mining also depends on the power of the technical equipment. Real devices that mine on a regular basis are a complex of advanced processors and video cards. An analysis of the dependence of mining time on processor frequency was performed for mining difficulty - 5.

CONCLUSION

Additional processor power can significantly reduce mining time. The next study is an analysis of the dependence of block creation time on the number of transactions. The main resource load at this stage is the calculation of hash functions for transactions, for a block, as well as the construction of a hash tree. From the data obtained, it can be concluded that the number of transactions in a block is its main resource load. This should be taken into account when selecting the maximum number of transactions per block. The last study is to identify the dependence of blockchain validation time on the number of blocks. The data obtained shows a linear dependence of blockchain validation time on the number of blocks. This indicates that the validation process is not overloaded with resource-dependent operations. When expanding the blockchain over time, the expected duration of blockchain validation can be calculated according to the linear dependence.

ACKNOWLEDGMENTS AND FUNDING SOURCES

This publication was prepared with the grant support of the National Research Foundation of Ukraine, project registration number 187/0012 dated 08.01.2024 (2023.04/0012) "Development of an information system for automatic detection of sources of disinformation and unreliable behavior of chat users" within the framework of the competition "Science for Strengthening Ukraine's Defense Capability".

REFERENCES

[1] Li X., Wang C. The technology and economic determinants of cryptocurrency exchange rates: The case of Bitcoin. Decision support systems, 2017, 95: 49-60. https://doi.org/10.1016/j.dss.2016.12.001

[2] Чубенко А.Г., Лошицький М.В., Павлов Д.М., Бичкова С.С., Юнін О.С. Термінологічний словник з питань запобігання та протидії легалізації доходів, одержаних злочинним шляхом, фінансуванню тероризму, фінансуванню розповсюдження зброї масового знищення та корупції, К.: Ваіте, 2018. – 826 с.

[3] Allison. Behind the industry of counterfeit products in China and lawsuit success cases, 2021. URL: https://daxueconsulting.com/counterfeit-products-in-china/

[4] Butticè, V., Caviggioli, F., Franzoni, C., Scellato, G., Stryszowski, P., & Thumm, N. Counterfeiting in digital technologies: An empirical analysis of the economic performance and innovative activities of affected companies. Research Policy, 49(5) (2020) 103959. https://doi.org/10.1016/j.respol.2020.103959

[5] Richter F. The Industries Most Affected by Counterfeit Products, 2019. URL: https://www.statista.com/chart/17410/counterfeit-and-pirated-products-by-category/.

[6] Prokipchuk, O., Chyrun, L., Bublyk, M., Panasyuk, V., Yakimtsov, V., & Kovalchuk, R. (2021). Intelligent System for Checking the Authenticity of Goods Based on Blockchain Technology. In MoMLeT+ D, pp. 618-665.

[7] Kumar R., Tripathi R. Traceability of counterfeit medicine supply chain through Blockchain. In: 11th international conference on communication systems & networks (COMSNETS). IEEE, 2019. p. 568-570. https://doi.org/10.1109/COMSNETS.2019.8711418

[8] Alipour S. Ninety Eight Per Cent Of Fake Or Lookalike IPhone Chargers Put Consumers At Risk Of Lethal Electric Shock And Fire, 2017. URL: https://www.electricalsafetyfirst.org.uk/media-centre/press-releases/2017/12/ninety-eight-per-cent-of-fake-or-lookalike-iphone-chargers-put-consumers-at-risk-of-lethal-electric-shock-and-fire/.

[9] Casassus B. Health agency reveals scourge of fake drugs in developing world, 2017. URL: https://www.nature.com/news/health-agency-reveals-scourge-of-fake-drugs-in-developing-world-1.23051.

[10] Hamelin N., Nwankwo S., El Hadouchi R. Faking brands': consumer responses to counterfeiting. Journal of Consumer Behaviour, 12(3), (2013) 159-170. https://doi.org/10.1002/cb.1406

