International Journal
of Applied Information Technology

http://journals.telkomuniversity.ac.id/ijait/

Building Private Blockchain For Recording Student Values Data in Environment Faculty of Engineering at Universitas Muhammadiyah, Ponorogo

Aslan Alwi [a, *], Moh. Bhanu Setyawan [b], Munirah [c]

[a,b,c] *Engineering Department, Universitas Muhammadiyah, Ponorogo*

elangbijak4@gmail.com, mohammad.setyawan@gmail.com, munirah.mt@gmail.com

ARTICLE INFO

Received August 15th, 2020
Revised October 22nd, 2020
Accepted September 16th, 2020
Available online March, 2nd, 2021

Keywords:
Blockchain, Ledger
Protocol, Block, Transaction**.**

ABSTRACT

Storing student grades in universities is a sensitive issue when it comes to the security of campus computer networks. In this paper, we propose a blockchain model to ensure the secure storage of student grade data. Each record of recording student grades by a lecturer can be seen as a record in a journal or ledger. All records in one of the departments in a faculty can be seen as a student grade journal. Modeling this data by viewing data records as a journal allows us to model a database for all records in an encrypted and distributed manner. Namely, a model that resembles the blockchain model. In this perspective, a blockchain network can be made in which each node is stored in a different direction. So if there are three departments within the faculty, a blockchain with three vertices can be made. Each node keeps a copy of the overall student grade journal data. So that each node can mutually verify the validity of transaction records in each department. Student value journals are structured in such a way that they form a blockchain that builds hashes that lock together. This modeling of student value records answers the problem of being vulnerable to student scores hacked by irresponsible parties both from within the campus and from outside the campus. Security follows the protocols in the blockchain architecture.

* Corresponding author at:
Engineering Departement, Universitas Muhammadiyah Ponorogo
Jl. Budi Utomo No. 10, Ponorogo, 63471
Indonesia.
E-mail address: elangbijak4@gmail.com

ORCID ID:
- First Author: 0000-0002-2253-7269
- Third Author: 0000-0002-4269-881X

https://doi.org/10.25124/ijait.v4i01.3122
Paper_reg_number IJAIT000040104 2020 © The Authors. Published by School of Applied Science, Telkom University.

1. Introduction

Blockchain is one component of the technology in the Industrial Revolution 4.0 and is an important issue for all countries in the world to be able to enter perfectly in the revolutionary era. Therefore, it is our duty as a nation to try to master this technology and break our own free and sovereign path in science and technology. According to [1] in his paper, there are mainly three types of blockchains: public (permissionless), consortium (public permission), and private. They possess different characteristics regarding who can access, write, and read the data on the blockchain. The data in a public chain can be viewed by all and anyone can join and contribute to both consensus (in theory) and changes to the core software.

Blockchain security is based on community validation to keep replicated ledger content synchronized across multiple users or authorities. Blockchain is also the technology that underlies digital currencies such as bitcoin. But on the other hand, the anonymity of the blockchain is also a cause for widespread concern [2].

As is the case with the application of blockchain in e-learning systems, by comparing with traditional e-learning systems based on a centralized database, using a distributed ledger and cryptographic technology from the blockchain, it is possible to permanently store educational achievements and ensure that they are trustworthy, secure and cannot be faked [3]. The application to this e-learning system also inspires the application of blockchain specifically to store student grades at universities.

Many researchers have developed blockchain-based systems in various sectors, for example, securing bank payments, or ensuring traceability in the food chain. The reason is that blockchain has the advantage of providing transparent and tamper-proof information exchange. Instead of using a centralized third party to verify transaction data, multiple actors, on a distributed basis, participate in a verification process called mining. The quality of the information shared is guaranteed through a comprehensive and rigorous verification process that follows certain calculations and updating rules [4]. It is hoped that the nature of blockchain will carry over to the application of blockchain to maintain the security of student grades at universities.

Research conducted by [5] for a blockchain model for drug distribution in the pharmaceutical industry also inspired our research especially for the creation of a blockchain model application interface for universities.

Blockchain as a technology, we can see its foundation in Satoshi's paper [6], we can see it in several ways. The first point of view is that we can view it as an encrypted distributed storage system. The second point of view it can be seen as a journal or a distributed ledger. A third perspective we can view it as a distributed database system and finally in a perhaps more technical way of looking at it as a network of distributed objects or a network of distributed agents.

