Annals of Emerging Technologies in Computing (AETiC)
Vol. 5, No. 2, 2021

Review Article

Development and Evaluation of Blockchain based Secure Application for Verification and Validation of Academic Certificates

Elva Leka[1,2,*] and Besnik Selimi[1]

[1]South East European University, North Macedonia
el23618@seeu.edu.mk; b.selimi@seeu.edu.mk
[2]Polytechnic University of Tirana, Albania
*Correspondence: el23618@seeu.edu.mk

Received: 30th December 2020; Accepted: 7th March 2021; Published: 1st April 2021

Abstract: Academic degrees are subject to corruptions, system flaws, forgeries, and imitations. In this paper we propose to develop a blockchain smart contract-based application using Ethereum Platform, to store, distribute and verify academic certificates. It constitutes a trusted, decentralized certificate's management system that can offer a unified viewpoint for students, academic institutions, as well as for other potential stakeholders such as employers. The article describes the implementation of three main parts of our proposed solution that includes: verification application, university interface and accreditor interface. This application avoids administrative barriers, makes the process of deployment, verification, and validation of certificates faster, efficient, and more secure. Additionally, it offers confidentiality of the data by using AES encryption algorithm before creating transactions and allows bulk submission of multiple academic certificates.

Keywords: ***Blockchain; Ethereum; Smart Contract; Test Environment; Verification Application***

1. Introduction

Existing verification methods of academic degrees have some limitations relating to dependencies on third parties, being time consuming, ownership rights, availability, and costs [1, 2]. Many countries are faced with problems such as fake institutions, and fake degree certificates, where the United States lead with having over 2 million fake degrees in circulation [3-5].

Those issues can be avoided by utilizing more transparent and efficient technologies, such as Blockchain Technology. Its main characteristics are elimination of intermediaries, high level transparency, cost reduction and real time settlement [6, 7]. Blockchain does not involve any central authority, guaranteeing the integrity of transactions among peers.

The purpose of this study is to prove the feasibility of using Ethereum Blockchain [8,9] based smart contract [10, 11] to upload an academic certificate to a distributed ledger, verify it and make it accessible to a third party via 'smart contract' permissions.

A smart contract is a computer program intended to automatically execute specific actions based on set terms [12]. This will allow to define the structure of the certificates, and the identification of entity that can deploy, edit, and validate them. Programming language that is used to deploy smart contracts is called "Solidity" [13, 14].

The architecture of the proposed application and implementation details (Leka, 2020) are presented at our previous research article. It describes the proposed system, called BCert [15], which

Elva Leka and Besnik Selimi, "Development and Evaluation of Blockchain based Secure Application for Verification and Validation of Academic Certificates", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 22-36, Vol. 5, No. 2, 1st April 2021, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2021.02.003, Available: http://aetic.theiaer.org/archive/v5/v5n2/p3.html.

is used to store, distribute, and verify academic certificates. The main roles of this implementation are accreditation body, university, students, and employer. Meanwhile, this paper presents the architecture of the system and functionalities of the main components, where a ***university*** is responsible for issuing certificates with the valid information for the student, ***an accreditation body*** can validate a certificate and accredit a university, on the other hand ***an employer*** can check the validation and authenticity of a certificate.

This paper follows by describing the implementation part of the project, which contains three different applications:

(1) ***Verification application.*** It is responsible to verify whether the issued certificate is valid or not. The verified application will be deployed on a public web server, following all PWA (Progressive Web Application) [16] standards and protocols.
(2) ***University Interface.*** It is responsible for creating, signing, and issuing certificates to the network.
(3) ***Accrediting Interface.*** An accreditation body can issue a university by specifying its public address and additional information such as name. The university is marked as VALID and it can issue valid certificates. In case this university is found to be fraudulent, it can be marked as INVALID and it can no longer issue certificates.

Once a certificate has been added to the blockchain, it can no longer be removed, and every activity regarding this contract is publicly available. Due to its architectural design, blokchain technology provides complete immutability. Depending on how universities and accreditation bodies work together, a certificate can be verified later, or upon its insertion into the blockchain. Those applications connect to the same network and contract, thus sharing and accessing the same information and certificates.

The system is implemented in Ethereum Platform. We use ***Ganache*** for storing smart contracts, which is part of Truffle suite of Ethereum development tools. For the test environment, we have chosen ***Rinkeby***. It is an Ethereum testnet for tests and prototypes, being able to publicly access the test environment but not spend extra on transactions.

The structure of the paper is presented below. Section 2 introduces the related works. Next Section describes the proposed solution prototype and smart contracts deployment. Continuing with Section 4, which presents university application part and its interface, followed with verifier application in Section 5. Testing and Evaluations are conducted in Section 6. The last sections discuss future works and conclusions.

2. Relating works

Blockchain is a decentralized ledger that does not include other third parties for electronic transactions [17]. There are two types of blockchain implementation, private and public ones. In private blockchain number of stakeholders is limited and the speed of transaction is good. On the other hand, public blockchain have more expensive validation cost, unlimited number of members, very high energy is needed to be spent on consensus mechanisms [9, 12, 18]. As an open source blockchain-based platform, Ethereum, makes it easier to develop different decentralized applications [19]. Ethereum encourages peers to stay anonymous.

A variety of research and blockchain applications have been developed for educational purposes especially in certificate management and distributions. Moreover, blockchain technology can be used for university fee payments, student identity management, copyright management, evaluations of students, and e-learning interactions as well [20-23].

There are some proposed solutions that utilize blockchain to manage certificate distribution and validation. SmartCert [24] is a blockchain based digital credentials verification platform. It encrypts the certificates to provide transparency. The student must share the hash code with potential employer for certificate verification. SmartCert has some shortcomings in features, because it is vulnerable to attacks and has no clear methods of authenticity of parties.

In their article [25], the authors propose a framework which is used to verify educational certificates, using Hyperledger Fabrik. They identify security issues that need to be taken in consideration in process of verification in the blockchain, focusing on privacy, ownership, authentication, and confidentiality.

DocChain [26], is a semi-private blockchain solution used for issuances and verifications of degrees. The authors propose to use PoE (Proof of Existence) algorithm to achieve security. Their solution allows for bulk submission of multiple degree documents. On the other hand, authors of [27] present SPROOF which is a public permissionless approach used to issue, manage, and verify digital document, thus offering transparency and integrity of the certificate. Another proposed blockchain-based model solution [28] uses long-term verification of signature, for the verification of graduation certificates. Furthermore, to improve validity, confidentiality and security, the authors recommend using the SHA-256 algorithm.

University of Nicosia [29], EduCTX [30], UZHBC [31] have started their projects to issue and verify diplomas using blockchain technology. UNIC is the first university in the world to publish diplomas of all graduating students on the blockchain. It uses SHA-256 algorithm to create a hash from the certificate and intend to offer privacy, ownership, and integrity of the certificate. EduCTX is implemented on the open-source Arc Blockchain Platform and is based on processing, managing, and controlling credits (ECTS) that student gain for completed courses. University of Zurich uses a public Ethereum blockchain verification system to issue Diplomas. The UZHBC employs a smart contract for both issuance and verifications. Furthermore, the verification of the digital certificate does not rely on manual intervention by the university. The system ensures identity of digital certificates by cryptographic primitives.

