applied sciences

MDPI

Article

UniChain: A Design of Blockchain-Based System for Electronic Academic Records Access and Permissions Management

Eman-Yasser Daraghmi [1,*], Yousef-Awwad Daraghmi [2] and Shyan-Ming Yuan [3,*]

1 Department of Applied Computing, College of Applied Science, Palestine Technical University Kadoori, Tulkarm P.O. Box 7, Palestine
2 Department of Computer Systems Engineering, College of Engineering and Technology, Palestine Technical University Kadoori, Tulkarm P.O. Box 7, Palestine; y.awwad@ptuk.edu.ps
3 Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan
* Correspondence: e.daraghmi@ptuk.edu.ps (E.-Y.D.); smyuan@cs.nctu.edu.tw (S.-M.Y.); Tel.: +970-595-765601 (E.-Y.D.); +886-3-5712121 (ext. 56631) (S.-M.Y.)

Received: 9 October 2019; Accepted: 12 November 2019; Published: 18 November 2019

Abstract: Although blockchain technology was first introduced through Bitcoin, extending its usage to non-financial applications, such as managing academic records, is a new mission for recent research to balance the needs for increasing data privacy and the regular interaction among students and universities. In this paper, a design for a blockchain-based system, namely UniChain, for managing Electronic Academic Records (EARs) is proposed. UniChain is designed to improve the current management systems as it provides interoperable, secure, and effective access to EARs by students, universities, and other third parties, while keeping the students' privacy. UniChain employs timed-based smart contracts for governing transactions and controlling access to EARs. It adopts advanced encryption techniques for providing further security. This work proposes a new incentive mechanism that leverages the degree of universities regarding their efforts on maintaining academic records and creating new blocks. Extensive experiments were conducted to evaluate the UniChain performance, and the results indicate the efficiency of the proposal in handling a large dataset at low latency.

Keywords: blockchain; smart contracts; proof of authority; incentive mechanism; permission management; Electronic Academic Records

1. Introduction

Over the last decade, the adoption of new technologies for the daily management of Electronic Academic Records (EARs) have begun worldwide. However, EARs are mostly physically localized and institutes maintain separate EARs, which are not connected to each other. This causes difficulties for individuals when they transfer from one institute to another, and when they search for jobs or scholarships. In addition, the disconnection among institutes makes the learning data that were collected at previous institutes unavailable for analysis at current or future institutes. Essentially, a student may visit more than one university during his study, such as registering in one university while taking courses in another. The EAR will be stored in the database of the university that issued the record, which will be the only eligible university for editing it. This university also will be responsible for the record's maintenance and management. Students with access rights could query their EARs from different universities. Universities with access rights could query EARs of a common student from other university when there is a need. These situations cause a lack of coordinated data management and exchange. In other words, academic records are fragmented and isolated, rather than cohesive.

The need for multiple access to the EARs has raised the interoperability challenges between students and universities, which pose additional barriers to effective data sharing. Additionally, as technology is constantly evolving, several advanced techniques are developed to violate digital privacy and security. Unfortunately, academic records are considered as major targets for information theft since they include private and sensitive information, e.g., the students' names, identity numbers, contact info and addresses. Therefore, the adoption of blockchain technology in managing EAR can be considered as one of the great breakthroughs of the last half a century. A blockchain is a distributed database solution which stores a continually increasing set of data verified and confirmed by participants. Researchers believe that the blockchain technology can shape the academic industry in everything from protecting academic records and offering better student packages to streamlining billing.

Although the blockchain technology was first introduced through Bitcoin, extending its usage to non-financial applications is a mission for researchers. Managing EARs is one of the fields where the blockchain technology is believed to have considerable impact. The adoption of the blockchain in the managing EARs can be found in [1,2] Research in this field is relatively new but increasing very quickly. Since a few studies propose to employ the blockchain for managing EARs, there is still a need for more research to better understand, characterize and evaluate its utility in EARs management systems. Therefore, this work introduces a fully functional framework for applying blockchain technology to EARs.