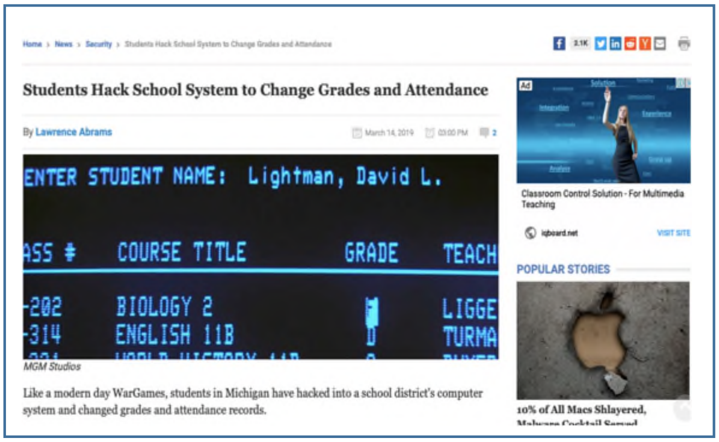
In this study, the basic format of writing student grade data into blockchain transactions refers to the basic form proposed by [7] which, if written in JSON format, is in the following form:

```
{           "previous-transaction-id": "FEDCBA987654321...",
            "owner-pubkey": "123456789ABCDEF...",
            "prev-owner-signature": "AABBCCDDEEFF112233..."          }
```

2. Problem Identification

As a first goal, information systems for student grades are an area for testing blockchain technology. There are many cases of students breaking into campus servers just to change their test scores to get the grades they want. Some examples of cases below provide facts that this is a reality that should be anticipated by the campus.

Figure 1 and Figure 2 provide news about students who broke into campus servers in a city abroad. All these events present a strong problem that the server or database that stores student value data is prone to be hacked by students. Therefore the blockchain system is expected to be a solution for storing student value data on campus.



Home › News › Security › Students Hack School System to Change Grades and Attendance

Students Hack School System to Change Grades and Attendance

By Lawrence Abrams March 14, 2019 03:00 PM 2

ENTER STUDENT NAME: Lightman, David L.
ASS # COURSE TITLE GRADE TEACH
-202 BIOLOGY 2 F LIGGE
-314 ENGLISH 11B D TURMA

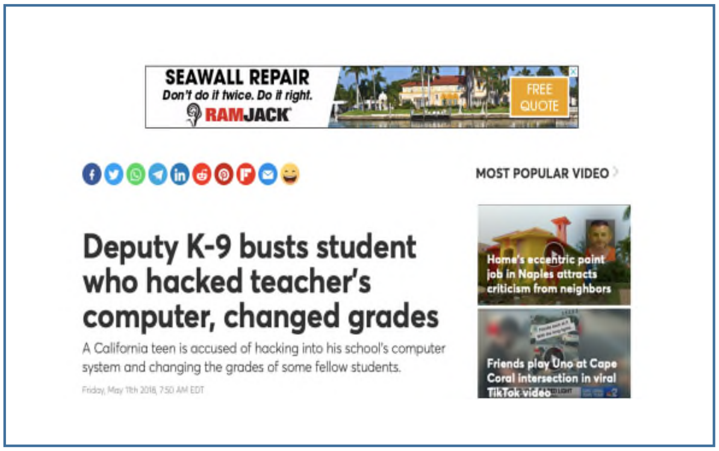
MGM Studios

Like a modern day WarGames, students in Michigan have hacked into a school district's computer system and changed grades and attendance records.

POPULAR STORIES

10% of All Macs Shlayered, Malware Cocktail Served

Figure 1 Case Example 1



MOST POPULAR VIDEO

Deputy K-9 busts student who hacked teacher's computer, changed grades

A California teen is accused of hacking into his school's computer system and changing the grades of some fellow students.

Friday, May 11th 2018, 7:50 AM EDT

Home's eccentric paint job in Naples attracts criticism from neighbors

Friends play Uno at Cape Coral intersection in viral TikTok video

Figure 2 Case Example 2

3. Research Methods

In general, the research flow mechanism that is built can be seen from the schema in Figure 3. The course of the research began by gathering information about the business plan of filling lecturer grades, then the metadata of student grades. From that information, a blockchain ledger meta data is created. In this case, the construction of columns for creating signatures (hash codes) and transaction columns. From this we get the blockchain data model.