Meanwhile, BlockCerts is the only fully developed and open-source implementation used for issuing blockchain verifiable records to students [32]. BlockCerts was developed by Massachusetts Institutes of Technology and Learning Machine [33]. In their solution, they propose to involve issuing, sharing and verification processes to manage the certificates registered on a blockchain network. BlockCerts has been adapted by The University of Rome "Tar Vergata" to issue digital diplomas [34]. BlockCerts does not offer a separate verification service for verifying certificates validity. Furthermore, it has some concerns relating security and privacy aspects, because the system does not have any registration process for issuers.

Although there are many issues related to the implementation of blockchain technology. General aspects that will have to be considered are: (1*) the environmental cost*, the amount of electricity that is needed to run complex code across many computers; (2) *the time aspect*, as blockchains grow, they might be slow and cumbersome; *(3) the trust aspect*, as it will take time for end users to make the trust leap and trust the technology, and *(4) the legal aspect* – the slow regulation of the value-based transfer of value over the Internet by the use of blockchain networks invites fraud [35, 36]. Furthermore, another concern to be considered is unsuitability of traditional Software Development Life Cycle (SDLC) models [37] for blockchain enabled smart contract-based applications. In their research article [37], the authors identify the issues and propose to develop new standard SDLC models, to be used for distributed blockchain applications.

The above-mentioned solutions do not provide details regarding authorization and confidentiality theme. Our proposed system utilizes Ethereum smart contracts and leverages the benefits of IPFS, to store the certificates in a decentralized file system. Up to our best knowledge, there is no other work proposing similar solution. We intend to ensure confidentiality and validity of the certificates. So, to provide confidentiality, the certificates will be encrypted with AES algorithm before creating the transaction. On the other hand, validity is ensured by signing the transaction with the private key of issuer university.

Furthermore, the system offers a user-friendly interface, making it easier to access certificates, without the need to create a private wallet, or without even knowing what blockchain technology is. Users will be able to access their certificate from different devices, having an internet connection, this includes old smartphones, personal computers, laptops, or even smart TVs, and with the introduction of a QR scanner, users will not even need to remember anything. With the use of optional features such as IPFS, PWA, Cameras and more, the user can access even more information, such as a physically scanned document, Photo ID, with less effort.

The system will offer other certain features such as real time online verification, third-party verification, usability, and revocation.

3. Proposed solution - Prototyping

The purpose of application: (1) This application should be able to connect to a series of peer-to-peer networks such as Ethereum, IPFS and other systems depending on future technology adoption; (2) It should be user friendly and should be supported by a wide range of devices as they will be used by users to easily access the data; (3) On the other hand this application will be used by institutions to assist in the development of the project and in the deployment of certificates; (4) Gather informative and usage data.

We have chosen ***Ethereum Blockchain*** to implement our solution, because this technology offers automation, ease of access and flexibility and deploying ***Smart Contracts*** will offer transparency, immutability, and security. The Smart contacts will be deployed in ***Remix Solidity*** Word processor/Ide [12, 13].

3.1. Proposed Architecture

The diagram in Fig. 1 shows a comprehensive system model.

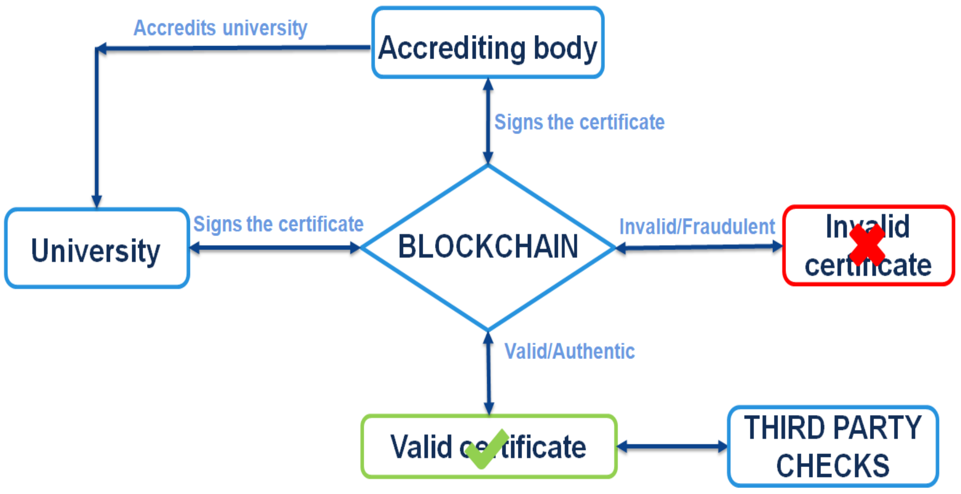


Figure 1. Architecture of issuing certificates

Issuing Certificates Interface. Once a student graduates or a certificate needs to be deployed, certain personnel is appointed to fill the needed forms and information. Once verified and approved, this information is signed by appropriate authorities, and if this university is accredited, the certificate is immediately added as APPROVED into the system, otherwise it is marked as PENDING. The encryption key is communicated to the student. This interface will be a typical web interface on a local network within the university.

Accrediting University Interface. An accreditation body takes care of adding universities into the approved list. The common attribute all universities have is a name or unique identifier. The accreditor can verify a university by specifying its public address and additional information such as name. The university is marked as VALID and it can issue valid certificates. In case this university is found to be fraudulent, it can be marked as Invalid and it can no longer issue certificates. The verification process is easy. User needs to put the certificate ID into the appropriate field, and the requested information will be shown. In upcoming work, other fields specific to the university can be implemented, but these will depend on further decisions of system implementation.

To verify the entered ID, in the client side can be used validator fields in front-end and backend. Furthermore, rate limiters, multiple layer protections and other techniques can be implemented to protect the system from DDoS attack.

On the other hand, the service rates are balanced between two main stakeholders of that service, graduated students, and employers. For students, the validity check of certificates is done at low cost and for employers, the system offers an easy authenticity and validity check of the certificate from trustable sources."

3.2. Development Environment

A development environment needs to be flexible, temporary, and easy to set up. Due to the nature of the project, a variety of different technologies will be used to have a usable product such as: blockchain, smart contracts, IPFS (InterPlanetary File System), PWA (Progressive Web Applications).

To develop the verifier application, following technologies need to be used:

- ***Web3JS***, which enables the client to communicate with the blockchain network and enable us to deploy, view, add, modify, and validate contracts on the blockchain.
- ***IPFS (InterPlanetary File System)*** to have more data, private logs, and efficiency. It helps to make the web faster and safer. IPFS, as a peer-to-peer hypermedia protocol, is used for storing and sharing data in a distributed file system [38, 39]. In our project it is planned to be used in storing photos, ID, avatars, or another biometric ID.
- ***PWA (Progressive Web Application)*** [16, 40] to use its benefits such as: *Cross-platform, easy to use, enhances security*. PWA is a type of application build and delivered using web technologies such as HTML, CSS, JavaScript and is intended to work on many platforms and devices such as mobiles and desktops. In this case, it will be used to make the verification process more accessible and available, without special hardware and additional software. PWA is delivered through common web protocols, but has reasonably more potential than a website, thus it needs to fulfil some requirements before being called a PWA, some of which are: use https, ability to be installable, respond offline, be fast and reliable.
- ***OpenShift Online***, which allows us to deploy containers form pre-build images or from other sources without having to worry about server infrastructure and without introducing extra costs.
- ***Docker.*** Docker is the defacto standard to build and share containerized applications from desktop to the cloud [8]. Docker allows us to have images that run on multiple containers and can be easily redeployed and scaled. We are using a 'dockerfile' hosted on github that contains all the commands a user could call on the command line to assemble an image and build feature that ***hub.docker.com*** comes with.

