Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2020 SCPE. Volume 21, Issues 4, pp. 673–688, DOI 10.12694:/scpe.v21i4.1806

FORGERY PROTECTION OF ACADEMIC CERTIFICATES THROUGH INTEGRITY PRESERVATION AT SCALE USING ETHEREUM SMART CONTRACT

AUQIB HAMID LONE[*] AND ROOHIE NAAZ[†]

Abstract. Academic credentials are precious assets as they form an evidence for one's identity and eligibility. Fraud in issuance and verification of academic certificates have been a long-standing issue in academic community. Due to lack of anti-forgery mechanisms there has been substantial increase in fraudulent certificates. The need of the hour is to have a transparent and reliable model for issuing and verifying academic certificates to eliminate fraud in the process. Decentralized, Auditable and Tamper-proof properties of Blockchain makes it possibly the best choice for issuing and verifying academic certificates. In this paper we propose a model, where regulatory body authorizes higher education Institutes (universities and colleges) for issuing academic certificates to students in a decentralized way. Anyone in the world can verify the authenticity of the certificate by triggering appropriate smart contract functions, thus eliminating any possibility of fraud in the process. In addition we used multi signature scheme where certificates are required to be signed by designated authority from Higher Education Institutes, thus allowing for multi-level checks on certificate contents before being successfully deployed on Blockchain. We have also provide Proof of Concept in Ethereum Blockchain and evaluated its performance in terms of cost, security and scalability.

Key words: Certificate Integrity, Forgery Protection, Smart Contract, Ethereum Blockchain

AMS subject classifications. 68M14, 94A60

1. Introduction.

Higher Education has global impact and coverage and information regarding everyone's educational accomplishments should flow in a smooth and secured way. Academic credentials are used worldwide and are an important asset for both students and professionals pledging for jobs, scholarships or academic visibility. Traditional methods for recording, issuing and verifying academic credentials are expensive (due to intermediaries that charge fees for their services), inefficient (due to delays in executing agreements) and vulnerable (because it uses central system which can be compromised due to fraud, cyberattack, or a simple mistake which can effect the entire system). Need of the hour is to have secure, immutable and trustable academic credentials.

Blockchain in its simplicity is a series of connected tamper evident data structures called blocks, which contain or record everything that happens on some distributed systems on a peer-to-peer network . Each block is linked to and depends on previous block forming a chain, resulting in an append only system: a permanent and irreversible history that can be used as a real time audit trail by any participant to verify the accuracy of the records by simply reviewing data itself.

Blockchain was first developed for Bitcoin cryptocurrency and serves as distributed public ledger and transactions or events recorded on it are nearly impossible to tamper[16]. The driving force behind the interest in Blockchain research has been its key characteristics that provide security, anonymity and integrity without relying on trusted third party organisations. Initially Blockchain usage was restricted to cryptocurrencies only, since the advent of Ethereum: A next generation smart contract and decentralised application platform [8], applications beyond cryptocurrencies are being developed and explored.

Smart contract [20] is a piece of code which is stored in the Blockchain network (on each participant database). It defines the conditions on which all parties using contract agrees and certain actions described in the contract can be executed if the required conditions are met. As the smart contract is stored on every computer in the network, they all must execute it and get to the same result.

[*]Department of Computer Science and Engineering, NIT Srinagar, Jammu and Kashmir, India, 190006 (ahl@nitsri.net).
[†]Department of Computer Science and Engineering, NIT Srinagar, Jammu and Kashmir, India, 190006

The decentralised ledger functionality coupled with security provided by Asymmetric cryptography (Elliptic curve cryptography [15]) and distributed consensus algorithms (Proof of Work in case of Bitcoin and Ethereum [10]) of Blockchain, makes it a very attractive technology to solve the current financial as well as non-financial problems. Blockchain by design enforces integrity, transparency, authenticity, security and audit-ability thus making it possibly the best choice to make trust-less(distributed trust) systems to solve or improve traditional means for recording, issuing and verifying academic credentials. The goal of Blockchain is to provide anonymity, security and transparency to all its users.

1.1. Challenges and Limitations with Traditional methods for issuing and verifying academic certificates . One of the key challenges in traditional means of certificate verification is to deal with fake certificates. Traditional methods do not guarantee authenticity, security, tamper-resistance of academic records. Major limitations in traditional methods for issuing and verifying academic certificates are enumerated below:

- *Cost*: Traditional methods of certificate verification are costly as verification agencies charge fee for each certificate to be verified.
- *Time* : A great amount of time is lost in existing methods for verifying certificates as it depends on location and the response time of the issuing authority.
- *Availability*: Physical documents are susceptible to loss or damage. In case of loss or damage, individuals cannot readily avail duplicates of the certificates with existing process.
- *Third party dependency*: In traditional methods, organizations depend on third party verification agencies to verify the authenticity of the certificates with the issuing authorities.

Rest of the paper is organized as follows: Section 2 presents a brief overview of previous attempts for improving issuance and verification of academic certificates. Section 3 presents the description of proposed model with PoC in Ethereum Blockchain. Section 4 provides evaluation and analysis of proposed model. Finally, Section 5 concludes the paper and references are listed in the end. We also provide smart contract code for proposed model as an Appendix.

2. Related Work. Collecting literature to have a longitudinal and representative view of Blockchain applicability in certificate issuance and verification is challenging because of its rich diversity of Blockchain applications. In order to have clear-cut, unbiased, complete and broader perspective many sources have been explored including major online databases. One of the factors behind exploring major databases is their rich library of journals with high impact factors.

The University of Nicosia is the first university in the world to issue academic certificates whose authenticity can be verified through the Bitcoin blockchain. A Certificate granted to a student is issued as a PDF document. The Hash of this certificate is computed and registered in the Bitcoin blockchain as a transaction [1]. Another outstanding attempt in this direction is Blockcerts: an open standard for creating, issuing, viewing and verifying blockchain-based certificates. These digital records are registered on a blockchain, cryptographically signed, tamper-proof, and shareable. Blockcerts allows for batch issuance of certificates using Merkle trees to optimize storage in Blockchain [9]. Both [1] and [9] exploit Bitcoin's OP_RETURN feature to record certificate hashes. Blockcerts comes with one major shortcoming as it stores revocation list on centralized server, which could be exploited by the attackers to comprise the whole process. To overcome this author in [18] proposed Hypercerts, a decentralized mechanism for credential revocation, by combining the capabilities of Ethereum smart contracts and InterPlanetary File System (IPFS). The work presented in [5] proposed a Blockchain based scheme that guarantees integrity, proof of existence and also allowed to assess trust in students' data. This data is gathered in a so-called ePortoflio and includes completed assignments, course transcripts and granted certificates. It is stored in a peer-to-peer system called Interplanetary File System. Hashes of academic data are registered in Ethereum blockchain. Authors in [19] proposed a blockchain solution that allows users to assess the trust in academic data in terms of the reputation of the creators of such data. The assumption is that individuals trust a piece of data to be genuine if it has been issued by a reputable person or entity. To measure reputation, a currency called kudos is proposed.