Furthermore, making the protocol for adding blocks to the blockchain. Because the blockchain system is created as a private blockchain, the addition of blocks is in the form of assignments and authentication. Namely there are a number of lecturers who are given the authority to add blocks. However, each lecturer does not keep a copy of the entire blockchain, he only keeps a copy of the block he makes. The computers from departement that keep copies of the blockchain as a whole.

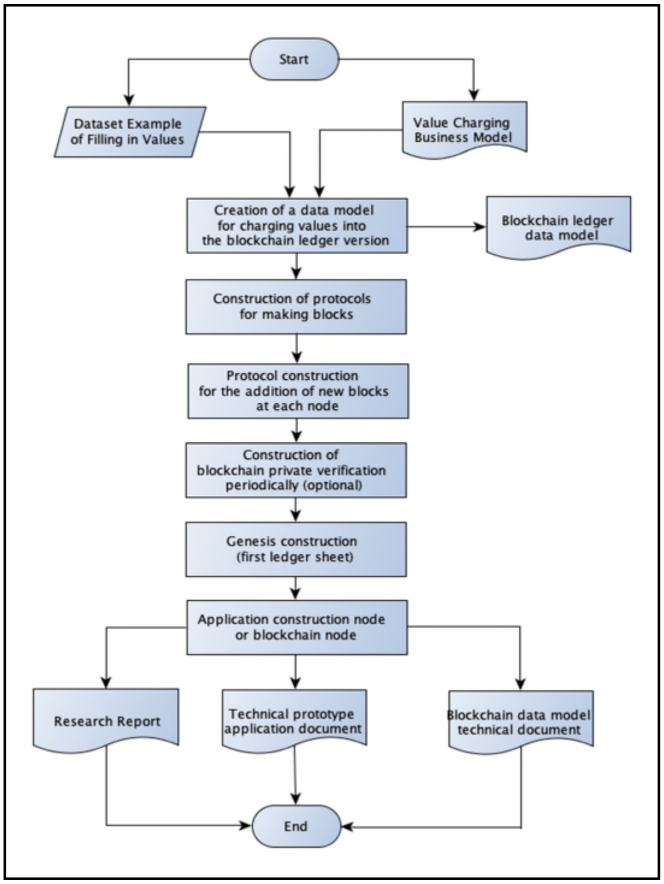

Start
Dataset Example of Filling in Values
Value Charging Business Model
Creation of a data model for charging values into the blockchain ledger version
Blockchain ledger data model
Construction of protocols for making blocks
Protocol construction for the addition of new blocks at each node
Construction of blockchain private verification periodically (optional)
Genesis construction (first ledger sheet)
Application construction node or blockchain node
Research Report
Technical prototype application document
Blockchain data model technical document
End

Figure 3 Research Schema

In this case, the computer of departement is located as an official node in the blockchain. Figure 4 below illustrates the simple architecture of the private blockchain.

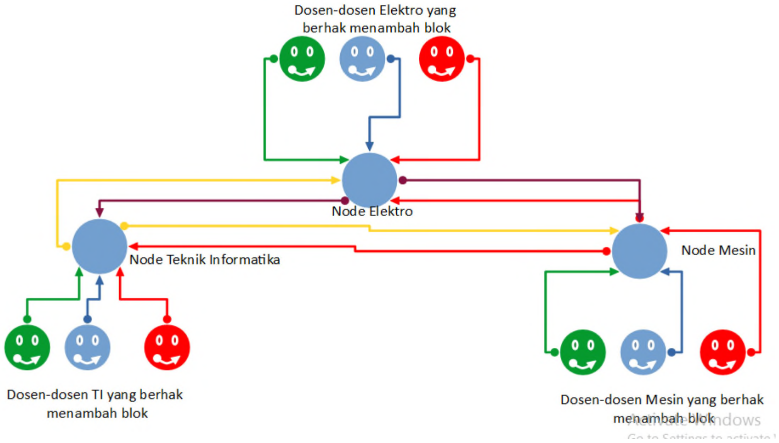

Dosen-dosen Elektro yang berhak menambah blok
Node Elektro
Node Teknik Informatika
Node Mesin
Dosen-dosen TI yang berhak menambah blok
Dosen-dosen Mesin yang berhak menambah blok

Figure 4 Planned Private Blockchain Architecture

Each lecturer is given the right to add a block. A block does not only consist of 1 transaction but may have more than 1 transaction. So that for one block construction, it can consist of n transactions. For example for 1 block is to contain all the data value of a learning class for 1 course. The hash process is done on the server side of the department computer system. Each lecturer only makes block candidates. Each candidate block is stored in the server's storage memory (as a pool).