3.3. The deployed smart contracts

The smart contracts are self-executing programs that utilize blockchain technology to digitally enforce verifying or negotiation of a contract. Therefore, they offer credibility between contracting parties, without involving third parties.

Thus, in our proposed implementation, we have deployed two smart contracts, called ***'Certificate Issuer'*** and ***'Permissions and role manager'***, and have implemented different functions inside them, to make the application works properly.

'Certificate Issuer' smart contract is responsible for adding and retrieving certificates on the network. Functions that are included in this certificate are:

- ***addCert()*** – responsible for adding new certificates to the system
- ***getCert ()*** – responsible for retrieving certificates using the student ID

'Permission and role manager' smart contract is responsible for managing accounts on the network. It defines each action an entity can perform. Functions that are included in this certificate are:

- ***setMaster()*** *function* – sets an account which will have full permissions to the system
- ***getAcds()*** *function* – returns a list of accreditor accounts
- ***getAcd()*** *function* – returns specific details about an accreditor
- ***addAcd()*** *function* – adds new accreditor to the system
- ***switchAcd ()*** *function* – switches the state of the accreditor to valid or invalid.

- ***addUni()*** *function* – adds a new university to the network
- ***switchUni()*** *function* – defines whether the university can add new approved certificates

Algorithm 1 and algorithm 2 presents the pseudocodes of ***'Verifying student'*** and **'Sending signed transaction'** respectively.

Algorithm 1: Pseudocode of verifying student (Client Side)

```
function getCert (id) {
        ConnectToContract(network, address);
        return certificateData;
        }

function decryptData(data, encryptionKey) {
        decrypted = AES.decrypt(data, encryptionKey);
        if(decryptFailed) {
                alert("Error, failed decrypting");
        }
return decrypted;
}
```

Algorithm 2: Pseudocode of verifying student (Client Side)

```
function sendTransaction(StudentID, encryptedData, state, contractAddress){
        privateKey = promptUser("Enter Private Key");
        certEncodedABI = contract.methods.addCert(StudentID, encrypted Data, state).encodeABI( );
    transaction = {
                        from: universityAddress,
                        Nonce: nonce + 1,
                        to: contractAddress,
                        gas: 3000000,
                        Data: certEncodedABI
                        };
web3.eth.accounts.signTransaction(transaction, privateKey).then(signedTransaction => { sent Transaction =
web3.eth.sendSignedTransaction(signedTransaction.rawTransaction);

sentTransaction.on('confirmation', (confirmationNumber, receipt) => {
                        log(confirmationNumber, receipt)
                        })

                        sentTransaction.on('receipt', receipt => {
                                log(receipt)
                        })

                        sentTransaction.on(error, error => {
                                log(error)
                                })
                    }
 }
```

4. University Application

The university interface is responsible for creating, signing, and issuing certificates to the network. Apart from those discussed, there are other functionalities involved in this interface, such as taking care of safely encrypting and sending the data, distributing the passphrase and certificate details, as well as offering a seamless experience for the user.

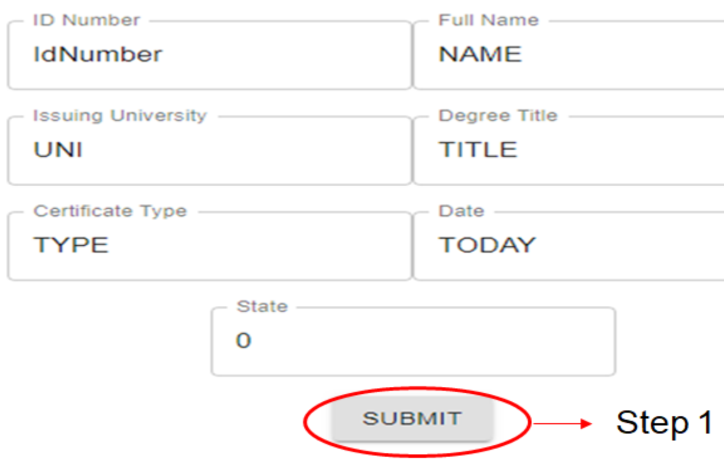
An important issue to concern is encrypting data, because of (1) sensitive information is transported to the blockchain and (2) how the transactions are made. There are multiple ways of encrypting data: (1) The first one is to use *'no encryption'* at all and considering that we will be dealing with sensitive data (this is immediately ruled out); (2) The next choice is to *handle encryption in the smart contract*, but the issue here is that the data is transferred unencrypted, thus it remains visible in the transaction history; (3) The next best solution is to *encrypt data before sending* it to the blockchain. At this stage, AES encryption is being used, where processed and formatted data is fed to the function handling encryption. After this stage, the transaction is built and ready to be signed and sent to the network. Only general information of student is visible, and other sensitive information has been hidden. The only way to decrypt this sensitive information is to use the key, which was encrypted with.

By using an encryption key, we can offer data integrity into the network. Keeping the system rather simple to use, we consider several ways of sharing this encryption key with the student. Firstly, the simplest way is through a printed certificate format, where a QR code with the encryption key is embedded with it. Secondly, considering that universities maintain contact with their students by establishing a connection through phone number and email, this information can be used for the key distribution. Upon certificate deployment, a SMS or an e-mail is sent to the student notifying them about the status of their certificate and the private key.

If the encryption key is lost, data is also permanently lost. This means that the information of the certificate is stored on the blockchain, but it is meaningless, because it cannot be decrypted. However, the certificate deployments are periodically saved into local computers of the university. Thus, with a student request, an extract copy of the certificate can be redeployed, or a new version if needed. The only downside of this method is that the student must manually request a redeployment or encryption key of the certificate.

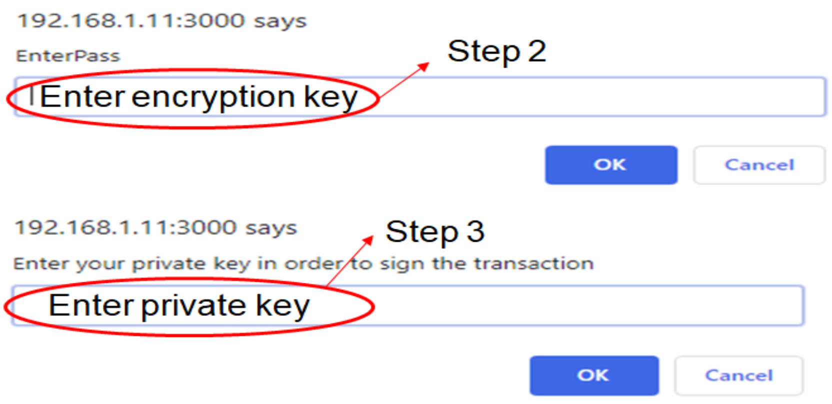
4.1. University Application Interface

So far, a basic interface has been created, as it is shown in Fig. 2, where the data is formatted and encrypted. Later a transaction is built, and after signing it with the private key, it is sent to the network. There is a config file called *config.js* responsible for the address used to sign the transaction, although a setting interface is planned to be developed, which will include other configuration options.