In this work, the architecture of a blockchain-based framework applied to EARs is proposed. The proposed framework aims at providing interoperable, secure, and efficient access to EARs by universities, students and third parties while maintaining the students' privacy. In addition, this work proposes timed-based smart contracts whose design meets the demands of EARs. These contracts are employed in the blockchain for governing the transactions, monitoring the computations performed on the EARs through the enforcement of the acceptable usage policies and managing the use of data after transmission. Advanced cryptographic techniques are also adopted by the proposed framework for providing further security.

In addition, since an EAR is a student's asset and not a cryptocurrency or a digital currency to be exchanged, unlike previously proposed blockchain-based systems for EARs, a new incentive mechanism is proposed. This mechanism leverages the degree of university nodes from the perspective of EARs systems by measuring their efforts regarding maintaining academic records and creating new blocks. University nodes with fewer degrees are more likely to be selected for creating the new block. The proposed approach rewards the "block's creator" an incentive that is added to its degree to decrease its probability of recreating the next block instead of just creating a digital currency, thus achieving a fairness among universities and ensuring the sustainability of the system. Moreover, the performance of the proposed system is measured (with respect to average response time, throughput and communication overhead) by conducting analyses on the EARs' queries. The results show that the proposed system efficiently handles a large dataset at low latency.

In summary, this research presents the design of a blockchain-based system for EARs that handles the issues of privacy, security, data fragmentation, data isolation, effective access to academic records, and system interoperability. The primary contributions of this work are fourfold:

1. A complete analysis regarding how the proposed UniChain system and the timed-based smart contracts can interact with the various demands of universities, students and third parties is presented.
2. How the proposal would address the longstanding issues of privacy and security is detailed.
3. An incentive mechanism that aims at evaluating the degree of universities regarding their work in maintaining EARs, which in turn enhances data quality for EARs, is proposed.
4. Extensive experiments to evaluate the performance of proposed frameworks on various aspects, including throughput, response time, and communication overhead, were performed.

2. Literature Review

2.1. Background

2.1.1. Blockchain

Blockchain technology was first introduced as the core technology behind the Bitcoin cryptocurrency proposed by Satoshi Nakamato in a pseudonymous paper in 2008 [3]. A blockchain is a distributed and decentralized peer-to-peer network where all transactions performed by participants are stored in a single immutable public ledger. Public–private key encryption is used to identify participants. The public key is used for public identification, while the private key is used for authorizing transactions created by the owners or for claiming an asset that has been encrypted with their public key.

Essentially, a block is a data structure including: a block header that includes a hash value of the previous block, timestamp as well as a Merkle root, and a data part that has relevant transactions' data. Typically, a transaction contains the public key of the sender, data, and the hash value of the preceding transaction. The data part enables the blockchain to store different electronic assets such as records, certificates, transcripts, property rights and licenses. All of the blocks are linked by the order of the hash value. In the blockchain, the block's chain is duplicated across the distributed blockchain network and stored by minor nodes.

2.1.2. Smart Contract

A smart contract is a way to digitally formalize and secure relationships over a network [1]. The main idea of smart contracts is to enable embedding different kinds of contractual clauses, collateral, bonding, and property rights in computer software or hardware so that a malicious breach of the contracts is dramatically reduced [4]. In this context, a smart contract is defined as an application that runs on the blockchain network and is executed by all network participants [5]. Smart contracts are computer codes that govern the blockchain transactions and define the conditions of mutually agreed contracts [6].

Recently, many blockchain-based projects have implemented smart contracts, such as the Ethereum platform and Hyperledger. They allow trusted agreements and transactions to be rendered among distinct, anonymous entities with no need for a central authority or external enforcement mechanism. The Ethereum platform allow the developing of smart contracts that suit the requirements of the desired system. In the context of adopting smart contracts in EARs systems, they allow the creation of scalable and dynamic conditions, terms and rules to securely exchange and sharing academic records.