Next is the construction of a protocol to periodically verify a copy of the blockchain. In this plan, each node verifies the meta file in the form of a hash chain only. Then create a genesis file, which is the first file in each node.

The final step is the creation of a server application at each node and wallet application at each lecturer. Each lecturer has a wallet application that sends proposals for adding blocks to the blockchain.

4. Results

In this chapter, several results have been achieved in this study. Some of the results achieved are as follows.

1. Blockchain specifications
2. The node specifications of the blockchain
3. Specification for authentication protocol for adding blocks
4. The Consensus Protocol Mechanism
5. The Node Registration Mechanism or Computer Registration Mechanism
6. Verification Mechanism for Lecturers Who Register
7. MVC architecture of the application node
8. Code for database access on the application node
9. Code to login to the application node
10. Code for the blockchain

Details of the results are as follows:

4.1. Blockchain Specifications

Blockhain is made as simple as possible to obtain better speed and security. Therefore, the blockchain referred to in this implementation is only an sql file that can be executed on any database server, but in case at this implementation is emphasized on the XAMPP MySQL or MariaDB server. Blockchain specifications are as follows:

1. Only a table in a database,
2. The table consists of 2 columns (fields),
3. The first column is the index number column of the block,
4. The second column is the encoded block column,
5. Each block column consists of a number of subfields and a payload-block,
6. Each payload-block consists of a number of transactions,
7. Each transaction consists of a number of sub-fields and payload-transactions.

The details of the blockchain file format in the sql file format that has been created are as follows.

```
-- Database: `blockchain`
--
```

```
-- --------------------------------------------------------
--
-- Struktur dari tabel `blockchain`
--
CREATE TABLE `blockchain` (
  `idblok` int(11) NOT NULL,
  `blok` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Indeks untuk tabel `blockchain`
--
ALTER TABLE `blockchain`
  ADD PRIMARY KEY (`idblok`);
--
-- AUTO_INCREMENT untuk tabel `blockchain`
--
ALTER TABLE `blockchain`
  MODIFY `idblok` int(11) NOT NULL AUTO_INCREMENT;
COMMIT;
```

The details of the blockchain file format in the json file format that have been created are as follows.

```
[
{"type":"header","version":"4.9.0.1","comment":"Export to JSON plugin for PHPMyAdmin"},{"type":"database","name":"blockchain"},{"type":"table","name":"blockchain","database":"blockchain","data":[]}
]
```

The details of the blockchain file format in the xml file format that have been created are as follows.

```
<?xml version="1.0" encoding="utf-8"?>
<!--
- phpMyAdmin XML Dump
- version 4.9.0.1
- https://www.phpmyadmin.net
- Host: 127.0.0.1
- Waktu pembuatan: 21 Agu 2019 pada 11.38
- Versi server: 10.4.6-MariaDB
- Versi PHP: 7.3.8
-->
<pma_xml_export version="1.0" xmlns:pma="https://www.phpmyadmin.net/some_doc_url/">
    <!--
    - Structure schemas
    -->
    <pma:structure_schemas>
        <pma:database name="blockchain" collation="latin1_swedish_ci" charset="latin1">
            <pma:table name="blockchain">
                CREATE TABLE `blockchain` (
                  `idblok` int(11) NOT NULL AUTO_INCREMENT,
                  `blok` text NOT NULL,
                  PRIMARY KEY (`idblok`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
            </pma:table>
        </pma:database>
    </pma:structure_schemas>

    <!--
    - Database: 'blockchain'
    -->
    <database name="blockchain">
        <!-- Tabel blockchain -->
    </database>
```

</pma_xml_export>

The details of the fields and sub-fields of blocks and transactions are stated in the following specifications.

1. Block index; States the block number. Is an automatic primary key and autoincrement.
2. Block; Blocks consist of encoding into text fields such as:
    - Hash from the previous block
    - Code from the blockchain version
    - ID code of the constructing node
    - Length of the payload-block size
    - Merkle tree hash of all transactions entered into the block
    - Payload-blocks are transactions that are loaded by a block
    - Hash from the current block
3. Transaction
    - Transaction code, can be a hash of the transaction itself
    - Length of this transaction

4.2. Blockchain Specifications

In this private blockchain architecture, a node that functions as a point in the blockchain that can add blocks to the blockchain is created as a separate web server. So in general, if there are 3 nodes that participate in the private bockchain, there are 3 stand-alone web servers.