ID Number: IdNumber
Full Name: NAME
Issuing University: UNI
Degree Title: TITLE
Certificate Type: TYPE
Date: TODAY
State: 0

Figure 2. University Interface



192.168.1.11:3000 says
EnterPass
Enter encryption key

192.168.1.11:3000 says
Enter your private key in order to sign the transaction
Enter private key

Figure 3. Entering Encryption key and signing the transaction

After submitting the form, users are required to enter their unique encryption key. This key is used to encrypt the data, allowing only authorized person to access the academic records. The encryption key will be different for each certificate. The next step requires to sign the transaction using a private key of the issuer university, which should be kept secret as it is presented in Fig. 3. It is university's responsibility to add certificates to the blockchain, validate/invalidate them if needed, to store encryption keys and logs in a private server as well.

5. Verification Application Implementation

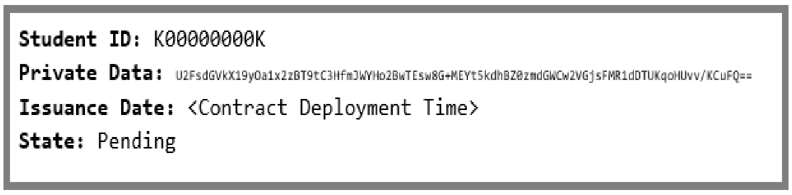
The verification application is supposed to verify whether the issued certificate is valid or not. On top of that it should show a set of basic student details, such as Student ID, Date of issuance and Certificate status. These details are supposed to show as little information as possible, while maintaining the privacy of student.

In case additional information is needed, it should be up to the student to allow access to this kind of information. This means that personal information should be somehow hidden from others and at the same time publicly accessible when needed. Such information might be: Full Student Name, University, degree, certificate type, and graduation date. The student is the one who allows access to this kind of information by providing the passphrase, which is the encryption key, that university issued to him/her by email, when the certificate was created. This key must be kept secret and in case the student loses it, the university has the responsibility to generate another one.

Furthermore, as an open-source Ethereum application, it makes the process of verification more transparent and trustworthy. Every employer/verifier should download the application from trusted sources and use it only on trusted networks. To conclude, this application would be intuitive, easy to use and transparent.

5.1. The Verifier Application

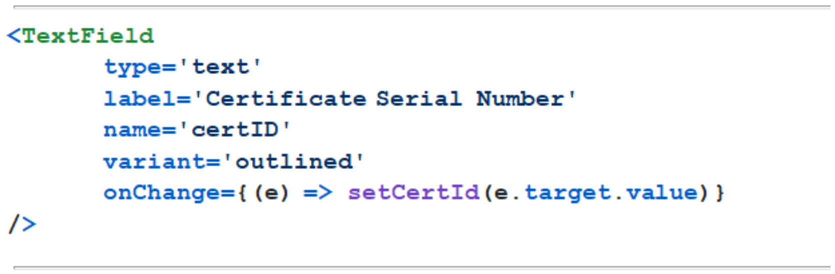
This version of the application depends only by one contract, which is the *'storing certificates' contract*. This contract is deployed on our test environment and after that we can start developing the verification application. To make development easier, an example certificate is deployed by default with the following details as are shown in Fig. 4:



```
Student ID: K00000000K
Private Data: U2FsdGVkX19yOa1x2zBT9tC3HfmJWYHo2BwTEsw8G+MEYt5kdhBZ0zmdGWCw2VGjsFMR1dDTUKqoHUvv/KCuFQ==
Issuance Date: <Contract Deployment Time>
State: Pending
```

Figure 4. Contract Deployment

The private data has been encrypted using AES and *crypto-js* library and can be decrypted using the same. But to decrypt, we need a decryption key, which will act as a kind of password. For this set of data, it is *"SomerandomPassPhrase"* and the data decrypted: *FullName, University, Degre, CertType, Date.*



```
<TextField
        type='text'
        label='Certificate Serial Number'
        name='certID'
        variant='outlined'
        onChange={(e) => setCertId(e.target.value)}
/>
```

Figure 5. The code that takes entered value of certificate serial number

First, a connection with the contract should be established. This is done by creating a new component called *'ConnectC'* which makes this possible. This component handles the configuration including smart contract address and network location, as well as connecting to the network itself.

The only module being used up to this point is Web3, which uses HTTP, ICP/WebSocket to allow us interacting with a local/remote Ethereum node.

Following, we need to make use of this connection. We import this component into the main App component and use it in conjunction with a form element. This way we can get all the data given the certificate serial number. We make use of React State features to store and update this information.

This piece of code takes the entered value (in our case the certificate serial number) and passes it to *certID* which stored the current *certificate ID* (also referred as serial number).

This function takes the submitted serial number to the ***ConnectC*** component and sees the returned certificate data to **cert** which is used to show ***unencrypted data.***

```
function handleSubmit(event){
        event.preventDefault();
        ConnectC(certId).then((r) => {setCert(r)});
        setDecData([]);
}
```

Figure 6. The Handle submit () function

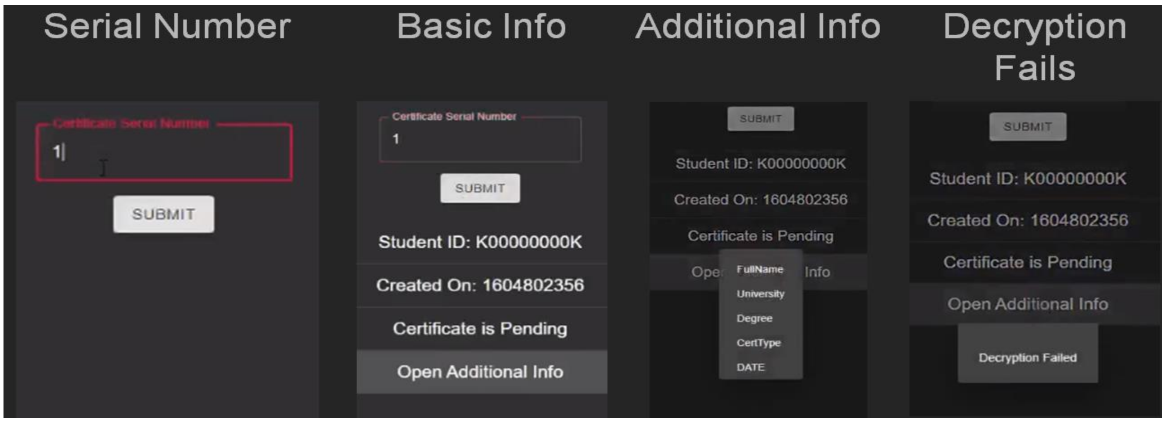
The Crypto-js module is used to encrypt data with ***AES*** encryption algorithm. The same module and encryption will be used to decrypt the data. Specifically, once the user requests additional data, he is prompted to enter a passphrase (decryption key). The following line takes care of decrypting the data:

```
AES.decrypt(encData, pass).toString(Utf8)
```

Furthermore, this piece of code takes the entered value (in our case the certificate serial number) and pass it. If the decryption is successful, the data is displayed on screen. Additional modules and libraries such as *Material-Ui* have been used to make the application more visually appealing, although at this point not much has been done regarding UI/UX.

5.2. Verifier Interface

The verifier interface will be used to verify whether the certificate is valid or not. It should be easy to access on most type of devices. Once the verification application is open, the interface is as in Fig.7, presented below. After the form has been submitted with the certificate serial number, a basic information regarding the contract is displayed, such as student ID, date of certificate's creation and status of the certificate. For further details, the student can use the last button *'Open Additional Info'*, which requires decryption key, to enable the user view additional information related to the degree, university that has issued that certificate, date of creation and so on. If the student enters a wrong decryption key, certificate verification process will fail.