2.2. Related Work

Academic institutions have invested much in developing electronic systems for better management and access of academic records. However, these systems suffer from the lack of protection of private information, and the difficulties in achieving lifelong learning logging by transferring academic records across multiples institutions because of the disconnection in learning records [7,8]. Solutions to such problems are based on blockchain, which can benefit the education systems by improving the security and the protection of creating and maintaining certificates, credentials and education records [7,9,10]. Simultaneously, it would allow several parties to share and access data with high consistency, immutability, and transparency [11,12].

To the best of our knowledge, few peer-reviewed papers have presented complete blockchain-based systems in education. In [7], an Ethereum-based system is proposed to store students grades and reward students with cryptocurrency. In [2], a Hyperledger Fabric-based system called Gradubique is proposed to allow instructors post exams and grades to the Gradubique network. According to Ocheja et.al., blockchain-based systems should not only record transcripts and certificates but also other student achievements [8]. Therefore, the authors of [8] proposed a blockchain-based system

that stores student learning activities and allows the transfer of learning records in a secure and verifiable way. In addition, an educational record repository based on blockchain is proposed in [1] for securely distributing educational certificates among academic institutions and third-party professionals. Focusing more on transferring credits among institutions, the authors of [13] proposed a blockchain credit transfer system based on the European Credit Transfer and Accumulation System (ECTS).

There are also blockchain-based systems developed by different universities such as the system at Massachusetts Institute of Technology (MIT), which is an open-source system that allows individuals to securely create and share official documents and academic transcripts [14]. Based on the MIT system, Blockcerts was developed to be an open standard for creating, issuing, viewing, and verifying blockchain-based certificates [15]. Further, Virginia colleges has started to develop blockchain-based diplomas to distribute student degrees through the decentralized computer networks that power Bitcoin [16]. To go beyond transcripts and certificates, the United Arab Emirates University (UAEU) developed a large-scale project called Passport that utilizes the benefits of blockchain for improving the educational and the organizational efficiency [17].

Despite the entire benefits of blockchain to education systems, the aforementioned studies and attempts are still under development and their scalability has not been tested. In addition, there are some challenges facing the application of blockchain in education. The efficiency of blockchain is the main challenges facing blockchain in education, as stated in [7,18]. In addition, academic institutions do not easily accept the trustless principle and openness to the globe [7]. It is further stated that blockchain technologies, platforms, and smart contracts are not easy to understand by learners, educators, and other professional parties in the chain [8,18].

3. UniChain Architecture

3.1. Overview

This section details the architecture that would be built on the top of existing university databases. To minimize the requirements of storing the EARs in the blockchain and to utilize the existing systems, academic records would be continuously stored in the university databases. In the proposed design, university nodes are responsible for the blockchain maintenance since in reality universities maintain and manage their students' records, while students can only read data. All accesses to the EARs are performed through the blockchain, and, accordingly, the history of those accesses will be stored in the blockchain to provide a full view of all events occurred to students' records, hence ensuring the integrity of data and preventing misuse of student records. All log details in addition to the record ownership metadata are added to the chain. To ensure the integrity of data, the proposal employs the hashing method, i.e., SHA-256. In UniChain, a hash value of the access link that is generated during the record's issue (i.e., this link is used to access the stored EAR) is kept in the blockchain instead of keeping the link itself. To access an academic record, the encrypted access link is sent over HTTPS to the associated participant who has access rights. Thus, the stored hash value of the link ensures that no alterations have been made outside the blockchain during the transfer as the value of the hash is unique to the original record. For providing further security, the access link, the key which is used to encrypt the EAR, and the EAR itself are stored in different locations. UniChain maintains the privacy by employing timed-based smart contracts for governing transactions. Security and access control are preserved by the adoption of advanced encryption and authentication techniques throughout the blockchain. Interoperability, auditability, and accessibility are ensured by the use of comprehensive logs. UniChain employs a new incentive mechanism integrated with the Proof of Authority (PoA) consensus algorithm for creating, validating, and appending new block.