Authors in [14] proposed a model of confidence in open and ubiquitous higher education, based on Blockchain technology. Proposed model is used to certify the acquisition of competencies by student trained in different educational institutions and is based on a consensus protocol of experts who are part of the system itself. Authors in [13] gave a different concept of using hybrid Blockchain comprising of 2 Blockchains. One private

Table 2.1
Summary of Related Work

Scheme	***Blockchain Technology***	***Features***					
		Multi-Sig	**Regulatory Authority Accredetion**	**Transparency**	**Privacy**	**Issuing Authority & Certificate Revocation**	**Accessibility**
[11]	Bitcoin	No	No	Yes	Yes	No	Yes
[10]	Bitcoin	No	No	Yes	Yes	Partial	Yes
[12]	Ethereum & IPFS	No	No	Yes	No	Partial	Yes
[17]	Ark	Yes	Yes	Partial	Yes	No	Partial
[18]	Ethereum	No	Yes	Yes	No	Yes	Yes

Blockchain for storing individual records and one public Blockchain for storing authentication information of private Blockchain in order to prevent tempering. Authors in [12] proposed Blockchain based platform for issuing, validating and sharing of Educational certificates. Authors in [21] proposed a blockchain-based higher education credit platform (EduCTX) with proof of concept in Ark Blockchain Platform, for creating a globally trusted higher education credit and grading system. Authors in [11] proposed a theoretical model for graduation Certificate verification using Blockchain Technology. The work presented by the authors in [17] proposed a Blockchain and smart contracts based scheme for higher education registry in Brazil. The authors in particular aims at digitization of degree certificates and academic credits for higher education in the Brazilian education system. Authors in [7] provided a detailed review on applicability of Blockchain in Education. In essence they discussed challenges and benefits of applying Blockchain in Education sector.

Proposed model provides numerous benefits more specifically to Higher Education Industry and in general to other industries also. Proposed model is generic in nature and can be applied to protect other digital documents also. Multi Signature scheme in proposed model allows for multi-level checks of certificate before being deployed to Blockchain. The potential benefits that proposed model could bring to Higher Education Industry are briefly summarized below:

- Certificate credentials published to Blockchain are immutable, trustful and verifiable.
- Simplifies the workflow of certificate issuance and verification, thus making whole process efficient and economical.
- Has the potential to transform education industry, by making processes involved more efficient, transparent, democratic and secure.
- Can be extended to be used for authentication and verification of other official statements or files also.

Proposed model is beneficial for Issuing Authority as it helps them to issue cryptographically-secure records that cannot be forged, records that are secure and auditable. Participants who have been issued degrees through proposed model will get benefited as they now own and can share cryptographically secure records, with instant access to academic accomplishments as Blockchain highly available in nature and instant verification as Blockchain eliminates dependency on issuing authority to verify records.

3. Proposed Model.

In this paper we propose a model for forgery protection of academic certificates through integrity preservation using Ethereum Blockchain in Higher Education. Simplified architecture of proposed model comprises of three parts as shown in figure 3.1. First part is about authorization request and response: wherein Higher Education Institutes request Higher Education Regulatory Authority for issuing certificates by presenting appropriate approval certificates and a list of delegated signature authorities. In response to Higher Education Institutes request, Regulatory Authority after proper verification, authorizes the institutes for issuing and revoking certificates to and from students. Part second of the proposed model is about certificate signing, issuance and revocation of certificates: wherein first delegated signature authorities of authorized Higher Education Institutes sign the certificates and then Institutes issue certificates to students. Under certain rare situations certificates are revoked from students by authorized institutes. Third and last part is about verification: wherein verifiers verify the authenticity of the certificates. Verification result is true if integrity test passes i.e. if details present on physical copy of certificate matches with the one present on the Blockchain else it returns false. Communication takes place via Blockchain network. For already issued certificates that are not deployed on Blockchain we have provided an export functionality in proposed model which helps Higher Education Institutes to put them on Blockchain.

Architecture of proposed model comprises of five main components namely Participants, Front-End, Core

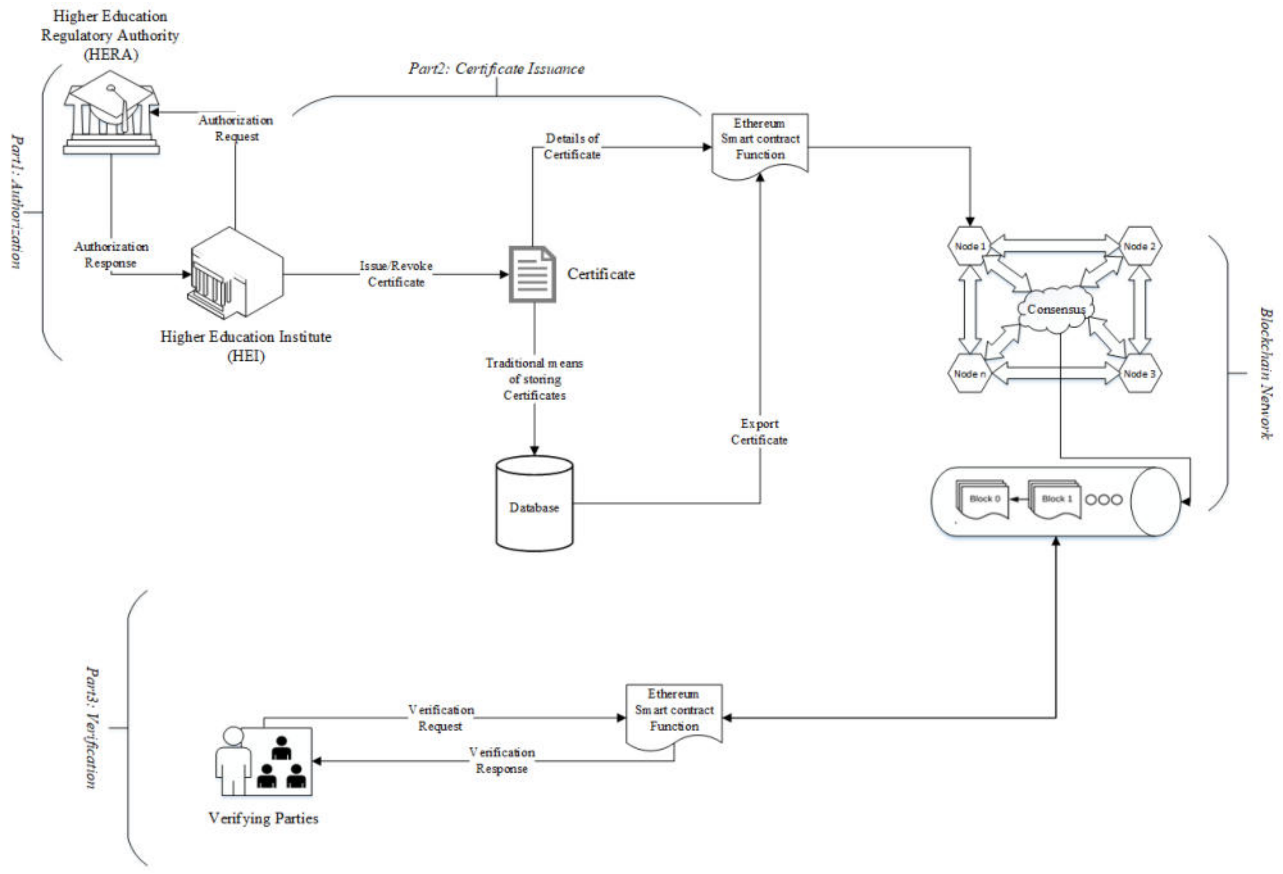


FIG. 3.1. *Simplified Architecture of Proposed Model*

Modules, Blockchain Network and Distributed Storage as shown in figure 3.2.