Certificate Serial Number: 1

Student ID: K00000000K

Created On: 1604802356

Certificate is Pending

Figure 7. Verification Application Interface

At this point in time, this application performs basic functionality by manually entering serial ID. A QR code scan feature is planned to be implemented, thus making the process easier than it currently is. Some extra information will be added and removed from 'Basic information' and 'Additional' windows, as well as adding a window for IPFS photo ID.

Additionally, this application must become more trustworthy by implementing several warnings and following certain protocols to raise awareness about phishing attempts and clones.

6. Testing and evaluations

Due to the complexity and the type of data being handled by the project, a significant testing time will be required. This is because data being stored into the blockchain is immutable, and once those data are in the network, they cannot be removed. Furthermore, we need to make sure there are no missing spaces, security holes or loops in function execution. At the same time, this application needs to be publicly available, and transactions easily verifiable. Other things that will be tested are the *costs of transaction*, *certificate deployments* and *action costs.* For this reason, and more, a set of requirements must be met to publicly release the application. Some of these requirements will depend on the approaches and optional technologies chosen by the application developer.

Early test versions regarding security and core functions will be deployed on a private blockchain network. This enables us to quickly revert changes, monitor costs and raw data transfers. Moreover, every data transacted in this stage will be completely private and core functionality will be prioritized. This means the UI/UX might be far from the final product, and only a small group of people will have access to this early version.

Next stage consists of deploying the product to a public test network. At this stage core functionality should not have any issues or bugs. The focus of this stage is to bring the interfaces to an intuitive state, where users will have no problems realizing how to use the application. Additionally, several error trackers, statistical and feedback handlers will be made, allowing us to fine tune the application, ideally making it error free and easily upgradable, and prioritizing the enhancement of current features. Means of gathering information during testing:

1. A closed feedback system consisting of tester-developer communication.
2. Several costs, transactions, data, usage information, which are monitored by using appropriate software and means. Such software will depend on the developer of the application.

The first graph in Fig. 8.a, shows the time taken for a block to be included into the blockchain network. This is only for the past year and depends on global factors. The second graph 8.b shows the gas price in gwei. Gas price is another factor we cannot affect. It depends on global factors and the whole network status.

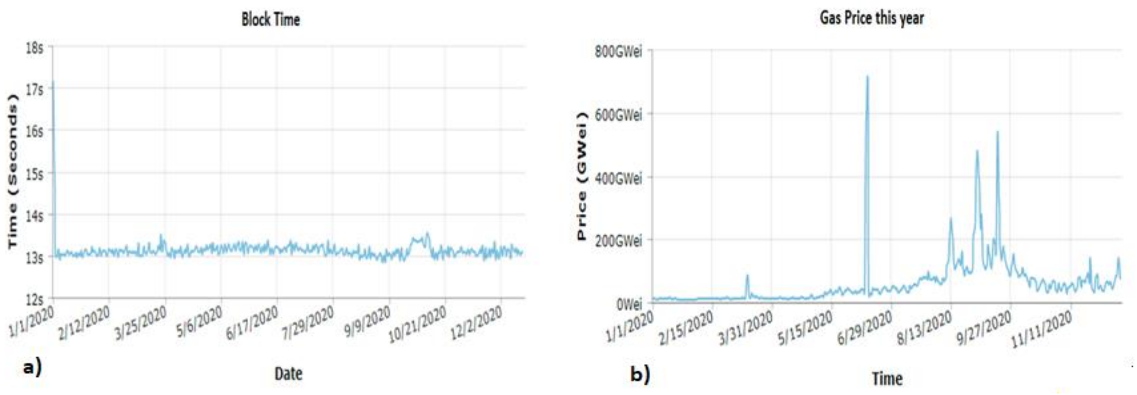


Figure 8. a) *'Block Time'* and b) *'Gas Price'* of Ethereum platform for year 2020 (*Data Source from EtherScan.io)*

Figures below will present *CPU*, *Memory* and *Network* usage, using **SysGaugeMonitor**. Those tests has been performed on a 3.4GHz Core on Windows 10. The memory usage, from the time (in seconds) the application is opened, until it is closed, is presented in Fig. 9.a. As seen, the memory usage at the end starts declining. Should be noted this benchmark also includes a chromium browser which is required to run a PWA, and will greatly differ from other devices, and future versions.

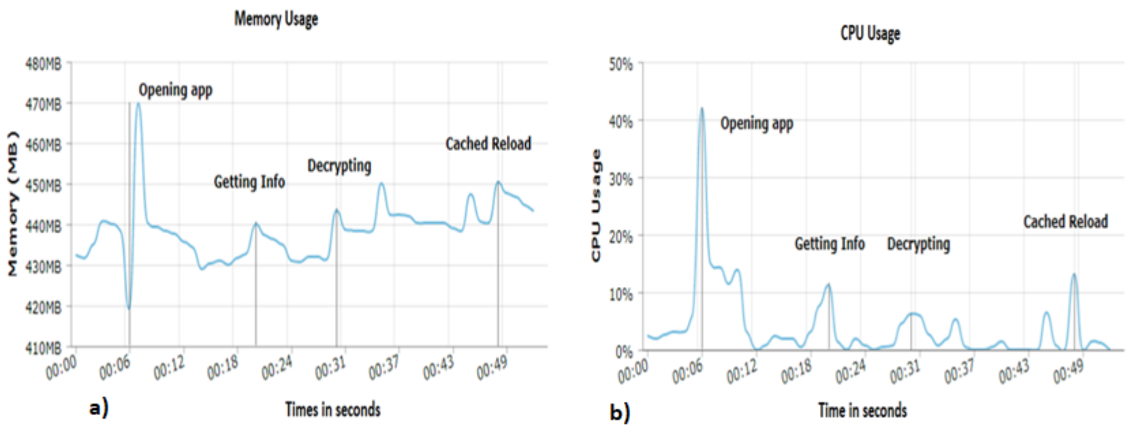


Figure 9. *a) Memory Usage* and *b) CPU usage* of Application

CPU usage is shown in Fig.9.b . Values may be different on other devices as well, however as this graph shows, the application can run smoothly on low-end devices, given it fits the minimum requirements of the final product.

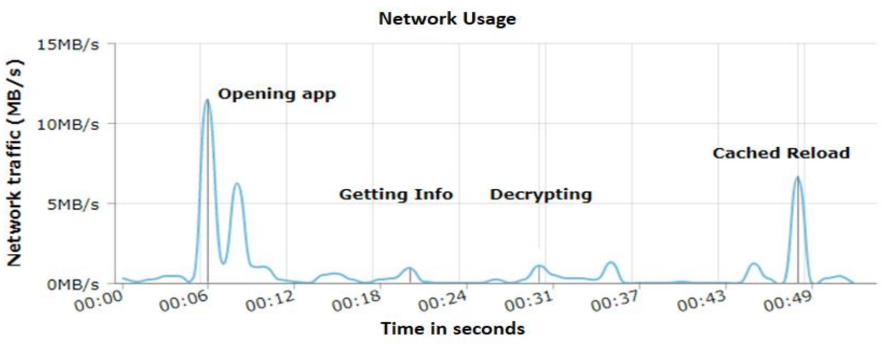


Figure 10. Verification Application Interface

The graph in Fig.10 represents the network usage this application utilizes. Due to the small size of the application, this is relatively low and barely measurable. A real difference can be seen on very slow devices or network connections.