3.2. Preliminaries

3.2.1. Incentive Mechanism

This work proposes a new incentive mechanism integrated with the PoA consensus algorithm to leverage the degree of universities from the perspective of EARs management systems. The proposed incentive mechanism measures the universities' efforts regarding maintaining EARs and creating new blocks. An academic record is a student asset and not a digital currency or a cryptocurrency to be exchanged. Thus, the degree of a university node indicates the significance a node owns regarding its quantity and quality of academic records. The proposal defines quality in academic records as having the attributes of legibility, completeness, consistency, correctness, and non-redundancy. The total quality of all EARs for all students stored in the university database evaluate the degree of a node.

The new incentive mechanism indicates that universities with fewer degrees are more likely to be selected for creating the new block. The university node that has the least degree is classified as "a block's creator" node, while the nodes with degrees greater than the average degrees of the network are considered as "voters". Voter nodes are responsible for the validation process when adding new nodes to the system. They validate whether the ID is suitable for the requested role and guarantee that the node is a legitimate academic university or third party, thereby decreasing the possibility that illegitimate nodes can join the system. The "block's creator" node is rewarded an incentive that is added to its degree for potentially reducing its probability of recreating the next block instead of just creating a digital currency, thus achieving a fairness among universities and ensuring the sustainability of the system.

This work aims at improving the data quality of EARs since universities with the following attributes have their degrees increased: (1) fill in more legal, correct, consist, complete and non-redundant items of an existing academic record; (2) create new legal, complete, consistent, correct and non-redundant records; and (3) generate a new block. Accordingly, they have less probability to perform the computational task of creating the new block.

3.2.2. Proof of Authority (PoA)

Several types of proofs, such as the Proof of Work (PoW) and the Proof of Stake (PoS), are employed in different blockchain-based systems to determine the miner block to be appended next. The PoA [19] is a consensus algorithm proposed by Gavin Wood, Ethereum co-founder and former CTO, in 2017, as a replacement for the PoW. It can be used for setting a private blockchain by considering the value of participants' identities to create a set of "authorities" that are allowed to create new blocks and secure the blockchain network. In other words, block validator "authorities" that are arbitrarily chosen as trustworthy entities are not staking coins but their own reputation instead to maintain security. According to the PoA, verifying the blocks and transactions by authorities who act as moderators of the system achieves several benefits: maintaining the privacy of the system while acquiring the benefits of the blockchain technology; improving the security of the system; minimizing the computational intensiveness; and increasing the system performance as it provides lower transaction acceptance latency and steady time intervals for issuing blocks.

3.3. Software Components

The software components of the UniChain system are proposed in this section (see Figure 1).

Generally, UniChain is an Ethereum-based management system to enable permissions management and accessing EARs. UniChain is decentralized solution that utilizes extensible smart contracts to encode access rights. The UniChain system consists of two types of nodes: student nodes and university nodes. A student node consists of five main components: EARs Interface, Backend Library, Ethereum client, Cipher/Decipher Manager and Data Base (DB) Manager. In addition to these mentioned components, a university node has a Records Evaluation Manager (REM) component that is responsible for calculating the degree of nodes. The next sections describe these components in detail. As shown

in Figure 1, a user requests a service (add, update, read, etc.) from the UniChain system via a web page (1). The request will be transferred to the Ethereum client via the Backend library (2 and 3). The backend library first performs low level formatting and parsing to communicate with the Ethereum client (2). The Ethereum client in turns communicates with the smart contracts to query the "access rights" regarding the received request (4). Upon receiving the "approval" from the blockchain network, the Ethereum client will transfer it to the Cipher/Decipher Manager through the backend library (5–7). The Cipher/Decipher Manager performs the encryption/deception schema and transfers the query to the DB Manager that is responsible for accessing the records DB (8 and 9). More details regarding the interaction between the nodes and the smart contracts in different scenarios are presented in Section 4. Furthermore, a university node evaluates the EARs stored in its database upon receiving a notification by the use of the REM component. This component communicates with the database through the DB Manager (a–c).