3.1. Participants. They are the real actors in any network. Participants mainly represent business but have the potential of representing people, regulators or other stakeholders. In proposed model participants include Higher Education Regulatory Authority, Higher Education Institutes and Verifiers. Regulatory authority and Institutes act as full nodes, storing entire copy of the Blockchain, while as Verifiers need not to store any Blockchain data. Verifiers verify claims made by any participant by fetching proof from Blockchain via front-end and appropriate core module of the proposed model. Every action performed by participants gets recorded on Blockchain via appropriate communication link. Thus in proposed model participants can be made liable for their actions on Blockchain network.

3.2. Front-End. Proposed model front-end is developed with the help of HTML5 and CSS. Appropriate JavaScript libraries (Web3.js [6]) have been used to connect front end with Blockchain network. It is beneficial to perform certain tasks at client side itself rather than computing them on Blockchain, because every computational step has a cost in Blockchain to be paid by participant who initiated the task. Certain validation checks can also be enforced at client side itself.

3.3. Core Modules. They are heart and soul of the proposed model and facilitate the communication of participants with Blockchain Network. Participants store and retrieve the certificate details to/from the Blockchain by calling an appropriate core module. Core modules are basically Ethereum smart contract functions. Proposed model Smart contract is written in Solidity language [4], compiled and tested both on Remix IDE [3] and Ganache(testrpc) [2] private network. We first defined the academic certificate and higher education

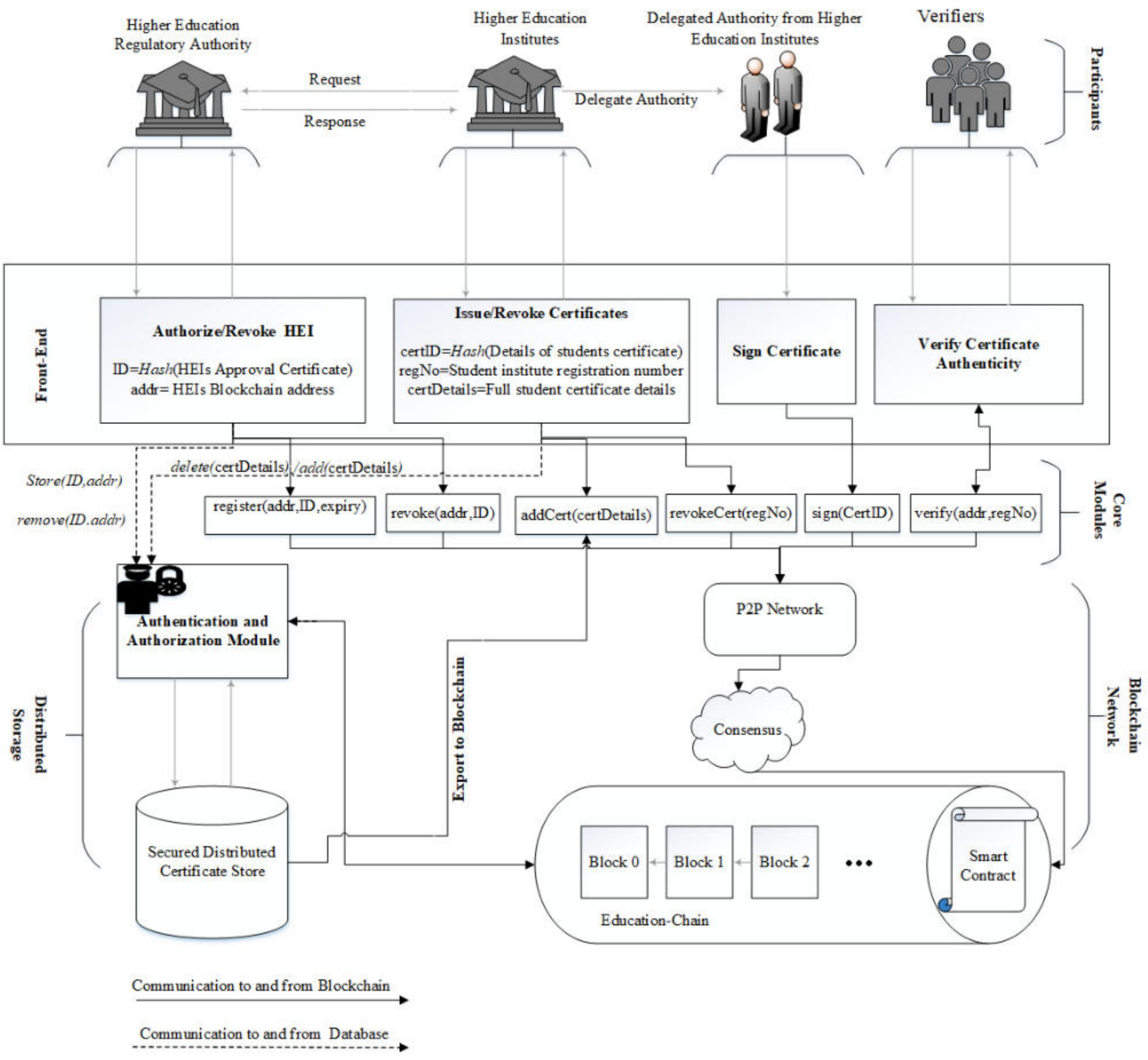

Higher Education Regulatory Authority
Higher Education Institutes
Delegated Authority from Higher Education Institutes
Verifiers
Participants
Request
Response
Delegate Authority
Front-End
Authorize/Revoke HEI
ID=Hash(HEIs Approval Certificate)
addr= HEIs Blockchain address
Issue/Revoke Certificates
certID=Hash(Details of students certificate)
regNo=Student institute registration number
certDetails=Full student certificate details
Sign Certificate
Verify Certificate Authenticity
Store(ID,addr)
remove(ID.addr)
delete(certDetails) /add(certDetails)
register(addr,ID,expiry)
revoke(addr,ID)
addCert(certDetails)
revokeCert(regNo)
sign(CertID)
verify(addr,regNo)
Core Modules
Authentication and Authorization Module
P2P Network
Consensus
Distributed Storage
Export to Blockchain
Blockchain Network
Secured Distributed Certificate Store
Block 0
Block 1
Block 2
Smart Contract
Education-Chain
Communication to and from Blockchain
Communication to and from Database

FIG. 3.2. *Operational Flow of Proposed Model*

institutes approval certificate as a solidity structures comprising of following information:

```
Struct approvalCert contains
    bytes32 hash;
    bool registered;
    uint expiry;
    bool revoked;
    address[] signAuth;
    // other optional stuff
```