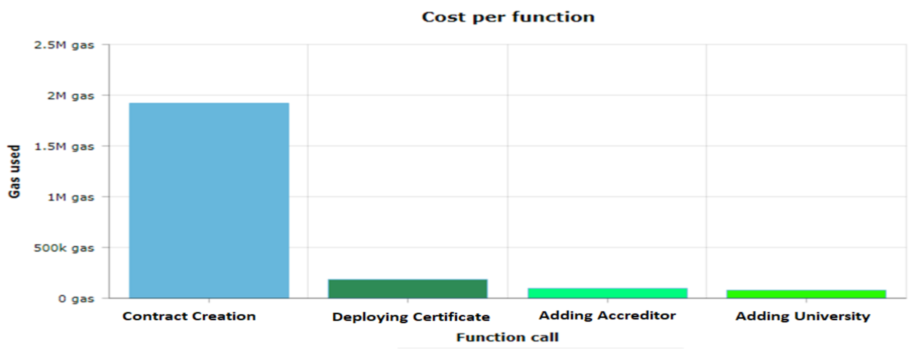


Figure 11. Verification Application Interface

The above graph in Fig.11 shows the cost in gas in order to perform an operation. These metrics include a single execution. Sequential and parallel executions will have different outcomes depending on their combination. Converting this to fiat currency depends on gas price, which is affected by global factors. The *etherscan APIs tool* can specifically be used to make the convertation to fiat currency.

Fig. 12 shows the ability to parallelize requests and add certificates in bulk. This part of the graphic highly depends on the first two graphics, as blocktime and gas price are the main factors that determine the time for the transaction (certificate addition) to be confirmed. We have taken the best (highest and fastest) price at the time of the benchmark. (according to Etherscan gastracker API).

Adding a small number of certificates requires almost no parallelization, however the more we add, the heavier the parallelization becomes. In this example we are adding 6 certificates at a time through 10 accounts simultaneously, which adds up to 60 certificates at once. This will require finer tuning to get right. We specify a block time on our test network close to mainnet with fast gas price preset. Then we issue certificates on this network. This number is expected to improve, however it will be up to the university to decide (through a config panel). Should be noted that the more data we are adding, the more expensive the process becomes, and should be careful this process does not exceed the account balance limit.

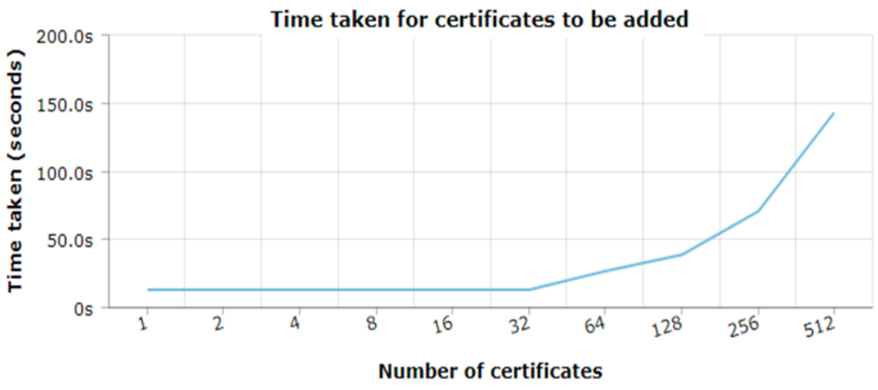


Figure 12. Parallelization of Certificates to be added into the blockchain network

7. Future Development

This section presents the future development of the proposed implementation regarding verifier, university, and accreditor applications as follow.

Verifier Application: On the verifier application, the user currently needs to manually enter a serial number and encryption key. Future includes adding a QR code scanner directly into the application, and the student can present a QR code to the verifier, thus the verification process will be instant without having to manually enter a serial number or passcode if preferred. Another plan is to add features to increase application usage awareness, so that everyone gets to know how to use it properly and not get tricked by off-market fake applications or clones.

University Application: The next most important feature to be added is a configuration panel, where each university can tweak the application settings according to their way of operating, such as the way the encryption keys are distributed, custom gateways, a history of deployed certificates with their respective state (pending, confirmed, declined), transaction hash and other important details. On top of that, depending on the parallelization technique used, the university might change the maximum number of certificates added at once to fit their requirements. Additionally, it is planned an import/export file, thus the certificates can be added in bulk. Furthermore, these certificates can be generated and signed on different computers. An example would be that a certificate is generated on personal computers, which may belong to authorized personnel, and can be later signed and deployed on the computers belonging to the university. The third feature planned is the way to distribute the encryption key, this will depend on how the universities establish the connection to the students, however an email distribution should be the default.

Accreditor Application: Updates on this application are similar the university application, they will include a configuration panel, import/export feature, and a distribution method. A unique feature is the ability to add multiple accounts per university, so each university can use multiple accounts to add certificates. An important feature would be the ability to invalidate or suspend universities, in case of an ongoing investigation or other reasons.

8. Conclusions

The system we propose will be a user-friendly application, which should be easy to use within a few clicks. The output of the implementation will be: (1) an application with the essential features to be used in the testing phase, as well as other beta features to get more feedback on what users prefer. It should offer transparency to fulfil our requirements; (2) A university management interface used by the accrediting body, which will be used to add, validate, and invalidate universities as well as certificates; (3) A certificate management interface used by the universities. It will be responsible for the deployment of new contract, one by one or in bulk; (4) A verification application widely available. This will be used to verify whether the certificate is valid or not. It should be easy to access on most types of devices (Phone, Personal Computer, Workstation, Laptops, Servers)

The smart contract will sit in the backend and will handle the inner functions of the final product, such as the data structure, various functions, and permissions. Every piece mentioned above will interact specifically with this smart contract.

The main plan is to add some parallelization, so the universities can deploy hundreds of certificates at once. Other additions include the use of IPFS, which will be used to store additional objects and files that belong to the student. They might be scans of the students ID, student grades, photo and other documents deemed necessary. Additional changes will depend on the features mentioned above. And of course, additional user interface updates will be issued, as well as updates that make it easier to use the application, and other issues that arise during testing.

References

[1] Osman Ghazali and Omar S. Saleh, "A Graduation Certificate Verification Model via Utilization of the Blockchain Technology", *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, Print ISSN: 2180-1843, Online ISSN: 2289-8131, pp. 29-34, Vol, 10, No.3-2, 2018. Published by: Universiti Teknikal Malaysia Melaka (UTeM), Available: https://journal.utem.edu.my/index.php/jtec/article/view/4707.

[2] Ankit Singhal and R.S Pavithr, "Degree Certificate Authentication using QR Code and Smartphone", *International Journal of Computer Applications*, Print ISSN: 0975-8887, pp.38-53, vol. 120, no.16, June 2015. Published by IJCA Journal, DOI: 10.5120/21315-4303.

[3] Jayesh G. Dongre, Kishore. T. Patil, Sonali M. Tikam and Vasudha B. Gharat, "Education Degree Fraud Detection and Student Certificate Verification using Blockchain", *International Journal of Engineering Research & Technology (IJERT)*, ISSN: 2278-0181, pp. 300-303, Vol. 9, No. 7, July 2020, DOI: 10.17577/IJERTV9IS070156.