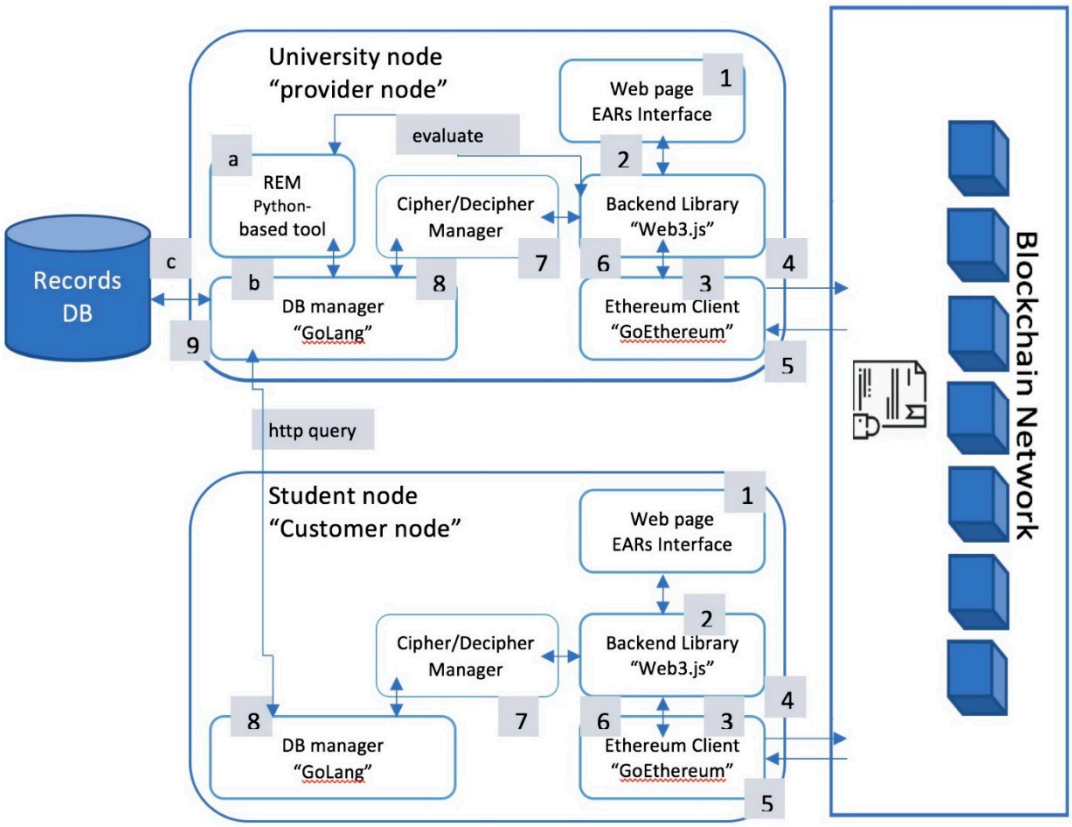

Figure 1. Software Components of the UniChain System.

3.3.1. Records Evaluation Manager (REM)

REM is a python-based tool that will be installed and configured only on the university nodes to compute and evaluate their degrees in the blockchain during the initialization stage based on the quantity and the quality of EARs stored in each university database. REM extracts features, manages relevant data, and classifies the unstructured parts of a record in order to prepare an EAR for quality evaluation. A classification schema that integrates lexical, semantic and syntactical analysis of the record will be employed in this work [20]. After classification, the degree of a node can be computed as given in Equation (6). The degree of each node will be stored in the Nodes Contract (NC) to be used for determining voter nodes and selecting the node to generate the next block. Figures 2–4 show the structure of anonymous EARs.