- *hash*: Uniquely identifies the contents of Higher Education Institutes approval certificate issued by Higher Education Regulatory Authority . Hash is obtained by taking the SHA256 of contents of the approval certificate and other related information.
- *registered*: It comprises of boolean value, indicating whether Higher Education Institute is authorized by Higher Education Regulatory Authority.
- *expiry*: It is a unix timestamp indicating validity of approval certificate.
- *revoked*: It comprises of boolean value, indicating whether Higher Education Institutes approval certificate is expired. If revoked is true Higher Education Institute can no longer issue certificates.
- *signAuth*: An array of type address, containing list of Ethereum account addresses from Higher education Institute authorized for signing the certificate .

```
Struct studentCert contains
    bytes32 certHash;
    string regNumber;
    string instName;
    string stuName;
    string stuGrade;
    string stuDegree;
    address[] signedBy
    // other optional stuff
```

- *certHash*: Uniquely identifies the contents of students academic certificate issued by Higher Education Institute . Hash is obtained by taking the SHA256 of contents of the certificate and other related information.
- *regNumber*: regNumber is registration number assigned to the student by Higher Education Institute. It uniquely identifies the credentials of student.
- *instName*: Full name of the Higher Education Institute which issued the certificate.
- *stuName*: Full name of the student to whom certificate was issued.
- *stuGrade*: This represents academic credits that student obtained.
- *stuDegree*: Name of the course that student completes.
- *signedBy*: Array of Ethereum account addresses who signed the certificate.

In order to ensure the integrity of smart contract execution on participant actions, core modules (smart contract functions) are restricted based on the role. All functionalities of the smart contract were effectively screened for role limitation. Role restrictions are achieved with the help of solidity modifiers. Following modifiers are used in the proposed model for role restrictions.

1. *onlyOwner*: This modifier allows Higher Education Regulatory Authority which deploys smart contract to grant/revoke authorization to Higher Education Institutes. No other participant can authorize/revoke Higher Education Institutes, as owner is set in smart contract constructor itself. Hence, if any participant other than Higher Education Regulatory Authority tries to authorize any other participant, the transaction reverts as *onlyOwner* modifier returns false.
2. *onlyAuth*: This modifier allows only authorized Higher Education Institutes to issue certificates. Hence, if any participant other than authorized Higher Education Institute tries to issue certificates, the transaction reverts as *onlyAUth* modifier returns false.
3. *ifExists*: This modifier allows *addCert()* function to execute only if certificate hasn't been issued earlier.
4. *isSignedByAll*: This modifier allows *addCert()* function to execute only if certificate has been signed by delegated authorities.
5. *notSet*: This modifier allows *sign()* function to execute only if certificate hasn't been already signed by the delegated authority.
6. *onlyIncharge*: This modifier allows *sign()* function to execute only if certificate is being signed by authorized delegate authority, else transaction reverts.

Core modules in proposed model perform four basic functions 1) Higher Education Regulatory Authority authorizing or removing Higher Education Institutes for issuing academic certificates. 2) Higher Education Institutes issuing or revoking certificates to or from students. 3) Authorized signers from Higher Education Institute signing the certificate 4) Verification about the authenticity of certificates. Core modules (Ethereum smart contract functions) are triggered by the participants via front-end in the network. The constraints like who should access what function and under what conditions access should be granted to participants, are all enforced through solidity modifiers defined earlier. Pseudocode of the functions is presented below in the form of algorithms.

Authorize Higher Education Institute. No Higher Education Institute can issue or revoke certificates unless they are authorized by Higher Education Regulatory Authority (responsible for deploying smart contract). *register()* smart contract function takes Higher Education Institutes Ethereum account address, hash of Higher Education Institutes approval certificate (issued by Higher Education Regulatory Authority), expiry date of approval certificate and array of Ethereum account addresses (whose signatures are required for the execution of *addCert()* transaction) as input. On successful execution of the *register()* function, Higher Education Institute gets registered under Higher Education Regulatory Authority. Appropriate modifier is used so that *register()* smart contract function can only be executed Higher Education Regulatory Authority. *register()* smart contract function is briefly summarized in Algorithm 10.

Algorithm 1: *register()*

```
Input: address institution, bytes32 certHash, uint notValidAfter, address[] signee
Result: Registers the Higher Education Institute under Higher Education Regulatory Authority for certificate
        issuance and revocation
if onlyOwner then
    if Higher Education Institute aleardy authorized then
        revert
    else
        1. Set the hash of approvalCert struct corresponding to Higher Education Institutes
           address(institution) with certHash
        2. Set the expiry of approvalCert struct corresponding to Higher Education
           Institute address(institution) with notValidAfter
        3. Set the revoked of approvalCert corresponding to Higher Education Institutes
           address(institution) with false
        4. Set the registered of approvalCert struct corresponding to Higher Education
           Institutes address(institution) with true
        5. Push the signAuth of approvalCert struct corresponding to Higher Education
           Institutes address(institution) with signee
        6. Emit Registered event
else
    revert
```

Remove Higher Education Institute. revoke() smart contract function takes Higher Education Institutes Ethereum account address as input. On successful execution of *revoke()* function Higher Education Institute gets removed from Higher Education Regulatory Authority approved institutes list and no longer can issue or revoke certificate to or from students. Appropriate modifier is used so that *revoke()* smart contract function can only be executed Higher Education Regulatory Authority. The removed Higher Education Institutes address (institution) is pushed to revoked array, to prevent them from issuing certificates in future. *revoke()* smart contract function is briefly summarized in Algorithm 11.

Add Certificate. Add Certificate function takes certificate details as input and associates them with registration number of the student and Ethereum account of Higher Education Institute. Only authorized Higher Education Institutes can execute *addCert()* function. Furthermore all designated signers from Higher Education Institute must sign on the certificate for the *addCert()* function to get successfully executed. *addCert()* smart contract function is briefly summarized in Algorithm 12.

Algorithm 2: ***revoke()***

```
Input: address institution
Result: Removes Higher Education Institute under Higher Education Regulatory Authoritys authorized list
if onlyOwner then
    if Higher Education Institute aleardy authorized then
        1. Set the revoked of approvalCert corresponding to Higher Education Institutes address(institution)
          with true
        2. Push institution address to Higher Education Regulatory Authoritys revoked array
        3. Emit Revoked event
    else
        revert
else
    revert
```

Algorithm 3: ***addCert()***

```
Input: bytes32 hash, string regno, string instname,string name,string grade, string degree
if authorized && isSignedByAll then
    if certificate aleardy exists then
        revert
    else
        1. Set the hash of studentCert struct corresponding to students regno(registration number)
           and Institute account address with certHash
        2. Set the regNumber of studentCert struct corresponding to students regno(registration number)
           and Institute account address with regno
        3. Set the instName of studentCert corresponding to students regno(registration number)
           and Institute account address with instname
        4. Set the stuName of studentCert struct corresponding to students regno(registration number)
           and Institute account address with name
        5. Set the stuGrade of studentCert struct corresponding to students regno(registration number)
           and Institute account address with grade
        6. Set the stuDegree of studentCert struct corresponding to students regno(registration number)
           and Institute account address with degree
        7. Emit CertAdded event
else
    revert
```

Sign Certificate. sign() function takes student registration number, certificate hash and Institute address as input and on successful execution signs the certificate. Only authorized signature authority from concerned Higher Education Institute are allowed to sign the certificate and that too only once. Delegated signature authority approves certificate by signing on the hash of certificate contents corresponding institute address and student registration number. *sign()* smart contract function is briefly summarized in Algorithm 13.