[4] Anita Indu, "Implementation of Blockchain Technology in Education Sector: A Review", *International Journal of Research and Analytical Reviews (IJRAR)*, Online ISSN: 2348-1269, Print ISSN: 2345-5138, pp. 351-355, Volume 6, Issue 2, June 2015. Available: http://ijrar.com/upload_issue/ijrar_issue_20543750.pdf.

[5] Evelyn Garwe, "Qualification, Award and Recognition Fraud in Higher education in Zimbawe", *Journal of Critical Studies in Education*, ISSN 2162-6952, pp. 119-135, Vol. 5, No. 2, May 2015, DOI: 10.5296/JSE.V5I2.7456, Available: www.microthink.org/journa/index.php/jse/article/view7456/6151.

[6] Mahdi H. Miraz and Maaruf Ali, "Application of Blockchain Technology beyond Cryptocurrency", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 1-6, Vol. 2, No. 1, 1st January 2018. Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2018.01.001, Available: www.aetic.theiaer.org/archive/v2/v2n1/p1.html.

[7] Jingasha Dalal, Meenal Chaturevdi, Himani Gandre and Sanjana Thombare, "Verification of Identity and Educational Certificates of Students Using Biometric and Blockchain", *Proceedings of 3rd International Conference on Advances in Science and Technology (ICAST)*, 2020. Available: https://ssrn.com/abstract=3564638.

[8] Kentaroh Toyoda, Koji Machi, Yataka Ohtake and Allan N. Zhang, "Function – level Bottleneck Analysis of Private Proof-of-Authority Ethereum Blockchain", *IEEE Access*, ISSN: 216-3536, pp. 141611- 141621, Volume 8, July 2020, DOI: 10.1109/ACCESS.2020.3011876, Available: https://ieeexplore.ieee.org/document/9146870.

[9] Y. Hao, Y. Li, X. Dong, L. Fang, and P. Chen, "Performance Analysis of Consensus Algorithm in Private Blockchain", *IEEE Intelligent Vehicles Symposium (IV)*, ISSN: 1931-0587, Vol. 8, pp. 280–285, 2018, Published by IEEE, DOI: 10.1109/IVS.2018.8500557, Available: https://ieeexplore.ieee.org/document/8500557.

[10] Alessio Meneghetti, Tommaso Parise, Massimiliano Sala and Daniele Taufer, "A survey on Efficient Parallelization of Blockchain-based Smart contracts", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 9-16, Vol. 3, No. 5, 15th December 2019, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETic.2019.05.002, Available: http://aetic.theiaer.org/archive/v3/v3n5/p2.html.

[11] Marten Risius and Kai Spohrer, "A blockchain Research Framework: What We (don't) Know, where we Go from Here, and How we will Get there", *Business and Information Systems Engineering*, ISSN: 1867-0202, pp. 385-409, Vol. 59, Issue 6, December 2017. Published by: Springer, DOI: 10.1007/s12599-017-0506-0, Available: https://link.springer.com/article/10.1007/s12599-017-0506-0.

[12] Gustavo A. Oliva, Ahmed E. Hassan and Zheng Ming (Jack) Jiang, "An exploratory study of smart contracts in the Ethereum blockchain platform", *Empirical Software Engineering*, Print ISSN: 138-3256, Online ISSN:

1573-7616, pp. 1864-1904, Vol 25, No. 3, 2020, Published by SpringerLink, DOI: 10.1007/s10664-019-09796-5, Available: https://link.springer.com/article/10.1007%2Fs10664-019-097.

[13] Ming Li, Jian Weng, Anjia Yang, Jiasi Weng and Yue Zhang, "Towards Interpreting Solidity Smart contract: An Automatic and Practical Realization", *IEEE Transactions on Services Computing*, ISSN: 1939-1374, 2020, Published by: Cryptology ePrint Archive, Available at: https://eprint.iacr.org/2020/574.pdf.

[14] Purathani Praitheeshan, Lei Pan, Jianshan Yu, Joseph Liu and Robin Doss, "Security Analysis Methods on Ethereum Contract Vulnerabilities - A Survey", *CoRR, arXiv:1908.08605v3 [cs.CR],* September 2020, Published by ArXIV, Available at: https://arxiv.org/pdf/1908.08605.pdf.

[15] Elva Leka and Besnik Selimi, "BCERT – A decentralized Academic Certificate System Distribution Using Blockchain Technology*", International Journal on Information Technologies and Security,* ISSN: 1313-8251, pp. 103-118, Vol. 12, No. 4, December 2020, Available: https://ijits-bg.com/contents/IJITS-N4-2020/2020-N4-09.pdf.

[16] Tim A. Majchrzak, Andreas Biorn-Hansen and Tor-Morten Gronli, "Progressive Web Apps: The Definite Approach to Cross-Ptaform Development?", *Proceedings of 51st Hawaii International Conference on System Sciences*, ISBN: 978-0-9981331-1-9, pp.5735-5744, January 2018, DOI: 10.24251/HICSS.2018.718, Available: http://hdl.handle.net/10125/50607.

[17] Gavin Wood, "Ethereum: A secure Decentralized Generalized Transaction Ledge", *Ethereum Project Yellow Paper*, *EIP-150 Revision*, pp.1-32, 2017, Available: https://gavwood.com/paper.pdf.

[18] John M. Madellin and Mitchell A. Thornton, "Consideration of Quality Attribute Tradeoffs of the Blockchain Pattern in the Software Development Process", *Annals of Emerging Technologies in Computing (AETiC),* Print ISSN: 2516-0281, Online ISSN: 2516-029X, Vol. 3, No. 4, pp. 15-27, DOI: 10.33166/AETiC.2019.04.002, Available: http://aetic.theiaer.org/archive/v3/v3n4/p2.html.

[19] Erinc Karatas, "Developing Ethereum blockchain-based document verification smart contract for Moodle Learning management System", *International Journal of Informatics Technologies*, pp. 399-406, Vol. 11, Issue 4, October 2018, DOI: 10.17671/gazibtd.452686, Available: https://files.eric.ed.gov/fulltext/ED594770.pdf.

[20] Dinesh Kumar K, Senthil P and Manoj Kumar D. S, "Educational Certificate Verification System Using Blockchain", *International Journal of Scientific and Technology Research (IJSTR),* ISSN: 2277-8616, pp. 82-85, Volume 9, Issue 03, March 2020, DOI: Available: https://www.ijstr.org/final-print/mar2020/Educational-Certificate-Verification-System-Using-Blockchain.pdf.

[21] Muhamed Turkanovic and Blaz Podgarelec, "Signing Blockchain Transactions using Qualified Certificates", *IEEE Internet Computing*, Print ISSN: 1089-7801, Online ISSN: 1941-0131, Vol. 24, Issue 6, pp. 37-43, December 2020, Published by IEEE, DOI: 10.1109/MIC.2020.3026182, Available: https://ieeexplore.ieee.org/document/9206057.

[22] Ebin Mathew, Maria Paulson, Reshma Joy and Jisha P. Abraham, "E-certificate Generation using Blockchain", *International Journal of Computer Trends and Technology (IJCIT),* Print ISSN: 2349-0829, Online ISSN: 2231-2803, pp.70-73, Volume 68, Issue 3, March 2020, Available: https://www.ijcttjournal.org/2020/Volume-68%20Issue-3/IJCTT- V68I3P114.pdf.