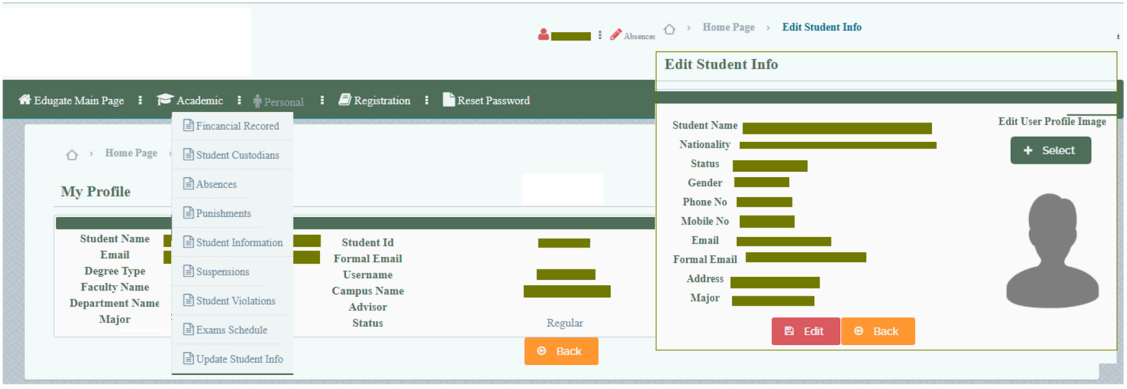

Figure 2. The structure of EAR—Example 1.

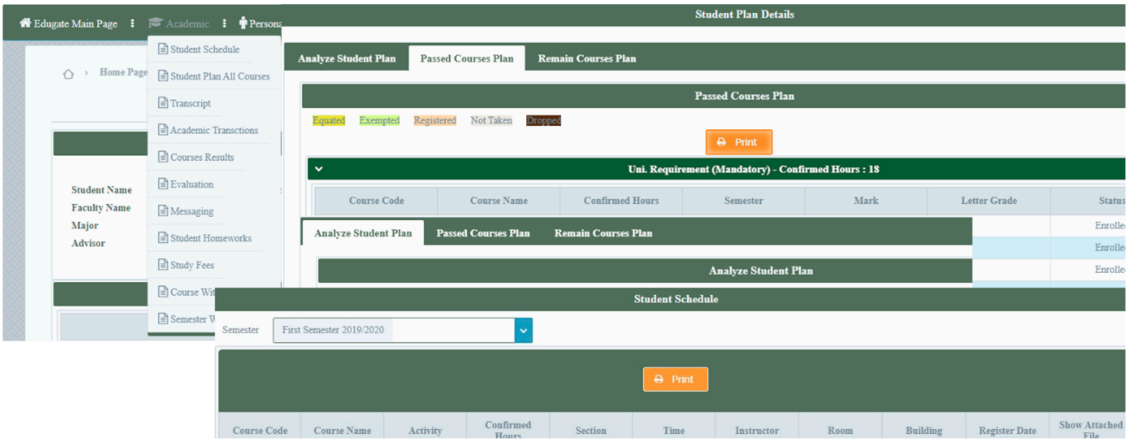

Figure 3. The structure of EAR—Example 2.

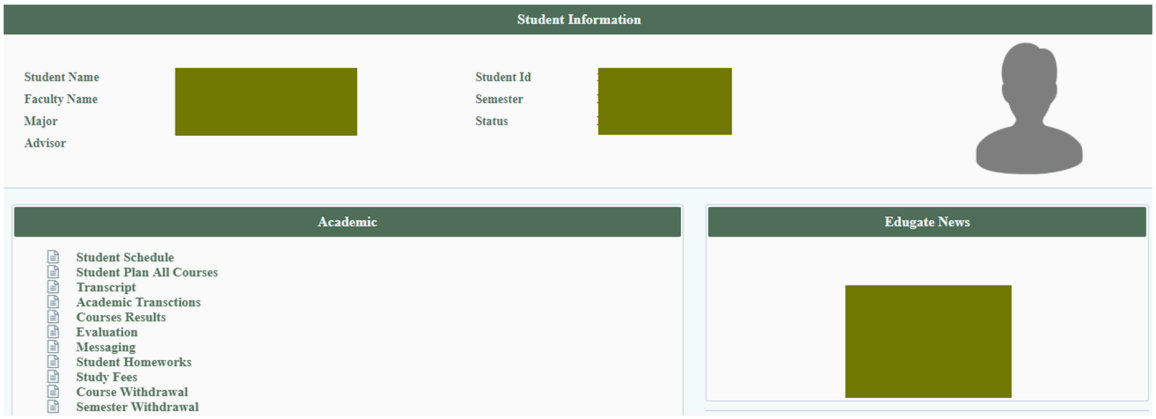

Figure 4. The structure of EAR—Example 3.