Revoke Certificate. revokeCert() function takes student registration number as input and removes the certificate details from Blockchain by deleting corresponding entry from smart contract mapping data structure. The only check this function does is to ensure function is executed only by authorized Higher Education Institute and certificate already exists. Successful execution of this function results in negative Gas, given as an incentive for freeing some storage space on Blockchain. *revokeCert()* smart contract function is briefly summarized in Algorithm 14.

Verify Certificate. verify() smart contract function takes registration number and Hash printed on certificate as input and on successful verification returns the certificate view from Blockchain else verification fails. *verify()* function does not modify the state of the Blockchain, it only displays information retrieved from Blockchain. *verify()* smart contract function is briefly summarized in Algorithm 15.

Algorithm 4: ***sign()***

```
Input: Higher Eduaction Institute address, Hash of Certificate, Registration Number of Student
Output: Signs the certificate
if onlyAuthorizedSigner && notSignedEarlier then
    1. Sign the certificate
    2. Push the signer address to studentCert array signedBy
    3. Emit signed event
else
    revert
```

Algorithm 5: ***revokeCert()***

```
Input: string regno
if authorized then
    if certificate aleardy exists then
        1. Delete the certificate details of studentCert struct corresponding to students
           regno(registration number) and Institute account address
        2. Emit RevokedCert event
    else
        revert
else
    revert
```

3.4. Blockchain Network. It comprises of Peer-to-Peer (P2P) network and Consensus protocol that governs the communication over P2P network. In proposed model Blockchain network is private, where regulatory authority and authorized institutes can only act as validators or miners validating transactions, periodically collecting and creating blocks in the network. Higher Education Regulatory Authority is responsible for configuring, operating and maintaining Blockchain network. Regulatory authority also manages how other participants access and use the network.

3.5. Distributed Certificate Store. It comprises of distributed storage with authorization and authentication module for safely storing and preserving the original certificate details. This represents the traditional method for storing and preserving certificate records. But in proposed model it is optional for Authorized Higher Education Institutes to store newly issued certificates in distributed certificate store, however they can use export functionality of the proposed model to export already issued certificates from certificate store for deploying them to Blockchain.

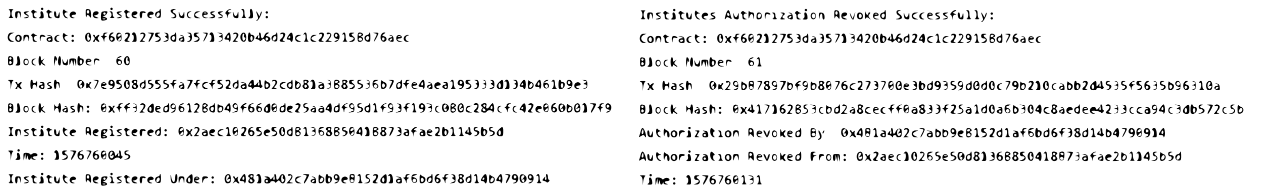
4. Experimental Evaluation and Analysis. This section presents the details of Experimental Evaluation and Analysis of proposed model. For experimentation we used Remix [3] IDE in-browser developing and testing environment connected to private Ethereum Blockchain. We also used appropriate JavaScript code snippet for catching events which get triggered on execution of various smart contract functions for analysis purpose.

We first performed testing to validate and verify three key scenarios in proposed model 1) Authorization/Revocation by Higher Education Regulatory Authority 2) Certificate signing and issuance/revocation by authorized Higher Education Institutes and 3) Verification by any participant for certificate authenticity by analysing outputs and logs of corresponding emitted smart contract events.

Authorization/Revocation by Higher Education Regulatory Authority. Higher Education Regulatory Authority uses *register()* smart contract function to register and authorize Higher Education Institutes for issuing certificates to students. Any participant calling *register* other than Higher Education Regulatory Authority which deployed smart contract leads to execution failure. Revocation of Higher Education Institute from authorized list is done by calling *revoke* smart contract function by Higher Education Regulatory Authority. The Results of successful authorization and revocation of Higher Education Institute by Higher Education

Algorithm 6: ***verify()***

```
Input: Student Registration Number, Higher Eduaction Institute address, Hash of Certificate
Output: Displays the appropriate Certificate instance from EDUChain
if Registration Number exists && input Hash == Hash from Blockchain then
    1. Verification Successfull
    2. Return the Certificate view from Blockchain
    3. Emit verified event
else
    return verification unsuccessfull
```



```
Institute Registered Successfully:
Contract: 0xf60212753da35713420b46d24c1c229158d76aec
Block Number  60
Tx Hash  0x7e9508d555fa7fcf52da44b2cdb81a3885536b7dfe4aea195333d134b461b9e3
Block Hash: 0xff32ded96128db49f66d0de25aa4df95d1f93f193c080c284cfc42e060b017f9
Institute Registered: 0x2aec10265e50d8136885041887 3afae2b1145b5d
Time: 1576760045
Institute Registered Under: 0x481a402c7abb9e8152d1af6bd6f38d14b4790914
```

```
Institutes Authorization Revoked Successfully:
Contract: 0xf60212753da35713420b46d24c1c229158d76aec
Block Number  61
Tx Hash  0x29b87897bf9b8076c273700e3bd9359d0d0c79b210cabb2d4535f5635b96310a
Block Hash: 0x4171628 53cbd2a8cecff0a833f25a1d0a6b304c8aedee4233cca94c3db572c5b
Authorization Revoked By  0x481a402c7abb9e8152d1af6bd6f38d14b4790914
Authorization Revoked From: 0x2aec10265e50d8136885041887 3afae2b1145b5d
Time: 1576760131
```

(a) Successful Authorization (b) Successful Revocation

FIG. 4.1. *Authorization and Revocation of Higher Education Institute by Higher Education Regulatory Authority*

Regulatory Authority is shown in Figure 4.1.

Certificate Issuance and Revocation by authorized Higher Education Institutes to students. Only after authorization from Higher Education Regulatory Authority, Higher Education Institutes can issue or revoke certificates to/from students via Blockchain. Higher Education Institutes issue certificates by calling *addCert()* smart contract function and revoke by calling *revokeCert()*. Designated signature authority sign Certificates using *sign()* smart contract function to approve them for being deployed to Blockchain. The successful signing of certificate hash by delegated signature authority are shown in figures 4.2.a and 4.2.b. Figures 4.2.c and 4.2.d shows successful certificate issuance and revocation by authorized Higher Education Institute respectively.

Verification by any participant for certificate authenticity. Participants can verify the authenticity and integrity of certificate by calling *verify()* smart contract function with appropriate parameters. Figure 4.3 shows decoded output of successful execution of *verify()* function.