[11] Zhang L. Platformizing family production: The contradictions of rural digital labor in China. The Economic and Labour Relations Review, 32(3), (2021). 341-359. https://doi.org/10.1177/10353046211037093
[12] Horstmann C. S., Cornell G. Core Java 2: Fundamentals (Vol. 1). Prentice Hall Professional, 2001.
[13] Martin R. Clean Architecture: A Craftsman's Guide To Software Structure And Design. U.S.A.: Pearson, 2017.
[14] Freeman E., Robson E., Bates B., Sierra K. Head First Design Patterns: A Brain-Friendly Guide. " O'Reilly Media, Inc.", 2004.
[15] Walls C. Spring in Action. New York: Manning Publications, 2018.
[16] Walls C. Spring Boot in Action. New York: Manning Publications, 2018.
[17] Grinev S. Mastering JavaFX 10: Build advanced and visually stunning Java applications. New York: Packt Publishing, 2018.
[18] Chaitanya S., JSON Tutorial, 2015. URL: https://beginnersbook.com/2015/04/json-tutorial/.
[19] Aravind M. Gson Library, 2017. URL: https://medium.com/@manuaravindpta/gson-library-b7d4ef0381e2.
[20] Maven: The Definitive Guide: The Definitive Guide – New York: O'Reilly Media, 2008.
[21] Schlager R. The OSI Model: simply explained. New York: CreateSpace Independent Publishing Platform, 2013.
[22] Gourley D., Totty B., Sayer M., Aggarwal A. HTTP: The Definitive Guide. New York: O'Reilly Media, 2002.
[23] Kurniawan B. How Tomcat Works. New York: Brainy Software, 2005.
[24] Aquino C., Gandee T. Front-End Web Development. New York: Big Nerd Ranch Guides, 2016.
[25] Kommadi B. IntelliJ vs Eclipse Complete IDE Comparison, 2019. URL: https://medium.com/@bhagvankommadi/hi-team-6d2dee22d8b2.
[26] Reed J. Blockchain: The Essential Guide to Understanding the Blockchain Revolution. New York: CreateSpace Independent Publishing Platform, 2016.
[27] Walker G. Block Header. A summary of the data in the block, 2016. URL: https://learnmeabitcoin.com/technical/block-header.
[28] Walker G. Merkle Root. A fingerprint for all the transactions in a block, 2016. – URL: https://learnmeabitcoin.com/technical/merkle-root.
[29] Canty R. Understanding Cryptography with RSA, 2020. URL: https://jryancanty.medium.com/understanding-cryptography-with-rsa-74721350331f.
[30] Nagpal A. How to create your own decentralized file sharing service using python, 2018. URL: https://medium.com/@amannagpal4/how-to-create-your-own-decentralized-file-sharing-service-using-python-2e00005bdc4a.

COMPLIANCE WITH ETHICAL STANDARDS

Conflict of Interest: The authors declare that the research was conducted in the absence of any conflicts of interest.

AUTHOR CONTRIBUTIONS

Conceptualization, [V.V.]; methodology, [V.V., O.P.]; validation, [M.N.]; formal analysis, [V.V.].; investigation, [O.P.]; resources, [O.P.]; data curation, [M.N.]; writing – original draft preparation, [M.N.]; writing – review and editing, [M.N.]; visualization, [O.P.] supervision, [R.R.]; project administration, [R.R.]; funding acquisition, [R.R.].

All authors have read and agreed to the published version of the manuscript.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ДЛЯ АУТЕНТИФІКАЦІЇ ДАНИХ НА ОСНОВІ БЛОКЧЕЙНУ

Вікторія Височька[1], Олег Прокіпчук[1], Марія Назаркевич[2]*, Роман Романчук[1]

[1]Кафедра інформаційних систем та мереж
Національного університету «Львівська політехніка»,
вул. Степана Бандери 12, м. Львів, 79013, Україна
[2]Кафедра радіофізики та комп'ютерних технологій
Львівського національного університету імені Івана Франка,
вул. Драгоманова 50, Львів, 79005, Україна

АНОТАЦІЯ

Вступ. В статті описано процес розроблення технології інформаційної підтримки перевірки автентичності товарів на основі блокчейн. В роботі здійснено дослідження швидкодії розробленої системи залежно від корегування чотирьох параметрів: складність майнінгу, залежність часу майнінгу від потужностей технічного забезпечення, залежність часу створення блоку від кількості транзакцій та залежність часу валідації блокчейну від кількості блоків.

Матеріали та методи. Аналіз складності майнінгу проводився в межах від 1 до 7 (на складності 8 та більше, майнінг на робочому пристрої відбувається занадто довго). Цей параметр визначає кількість нулів, що мають бути на початку хешу, щоб вважати роботу підтвердженою. Далі складність збільшує загальний час роботи у декілька разів. Це пов'язано із збільшенням кількості операцій, що необхідно виконати при майнінгу. Звісно майнінг також залежить від потужностей технічного забезпечення, які мають бути комплексом передових процесорів та відеокарт. Аналіз залежності часу майнінгу від частоти процесора проведений для складності майнінгу - 5.

Результати. Додаткова потужність процесора може значно скоротити час майнінгу. Наступним дослідженням є аналіз залежності часу створення блоку від кількості транзакцій. Основне ресурсне навантаження цього етапу є обчислення хеш-функцій для транзакцій для блоку, а також побудова хеш-дерева. З отриманих даних можна зробити висновок, що кількість транзакцій блоку є основним його ресурсним навантаженням. Це варто враховувати при виборі максимальної кількості транзакцій до блоку.

Висновки. Останнім дослідженням є виявлення залежності часу валідації блокчейну від кількості блоків. З отриманих даних випливає лінійна залежність часу валідації блокчейну від кількості блоків. Це сигналізує про те, що процес валідації не перевантажений ресурсозалежними операціями. При розширенні блокчейну з часом можна розраховувати на очікувану тривалість валідації блокчейну за лінійною залежністю.

Ключові слова – мережевий зв'язок, інформаційна система, контрафактний товар, технологія блокчейн, життєвий цикл продукту, інтелектуальна система пошуку інформації.

Received / Одержано	Revised / Доопрацьовано	Accepted / Прийнято	Published / Опубліковано
24 February, 2025	18 March, 2025	24 March, 2025	26 June, 2025