[23] Bakri Awaji, Ellis Solaiman and Adel Albshri, "Blockchain-Based Applications in Higher Education: A Systematic Mapping Study", *Proceedings of 5th International Conference on Information and Education Innovations*, pp. 96-104, July 2020, Published by ACM International Conference proceedings Series (ICPS), DOI: 10.1145/3411681.3411688, Available: https://dl.acm.org/doi/10.1145/3411681.3411688.

[24] Tarek Kanan, Ahamad Turki Obaidat and Majduleen Al-Lahham, "SmartCert Blockchain Imperative for Educational Certificates", *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, Aman, Jordan, Print ISBN: 978-1-5386-7943-2, Online ISBN: 978-1-5386-7942-5, pp. 629-633, 2019. Published by IEEE, DOI: 10.1109/JEEIT.2019.8717505, Available: www.ieeexplore.ieee.org/document/8717505.

[25] Omar S. Saleh, Osman Ghazali and Muhammad Ehsan Rana, "Blockchain based Framework for Educational Certificates Verification*", Journal of Critical Reviews (JCR),* ISSN: 2394-5125, Vol. 7, No. 3, pp. 79-84, 2020, Published by Multidisciplinary Review Journal, DOI: 10.31838/jcr.07.03.13, Available: http://jcreview.com/fulltext/197-1583403182.pdf?1584339148.

[26] Saqib Rasool, Afshan Saleem, Muddesar Iqbal, Tasos Dagiuklas, Shahid Mumtaz *et al.*, "DocsChain: Blockchain based IoT Solution for Verification of Degree Documents", *IEEE Transactions of Compational Social Systems,* ISSN (CD-ROM): 2373-7476, Online ISSN: 2329-924X, PP(99):1-11, June 2020, Published by IEEE, DOI: 10.1109/TCSS.2020.2973710, Available: https://ieeexplore.ieee.org/document/9090855.

[27] Clemens Brunner, Fabian Knirsch and Dominik Engel, "SPROOF: A platform for issuing and verifying documents in a public blockchain", *5th International Conference on Information Systems Security and Security*, ISBN: 978-989-758-359-9, ISSN: 2184-4356, Vol. 1, pp. 15-25, 2019, Prague, Czech Republic, Published by Scitepress Digital Library, DOI: 10.5220/0007245600150025.

[28] Tomasz Hyla and Jerzy Pejas, "Long-term verification of signatures based on a blockchain", *Computer & Electrical Engineering,* Print ISSN: 0045-7906, Online ISSN: 1879-0755, pp. 106523-106535, Vol. 81, 2020, Published by Elsevier Ltd, DOI: 10.1016/j.compeleceng.2019.106523, Available: https://www.sciencedirect.com/science/article/abs/pii/S0045790618327381.

[29] Marinos Themistocleous, Klitos Christodoulou, Elias Iosif, Soulla Louca and Demetrios Tseas, "Blockchain in Academia: Where do we stand and where do we go"?, *Proceedings of the 53rd Hawaii International Conference on System Sciences*, ISBN: 978-0-9981331-3-3, pp. 5338-5347, January 2020, DOI: 10.24251/HICSS.2020.656, Available: http://hdl.handle.net/10125/64398.

[30] Muhamed Turkanovic, Marko Holbl, Kristjan Kosic and Marjan Hericko, "EduCTX: A Blockchain-Based Higher Education Credit Platform", *IEEE Access*, ISSN: 2169-3536, Vol. 6, pp. 5112-5130, 2017, DOI: 10.1109/ACCESS.2018.2789929, Available: http://www.ieeexplore.ieee.org/document/8247166.

[31] Jerinas Gresch, Bruno Rodriegues, Eder Scheid, Sail S. Kanhere and Burkhard Stiller, "The Proposal of a Blockchain-based Architecture for Transparent certificate Handling", *1st Workshop on Blockchain and Smart Contract Technologies (BSCT 2018), BIS 2018*. Lecture Notes in Business Information Processing, by Springer, Cham, Print ISBN: 978-3-030-04848-8, Online ISBN: 978-3-030-048449-5, Vol. 339, pp. 185-196. DOI: 10.1007/9783-030-0484905_16, Available: https://www.springer.com/us/book/9783030048488.

[32] Carmen Holotescu, "Understanding blockchain opportunities and challenges", *International Scientific Conference on eLearning and Software*, Vol.4, pp. 275-283, 2018, Bucharest, Romania, Published by: Editura Universitara, ISSN: 2066-026X, DOI: 10.12753/2066-026X-18-253.

[33] Yeray Mezquita, Roberto Casado, Alfonso Gonzalez-briones, Javier Prieto and Juan Manuel Corchado, "Blockchain Technology in IoT Systems: Review of the Challenges", *Annals of Emerging Technology in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 17-24, Vol. 3, No. 5, 15th December 2019, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETIC.2019.05.003, Available: http://aetic.theiaer.org/archive/v3/v3n5/p3.html.

[34] Guendaline Capece, Nathan Levialdi Ghiron and Francesco Pasquale, "Blockchain Technology: Redefining Trust for Digital Certificates", *Sustainability*, ISSN: 2624-8115, pp.8592-8604 , Vol 12, No.21, 2020, Published by MDPI, DOI: 10.3390/su12218952, Available: https://www.mdpi.com/2071-1050/12/21/8952/htm.

[35] Halvdan Haugsbakken and Inger Langseth, "The Blockchain Challenge for Higher Education Institution", *European Journal of Education,* ISSN (print): 2601-8616, ISSN online: 2601-8624, pp 41-46, Volume 2, Issue 3, 2019, DOI: 10.26417/ejed.v2i3.p41-46, Available: http://journals.euser.org/index.php/ejed/article/view/4503.

[36] Artyom Kosmarski, "Blockchain Adoption in Academia: Promises and Challenges", *Journal of Open Innovation: Technology, Market, and Complexity*, ISSN: 2199-8531, pp.117-132, Vol. 6, Issue 4, September 2020, DOI: 10.3390/joitmc6040117, Available: https://mdpi.com/2199-8531/6/4/117/htm.

[37] Mahdi H. Miraz and Maaruf Ali, "Blockchain Enabled Smart Contract Based Applications: Deficiencies with the Software Development Life Cycle Models", *Baltica Journal*, Vol. 33, Issue 1, 20th January 2020, ISSN: 0067-3064, pp. 101-116, Available: http://www.balticajournal.com/baltica/index.php/jTracker/index/IL1qQ.

[38] Anupam Tiwari and Usha Batra, "IPFS enabled blockchain for smart cities", *International Journal of Information Technology*, ISSN: 2511-2112, pp. Vol. 33, Issue 4, June 2020, Published by: SpringerLink, DOI: 10.1007/s41870-020-00568-9, Available: https://link.springer.com/article/10.1007/s41870-020-00568-9.

[39] Hsiaoshan Huang, Tian Sheuan Chang and Jhihyi Wu, "A secure File Sharing System Based on IPFS and Blockchain", *2020 2nd International Electronics Communication Conference (IECC 2020), Singapore*, pp 96-100, July 2020, Published by ACM International Conference Proceedings, ISBN: 978-1-4503-8993-8, DOI: 10.1145/3409934.3409948, Available: https://dl.acm.org/doi/10.1145/3409934.3409948.

[40] Dirk – Jan Rensema, "The Current State of Progressive Web Apps Thesis", *Krislstad Business Schools*, 2020. Available: www.diva-portal.org/smash/get/diva2:1449286/FULLTEXT01.pdf.