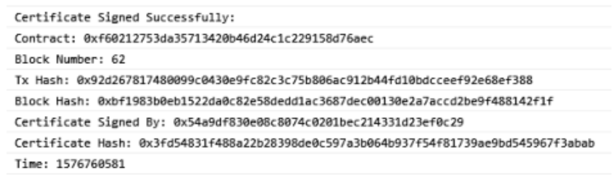
After successfully validating and verifying different functionalities of the proposed model we then analysed the feasibility of the proposed solution in terms of cost, security and scalability.

4.1. Cost Analysis. Every transaction executed on Ethereum Blockchain costs some Gas(unit of cost for a particular operation). In Ethereum Gas is paid in terms of Ether, as it is crypto fuel for running applications on the Blockchain network. There are transaction and execution gas costs for each function performed on the blockchain network. Execution cost includes the cost of internal storage in the smart contracts as well cost associated with any manipulation of Blockchain state. Transaction cost includes execution cost and the cost related to other factors like contract deployment and sending data to Blockchain network.

Table 4.1 shows the gas costs of smart contract functions of the proposed model. Smart contract functions are executed by the participants of the proposed model. *verify()* function costs least because it does not involve any updation in the Blockchain state, while as *addCert()* function costs the most because it considerably changes the state of the variables stored on Blockchain. *constructor()* function is a special function as it is related to the deployment of smart contract and is executed once in the life-cycle of the proposed model.

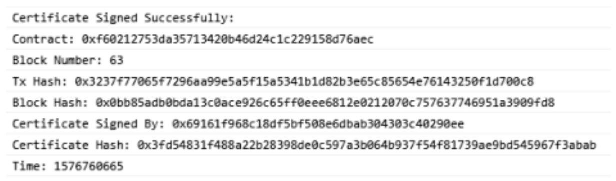
4.2. Security Analysis. In this section we present brief security analysis on how our proposed solution ensure key security goals such as integrity, non-repudiation, authorization, availability and accountability.

1. Integrity: Proposed model ensures integrity by storing traceability provenance data in immutable Blockchain infrastructure. Cryptographic hash functions make Blockchain immutable in nature.
2. Non-Repudiation: Every action is recorded in tamper-proof logs in proposed model and all actions



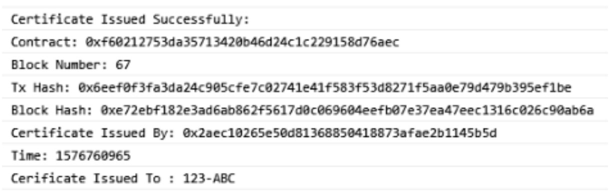
```
Certificate Signed Successfully:
Contract: 0xf60212753da35713420b46d24c1c229158d76aec
Block Number: 62
Tx Hash: 0x92d267817480099c0430e9fc82c3c75b806ac912b44fd10bdcceef92e68ef388
Block Hash: 0xbf1983b0eb1522da0c82e58dedd1ac3687dec00130e2a7accd2be9f488142f1f
Certificate Signed By: 0x54a9df830e08c8074c0201bec214331d23ef0c29
Certificate Hash: 0x3fd54831f488a22b28398de0c597a3b064b937f54f81739ae9bd545967f3abab
Time: 1576760581
```

(a) Successful Signature 1



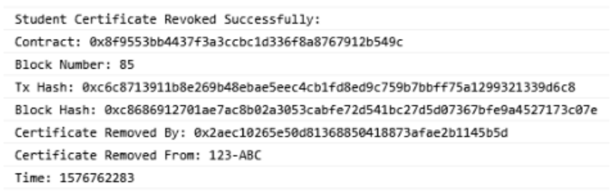
```
Certificate Signed Successfully:
Contract: 0xf60212753da35713420b46d24c1c229158d76aec
Block Number: 63
Tx Hash: 0x3237f77065f7296aa99e5a5f15a5341b1d82b3e65c85654e76143250f1d700c8
Block Hash: 0x0bb85adb0bda13c0ace926c65ff0eee6812e0212070c757637746951a3909fd8
Certificate Signed By: 0x69161f968c18df5bf508e6dbab304303c40290ee
Certificate Hash: 0x3fd54831f488a22b28398de0c597a3b064b937f54f81739ae9bd545967f3abab
Time: 1576760665
```

(b) Successful Signature 2



```
Certificate Issued Successfully:
Contract: 0xf60212753da35713420b46d24c1c229158d76aec
Block Number: 67
Tx Hash: 0x6eef0f3fa3da24c905cfe7c02741e41f583f53d8271f5aa0e79d479b395ef1be
Block Hash: 0xe72ebf182e3ad6ab862f5617d0c069604eefb07e37ea47eec1316c026c90ab6a
Certificate Issued By: 0x2aec10265e50d81368850418873afae2b1145b5d
Time: 1576760965
Cerificate Issued To : 123-ABC
```

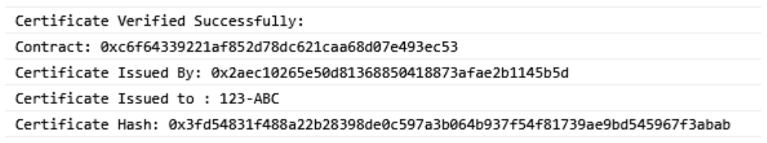
(c) Successfully Issued Certificate



```
Student Certificate Revoked Successfully:
Contract: 0x8f9553bb4437f3a3ccbc1d336f8a8767912b549c
Block Number: 85
Tx Hash: 0xc6c8713911b8e269b48ebae5eec4cb1fd8ed9c759b7bbff75a1299321339d6c8
Block Hash: 0xc8686912701ae7ac8b02a3053cabfe72d541bc27d5d07367bfe9a4527173c07e
Certificate Removed By: 0x2aec10265e50d81368850418873afae2b1145b5d
Certificate Removed From: 123-ABC
Time: 1576762283
```

(d) Successful Revocation

FIG. 4.2. *Signing and Issuance/Revocation of Certificates by Higher Education Institute*



```
Certificate Verified Successfully:
Contract: 0xc6f64339221af852d78dc621caa68d07e493ec53
Certificate Issued By: 0x2aec10265e50d81368850418873afae2b1145b5d
Certificate Issued to : 123-ABC
Certificate Hash: 0x3fd54831f488a22b28398de0c597a3b064b937f54f81739ae9bd545967f3abab
```

FIG. 4.3. *Successful Certificate Verification*

are linked and cryptographically signed by the initiator. No participant can deny their actions as everything is saved in the tamper-proof logs.

3. Authorization: In proposed model role restrictions have been enforced by using solidity modifiers to ensure proper authorization checks before executing any smart contract function.
4. Availability: Once certificate is deployed on Blockchain, it immediately becomes available to verifiers. The information stored on the Blockchain is saved in distributed and decentralized fashion and thus is immune to single point of failure.
5. Accountability: Since Ethereum account address of Higher Education Institutes is linked with approval certificate, thus Higher Education Institutes can be made accountable for their actions on Blockchain.