In this architecture, the node is implemented as a XAMPP web server with a MySQL database, so there are 3 XAMPP servers, each of which stands alone. However, all the servers share the same blockchain database. When a node adds blocks in the MySQL database, it announces (broadcasts) to all other nodes so that the other node also adds the same block. Of course, with the addition rules that are verified as a blockchain protocol. Technically, the details of node specifications are as follows.

1. Each node is implemented as an XAMPP web server that includes a MySQL database server. The choice of implementation is because the private blockchain is still under development. As a consequence, this option is not very safe if it enters the actual implementation stage on the internet, so it can be replaced with other alternatives such as WAMP, MAMP or LAMP that can be posted in the cloud, or other types of web servers that are not Apache family.
2. Each node is implemented as a web application with MVC architecture.
3. Each node has two application sides,
    1. The first side of the node application is the system admin application of the node where it can add blocks and verify blocks. Also to authenticate users and other nodes in the blockchain environment. The admin application function of the node is as follows.
        - To add blocks,
        - To do manual verification other than automatic verification,
        - To perform manual authentication in addition to automatic authentication,
        - The node can turn on and turn off the automatic verification function,

- The node can see all the list of nodes listed in the blockchain as well as details that can be seen or allowed to be seen by the super admin,
- The node can see the fork if it exists and delete the fork manually or automatically.

2. The second side of the node application is the user application, which any authorized user (lecturer) can use to read the blockchain. The application node has the following functions:
    - As a page for searching blocks, either per block or a number of blocks (range blocks),
    - As a page for decoding blocks, i.e. translating blocks into views of a number of fields that are easily read by humans,
    - As a page to print a number of blocks,
    - As a page to export a number of data blocks to other file types, for example excel, xml, json, csv or sql file,
    - As a page to download the blockchain for users who are interested in doing manual verification,
    - Everyone can register as a node on the private blockchain by first getting approval from 51% or more of all blockchain members (nodes),
    - Everyone can register as a user instead of a node on the private blockchain by first obtaining approval of the node where the user is registering,
    - Each node has the right to add blocks inside the blockchain,
    - Each node has the obligation to verify the block for each additional block that occurs in the blockchain ecosystem,
    - Each node has the right to approve or reject someone to become a node,
    - Each node must install XAMPP or equivalent in order to install the blockchain application,
    - Each node has a database:
        a. Blockchain database (core)
        b. Supporting database;
    - Consists of user and node authentication tables,
    - Transaction table that has not or wants to be inserted into a block. This table functions as a transaction buffer that comes from the user or admin node itself.

4.3. Specifications for Adding Block Authentication Protocols

4.3.1. Registration Protocol as a Node

- The user registers on the system, giving a username and password,
- The system creates a dynamic hash of the username and password (user hash),
- The system only remembers the hash and removes the username and password. (The system not just hashing password but password and

username in spesific combination using automata creating from of pasword).

- The system then creates a user hash table with architecture:
  Id_user | user_hash | id_node_digital_signing | digitalsignaturefor_userhash.
- This table is immediately filled in until all user node id's are present on the system, but only systems whose signature digital is filled in for the first time.
- The system then creates a digital signature for the user hash, the system as the first digital signature maker.
- The system then broadcasts the hashuser table to all nodes and requests their signatures.
- Each node then signs dynamically according to the id_node, (to solve the problem if one day the node forgets the password so it changes the password, this remains unchanged, for verification, one only needs to look at the last hash).
- Node signature = signature of the previous node which is dynamically hashed using the current node password,
- Signature node = dynamic hash uses the node password for the user hash.
- The node then saves a copy of the user's hash table for itself as material for future signature verification.

4.3.2. Block Addition Verification Protocol

- When the user wants to add blocks, the user enters a username and password into the system. then the system performs a dynamic hash so that the user hash is formed,
- The system then signs the user's hash using its password and checks / compares it with the previous signature on the user's hash table that is held and then checks if valid,
- The system then sends the user hash of all nodes to request verification of other nodes, if 51% or more nodes say OK then the user may add a block or login,
- If it is less than 51% then the system asks the user to re-register (create a new username and new password and all the blockchain files he saved are overwritten by the new one), then the REGISTRATION PROTOCOL process starts a new one,
- What if a node verifies a password change so that it no longer agrees with the old signature that it has saved? this is solved by the concept of 51%. where the user only needs to re-register and the new REGISTRATION PROTOCOL process starts, or that when the node changes the password it also changes the user's hash signature in the user hash table it holds.