3.3.2. DB Manager

The proposed system is integrated with the existing EARs stored in the university databases by creating access links to that records. The DB Manager, which is an API written in GoLang, provides access to the existing databases and is controlled by the permissions information stored in the blockchain. The DB Manager functions to navigate the existing database, and create access links for EARs. To ensure data integrity, the DB Manager creates a hash value for the created access link as well as the student's EAR to be stored in the Participants' Record Contract (RC) in the blockchain. It also creates a hash value for the log to be stored in the Logs Contract (LC).

3.3.3. Cipher/Decipher Manager

In the UniChian system, the Cipher/Decipher Manager functions as the encryption and decryption schemas. Generally, there are two encryption schemas used in the proposal. First, the symmetric key encryption schema is employed to encrypt the EARs. The Cipher/Decipher Manager first creates a symmetric key to encrypt the academic record, and then re-encrypts that key with the public keys of the: university node and student node. Second, the public key encryption schema is adopted to securely distribute information among parties over HTTPS and to encrypt the record that will temporarily be kept in the Deposit-Box Contract (DBC) to facilitate the access by a third party.

3.3.4. Ethereum Client

The Ethereum client is the access point to the Ethereum blockchain network as it includes all the functionalities required to join that network [21]. The proposed design works on a permissioned blockchain network; therefore, nodes with permissions will use the client to access the private blockchain. For the implementation of the proposed prototype, the GoEthereum client is used. It can be accessed by the use of Java Scrip Object Notation – Remote Procedure Protocol (JSON-RPC) endpoints on the Internet [22]. With GoEthereum, users can access their nodes' information over HTTPS using a wallet that may have different functionalities based on the type of node.

3.3.5. EARs Interface

EARs Interface is the web-based interface that is used for managing EARs by universities, viewing the EARs by students and managing the retrieval options as well as the data sharing.

3.3.6. The Backend-Library

This component abstracts the communication with the blockchain, exports a function call API, and interacts with the Ethereum client in order to perform low level formatting and parsing.

3.4. Smart Contracts

A blockchain-based system should have its own smart contracts to govern and monitor transactions. The proposed smart contracts employ a set of connectivity and timing functions to provide a reasonable period of time for performing transactions and thus ensuring an authorized transaction is intended. All contracts have "T" date field that is used for implementing the timing functions. In the proposed system, as shown in Figure 5, the smart contracts have the following.

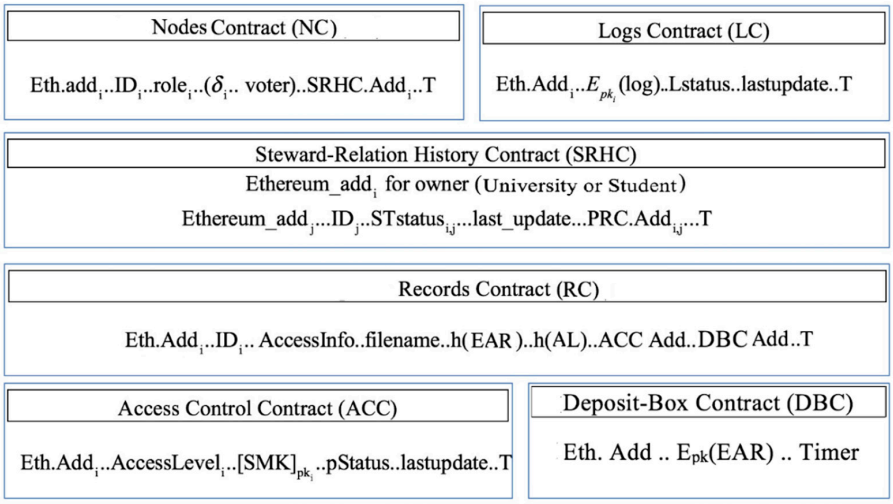

Figure 5. The proposed smart contracts.