4.3. Scalability Analysis. Since Blockchain is append-only database, thus its size only increases with time. Size of Blockchain at any instant of time *t* is the sum total of the size of all blocks present in the blockchain at time *t*. Size of the single block is the sum total of the size of all transactions in it and size of the block header H_S which is a constant. Scalability analysis of the proposed model is based on the following assumptions:

- Chain configuration parameters are same as that of Ethereum mainnet
- Blocks are being generated at a constant rate of 15s i.e. $T = 15$ seconds
- Header size is same as that of Ethereum mainnet

Blockchain size heavily depends on the workload of the system, which in turn depends on many different factors like block gas limit G (maximum amount of gas that transactions in a block can consume) and block time period T (represents the rate at which new blocks are created in the Blockchain network.). if $I_t(txn)$ is set of transactions included in the Blockchain at time t, H_s is the header size of the single block, then size of

TABLE 4.1
Cost of Smart Contract Functions in proposed model

Function Caller	**Function Name**	**Gas Used**	**Transaction size in Bytes**
Higher Education Regulatory Authority	*constructor*	2963591	10948
Higher Education Regulatory Authority	*register()*	152015	447
Higher Education Regulatory Authority	*revoke()*	86412	255
Higher Education Institute	*addCert()*	182043	735
Higher Education Institute	*removeCert()*	32870	351
Signature Authority	*sign()*	87708	383

the Blockchain $B_s(t)$ at time t can be approximated by the following equation:

$$B_s(t) \quad = \quad \frac{t}{T} * H_s \quad + \sum_{txn \in I_t(txn)} S(txn) \tag{4.1}$$

where *S(txn)* is size of the transaction. Since block creation rate is constant, with new block being generated after every T second, thus equation (4.1) can be rewritten as

$$B_s(t) \quad = \quad N * H_s \quad + \sum_{txn \in I_t(txn)} S(txn) \tag{4.2}$$

where N is number of Blocks in the Blockchain at time t. The factor $N * H_s$ is the total overhead due to block headers in the Blockchain at time t, thus equation (4.2) can rewritten as:

$$B_s(t) \quad = \quad H_o \quad + \sum_{txn \in I_t(txn)} S(txn) \tag{4.3}$$

where H_o represents total overhead due to headers in the Blockchain at time t. The first term in equation (4.3) is dependent on block period T and second term is variable in nature depending on the time, block gas limit G and workload of the system. The growth of Blockchain size over a time interval [t_1:t$_2$] can be approximated by the equation below:

$$B_s(t_1 \to t_2) \quad = \quad H_o^{t_1 \to t_2} \quad + \sum_{txn \in I_{t_1 \to t_2}(txn)} S(txn) \tag{4.4}$$

where $H_o^{t_1 \to t_2}$ is the total overhead due to block headers recorded in the Blockchain from time period t_1 to t_2 and $I_{t_1 \to t_2}(txn)$ is set of transactions recorded in the Blockchain from time period t_1 to t_2. Since we were not able to find any publicly available statistics about number of approval certificates given by Higher education regulatory authority to Higher education institutes and number of degree certificates issued by approved Higher education institutes to students. We considered synthetic workloads with n number of institutes authorized for issuing certificates and each institute issuing $10n$ certificates per year. Thus equation (4.4) reduces to:

$$B_s(t_1 \to t_2) = H_o^{t_1 \to t_2} + n + 10n^2 + 10n^2 \tag{4.5}$$

2nd term in equation (4.5) represents total register() transactions, 3rd term represents total *issueCert()* transactions and 4th term represents total *sign()* transactions, as at least one signature from signature authority of authorized higher education institute is required for certificate to be broadcasted on Blockchain. We used equation (4.5) for computing the annual growth in Blockchain size with different classes of workloads. Results are presented in the Table 4.2. The second column of Table 4.2 contains the values for Blockchain growth rate per year with block period equal to 15 seconds, third column of Table 4.2 includes the growth rate without considering block headers overhead and fourth column contains overhead percentages. Results have shown even

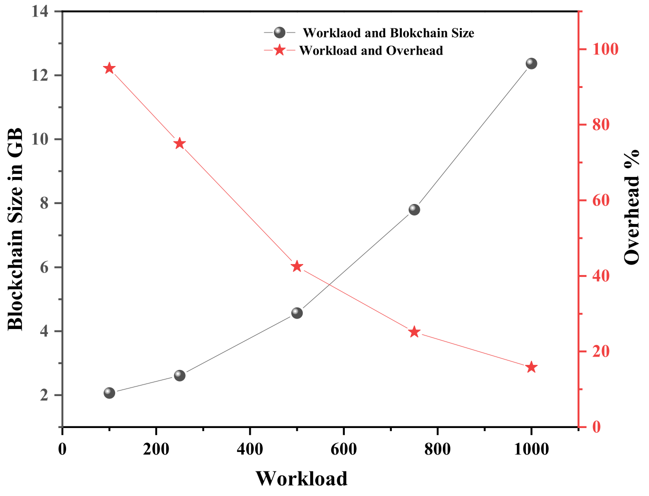


Fig. 4.4. *Blockchain Size/Year and Block Header Overhead % with different classes of Workload*

Table 4.2
Growth in Blockchain size with different values for n

Workload	***$B_s(t)$ with $T = 15s$***	***$B_s(t)$ - H_o***	***$H_o/B_s(t)$ %***
$n = 100$	2.062 GB/Year	0.104 GB/Year	94.9
$n = 250$	2.608 GB/Year	0.650 GB/Year	75
$n = 500$	4.561 GB/Year	2.603 GB/Year	42.9
$n = 750$	7.799 GB/Year	5.841 GB/Year	25.1
$n = 1000$	12.370 GB/Year	10.412 GB/Year	15.8

in case when 1000 institutes are authorized by regulatory body and institutes issues total of 10 million certificates per year, the growth rate is around 12.3 GB per year, which is acceptable given the storage capacities of modern-day high-end devices.

It is evident from the Figure 4.4, with increase in system workload Blockchain size increases, however overhead percentage decreases, thus proposed model scaling is limited by the workload of the system. In case of higher workload, Regulatory Authority can allow Institutes to issue certificates in batches, where a single transaction is done for issuing certificates to a batch of students thus reducing workload on the system. The concept of issuing certificates in batches was first implemented in Blockcerts [9].

5. Conclusions and Future Work. In this paper, we have proposed Blockchain based model for Academic certificate issuance and verification in Higher Education. Our proposed model architecture, algorithm, testing and implementation details are generic enough and can be applied to issue and verify other kind of certificates apart from academic ones. We provided the prototype of proposed model based on Ethereum Blockchain and evaluated its performance in terms of cost, security and scalability. Results confirm the feasibility of the proposed model. The future work will aim at developing complete optimized, privacy preserving end to end integrated framework for issuing and verifying academic certificates backed by distributed and decentralized storage with detailed case study on Indian Higher Education.

REFERENCES

[1] *Blockchain certificates (academic and others).*

[2] *Ganache a personal blockchain for ethereum development.* https://www.trufflesuite.com/docs/ganache/overview. Accessed: 19-12-2019.

[3] *Remix ide for ethereum smart contract programming.* https://remix.ethereum.org/. Accessed: 19-12-2019.