4.4. The Consensus Protocol Mechanism

The consensus mechanism for this private blockchain is based on university policy. Not based on a democratic way as on public blockchain consensus mechanisms, such as POW and POS. The details of the protocol for this private blockchain construction technically begin with the examination of the digital signature of the lecturer where the digital signature was previously formed by the system based on the username and password used to register using the registration protocol in part 4.3 above. This consensus is named as Proof of Signature (POSgn). The consensus details are as follows.

1. Users who wish to add value to the blockchain at a node are first asked for their username and password.
2. The node then reconstructs the user's digital signature using the username and password.
3. The node then checks whether the result, which is the user's digital signature, is registered in its hash table which was provided by the previous central system according to the registration protocol in part 4.3.
4. If not, the user is rejected, if registered, the node then broadcasts the user's username and password to all nodes (broadcast data is encrypted, according to the agreement on the encryption formula between nodes)
5. Each node then reconstructs the digital signature from the broadcast username and password. Then it checks if the digital signatures are registered in their respective hash tables. The hash table is sent or updated at any time in all nodes whenever a new user registers according to the registration protocol in part 4.3.
6. Each node then sends the check results to the node that broadcasts it first. If 51% of nodes claim to be registered in their respective hash tables then the nodes agree to add data to the blockchain by that user. If less than 51% then the user is rejected. Users can choose to undergo a registration protocol for new registrations.
7. Each user who successfully adds data is given point one in the user profile table in the system database.
8. If there is more than one user who wants to add data to the blockchain, the node checks the profile table of each user in the system database by sending a request to the central system. Users with additional data points according to the description of part 4.4 point 7 above are more then get the first priority to have their digital signature checked.
9. If there is more than one user who has the same number of points to add data as described in part 4.4 point 7 above, then the nodes randomly choose between them.
10. Consensus protocol completed.

4.5. The Node Registration Mechanism or Computer Registration Mechanism

Registration to become a node or a computer to become a node machine on a private blockchain network is based on university policy, not based on a democratic way like on a public blockchain. One of the policies is a policy on hardware and software specifications that can be nodes that guarantee the operation of all private blockchain protocols.

4.6. Verification Mechanism for Lecturers Who Register

Each lecturer who registers according to the registration protocol in section 4.3 point a receives a registration confirmation email in the email account registered in the lecturer database in the central system. Every lecturer who registers must click the confirmation link in the email sent.

5. Conclusions

We have just discussed three items out of the seven planned items, namely the Blockchain specifications, the node specifications of the blockchain and the specification for authentication protocol for adding blocks. The remaining four parts we plan to write in the next paper are MVC architecture of the application node, code for database access on the application node, code to login to the application node and code for the blockchain.

From the planned design of the blockchain it is hoped that a baby step infrastructure blockchain can be made which can be directly used for various purposes. This generic infrastructure is intended for the development of the private blockchain and blockchain consortium.

Some verification protocols are made different from blockchain in general which is public because it is not intended as a cryptocurrency infrastructure. But it is made more generic for a variety of versatile purposes for the implementation of blockchain technology on various problems.

Bibliography

[1] A. Hasselgren, K. Kralevska, D. Gligoroski, S. A. Pedersen, and A. Faxvaag, “Blockchain in healthcare and health sciences — A scoping review”, *Int. J. Med. Inform.*, vol. 134, February 2020, 104040.

[2] T. Lu, R. Yan, M. Lei, and Z. Lin, “AABN: Anonymity assessment model based on Bayesian network with application to blockchain”, *China Communications*, vol. 16, issue 6, pp. 55–68, 2019.

[3] C. Li, J. Guo, G. Zhang, Y. Wang, Y. Sun, R. Bie, “A Blockchain System for E-Learning Assessment and Certification”, *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2019.

[4] Y. E. Oktian, I. K. Singgih, and F. N. Ferdinand, “Serious Game for Blockchain Education Purposes (using Proof-of-Work consensus of Bitcoin)”, *5th International Conference on New Media Studies (CONMEDIA 2019)*, pp. 177–183, 2019.

[5] I. Haq and O. M. Esuka, “Blockchain Technology in Pharmaceutical Industry to Prevent Counterfeit Drugs”, *International Journal of Computer Applications,* vol. 180, no. 25, 2018.

[6] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” pp. 1–9, 2008.

[7] S. Peyrott, “An Introduction to Ethereum and Smart Contracts”, 2017.