[4] *Solidity a high-level language for implementing smart contracts.* https://solidity.readthedocs.io/en/develop/. Accessed: 01-08-2019.

[5] *University to open. open blockchain.*

[6] *Web3 javascript api to interact with ethereum nodes.*

[7] A. Alammary, S. Alhazmi, M. Almasri, and S. Gillani, *Blockchain-based applications in education: A systematic review*, Applied Sciences, 9 (2019), p. 2400.

[8] V. Buterin, *Ethereum: A next-generation smart contract and decentralized application platform, 2013*, URL {http://ethereum. org/ethereum. html}, (2017).

[9] E. Durant, A. Trachy, and . O. of Undergraduate Education, *Digital diploma debuts at mit*, Oct 2017.

[10] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, *On the security and performance of proof of work blockchains*, in Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, ACM, 2016, pp. 3–16.

[11] O. Ghazali and O. S. Saleh, *A graduation certificate verification model via utilization of the blockchain technology*, Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10 (2018), pp. 29–34.

[12] W. Gräther, S. Kolvenbach, R. Ruland, J. Schütte, C. Torres, and F. Wendland, *Blockchain for education: lifelong learning passport*, in Proceedings of 1st ERCIM Blockchain Workshop 2018, European Society for Socially Embedded Technologies (EUSSET), 2018.

[13] K. Kuvshinov, I. Nikiforov, J. Mostovoy, D. Mukhutdinov, K. Andreev, and V. Podtelkin, *Disciplina: Blockchain for education*, 2018.

[14] D. Lizcano, J. A. Lara, B. White, and S. Aljawarneh, *Blockchain-based approach to create a model of trust in open and ubiquitous higher education*, Journal of Computing in Higher Education, (2019).

[15] H. Mayer, *Ecdsa security in bitcoin and ethereum: a research survey*, CoinFaabrik, June, 28 (2016), p. 126.

[16] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008.

[17] L. M. Palma, M. A. Vigil, F. L. Pereira, and J. E. Martina, *Blockchain and smart contracts for higher education registry in brazil*, International Journal of Network Management, 29 (2019), p. e2061.

[18] J. Santos, *A non-siloed blockchain-based certification service.*

[19] M. Sharples and J. Domingue, *The blockchain and kudos: A distributed system for educational record, reputation and reward*, in European Conference on Technology Enhanced Learning, Springer, 2016, pp. 490–496.

[20] N. Szabo, *The idea of smart contracts*, Nick Szabo's Papers and Concise Tutorials, 6 (1997).

[21] M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, *Eductx: A blockchain-based higher education credit platform*, IEEE access, 6 (2018), pp. 5112–5127.

Appendix A. Smart contract code of the proposed model.

```
pragma solidity >=0.4.22 <0.6.0;
contract Main{
address private owner;
address[] private revoked;
// Events
event CertAdded (address indexed_from,string reg_no, uint date);
event signed(address indexed_from,bytes32 certHash, uint date);
event verified(address indexed_from,bytes32 certHash, string regno,address[] signers);
event Revoked(address from, address to, uint date);
event RevokedCert(address by ,string  regno, uint date);
event Registered(address from, address to, uint date);
//Structures
struct BlockCert{
bytes32 hash;
string reg_no;
string inst_name;
string name;
string grade;
string degree;
bool isSet;
address[] signedBy;
}
struct AuthCert {
bytes32 apprCert;
uint expiry;
bool revoked;
address[] inch;
bool registered;
}
```

```
constructor() public{
owner=msg.sender;
}
// Mappings
mapping (address => AuthCert) private authorized;
mapping(address=>mapping(string=>BlockCert)) certs;
mapping(address=>mapping(bytes32=>bool)) signatures;
// Main Modules
function register(address institution, bytes32 cHash, address[] memory incharges, uint notAfter) public onlyOwner {
authorized[institution].apprCert = cHash;
authorized[institution].expiry = notAfter;
authorized[institution].revoked = false;
authorized[institution].registered = true;
authorized[institution].inch = incharges;
emit Registered(owner, institution, now);
}
function revoke(address institution) public onlyOwner {
require(authorized[institution].registered);
authorized[institution].revoked = true;
revoked.push(institution);
emit Revoked(owner, institution, now);
}
function check(address institution) public view returns(bool) {
require(authorized[institution].registered);
if (authorized[institution].revoked || authorized[institution].expiry < now)
return false;
return true;
}
function sign(address inst, bytes32 certHash, string memory regno) public notSet(msg.sender,certHash) onlyIncharge(inst,msg.sender)
{
signatures[msg.sender][certHash] = true;
certs[inst][regno].signedBy.push(msg.sender);
emit signed(msg.sender,certHash,now);
}
function isIncharge(address inst,address addr) private view returns (bool)
{
for (uint i=0; i< authorized[inst].inch.length; i++)
{
if(authorized[inst].inch[i] == addr) return true;
}
return false;
}
function addCert(bytes32 hash,string memory regno,string memory  instname,string memory  name,string memory grade,
string memory degree) public onlyAuth(msg.sender) ifExists(msg.sender,regno) isSignedByAll(msg.sender,hash) {
certs[msg.sender][regno].hash= hash;
certs[msg.sender][regno].reg_no=regno;
certs[msg.sender][regno].inst_name=instname;
certs[msg.sender][regno].name=name;
certs[msg.sender][regno].grade=grade;
certs[msg.sender][regno].degree=degree;
certs[msg.sender][regno].isSet= true;
emit CertAdded(msg.sender,regno,now);
}
function revokeCert(address inst, string memory regno) onlyAuth(msg.sender) ifExists(msg.sender,regno) public
{
delete certs[inst][regno];
emit RevokedCert(msg.sender, regno, now);
}
function verify(address inst,string memory  regno)   public
returns (bytes32  hash,string memory  instname,string memory  name,
string memory grade,string memory degree, address[] memory signers){
hash= certs[inst][regno].hash;
regno= certs[inst][regno].reg_no;
instname= certs[inst][regno].inst_name;
name=certs[inst][regno].name;
grade=certs[inst][regno].grade;
degree=certs[inst][regno].degree;
signers= certs[inst][regno].signedBy;
emit verified(inst,hash,regno,signers);
}
function isSigned(address inst, bytes32 hash) private view returns (bool){
uint count=0;
for ( uint i = 0; i < authorized[inst].inch.length; i++)
{
if(signatures[authorized[inst].inch[i]][hash]){
count++;
}
}
return (count==authorized[inst].inch.length) ;
}
// Modifiers
modifier onlyIncharge(address inst, address sender){
require (isIncharge(inst,sender));
_;
}
modifier onlyAuth(address auth)
{
require (check(auth)) ;
_;
}
modifier notSet(address signer,bytes32 hash)
{
require(signatures[signer][hash]== false);
_;
}
modifier isSignedByAll(address escow, bytes32 certHash)
{
require (isSigned(escow,certHash));
_;
}
modifier onlyOwner {
require(msg.sender == owner);
_;
}
modifier ifExists (address inst,string memory regno  ) {
require( certs[inst][regno].isSet== false);
_;
}
}
```

Edited by: Dana Petcu
Received: Sep 22, 2020
Accepted: Dec 13, 2020