3.4.1. Nodes Contract (NC)

NC is a global contract that preserves the registration, the mining, and certain overwrite procedures for the blockchain. The NC determines voter nodes in the system along with the "block's creator" node, which will create the next block. It maps a node identification string to its associated Ethereum address identity, i.e., public key. The adoption of the identification strings rather than the public key directly is to allow already existing IDs to be used. Procedures and policies coded into the NC regulate adding and registering new IDs. This contract additionally determines the role of each participant node within the system i.e., university node, student node, third party, etc. in order to recognize nodes that have already registered and thus avoiding the case of double registration. For those nodes with a "university node" role, additional data fields will be appended to maintain the degree of the node along with a Boolean data field that determines whether a node is considered as a voter node or not. Additionally, the NC maps the Ethereum address of the node with its associated SRHC address.

- For determining the voter nodes, the NC stores the degree of each university node within the system that in turn will be used for calculating the average degree of the network and selecting voter nodes.
- For selecting the "block's creator", the NC assigns this task to the university node that has the least degree among all university nodes in the system.
- For mining, the NC functions using the PoA consensus algorithm [23] integrated with the proposed incentive mechanism.
- For the registration process, voter nodes in the NC are responsible for validating other nodes that demand a role with "higher levels" upon being added to the system. Voter nodes guarantee that no threat will be created to the system. Generally, during the blockchain initialization stage, the NC will be empty. An administrative node (i.e., temporary node) will be added as an initial university node and will be removed once enough nodes have joined the system.
- For the overwriting procedures (i.e., for nodes that may leave the system, such as a university node goes out of business or those nodes that may harm the system), the NC can be used to perform the overwriting procedures to revoke permissions associated with that nodes. The overwriting procedure is accomplished by submitting a request to remove a node from the system via a voter node and reaching the majority of votes by the rest of voters. This would then require overwriting the type of a node as terminated, removing it from the NC, and deleting its related information from the various contracts.

3.4.2. Steward-Relation History Contract (SRHC)

This contract maintains the steward relationship history of each participant node in the system where the student's academic record is stored and managed by the university node. The SRHC locates the history of academic records by holding a summary list of the steward relationship. For example, if a node's role is student, its SRHC will have references to all universities that it has been engaged with. On the other hand, the SRHC of a university has references to all student nodes which that university serves. Every node within the blockchain system will have a SRHC that will be created during the registration process.

Generally, the SRHC is identified by the Ethereum address of the SRHC owner node and stores the Ethereum addresses of all associated nodes, their related IDs, a stewardship statue, a last-update date field that indicates the last update on the status field, and an address to the applicable RC. User notifications can be enabled via the use of the stewardship status field, such as the stewardship is "newly" established, "awaiting pending updates", and "acknowledged student approval" or "acknowledged student denial". The stewardship status in the student SRHC is set by university node in the system every time they update the student record or as a part of establishing a new stewardship. Thus, students can be notified upon modifying the stewardship status field as a new stewardship is recommended or an update is available.

3.4.3. Records Contract (RC)

This contract tracks all records in which universities store for students and is generated when a new steward relation is established between two nodes. The RC includes several data fields with different purposes, and is identified by the Ethereum address of the owner that signifies the student that owns the record(s). Each record has a filename f, conditions, and AccessInfo. The filename indicates the identity string for the student record. The AccessInfo data field of the record specifies the needed information to find the EAR database of a university, i.e., the university's host name and the information for the port in a standard network topology. Moreover, to maintain data integrity, each record has a hash value h(AL) for the access link of the file, and a hash value h(EAR) of the stored record. Moreover, references to the AAC address and the DBC address are listed in the RC.

3.4.4. Logs Contract (LC)

This contract tracks all transactions performed on the EARs to facilitate adding/validating/appending blocks in the blockchain network. This contract is identified by the Ethereum address of the source of the transaction. It lists the transaction details in an encrypted log data field with a status field that indicates whether the new log has been added to the blockchain. It also stores the last update of the status field.

3.4.5. Access Control Contract (ACC)

The ACC includes all permissions related information which is specific to every record. It lists the Ethereum addresses for all nodes that have access permissions on the record. This contract specifies the level of that access (i.e., owner, read/edit, and blind-read), and a symmetric key encrypted with the public key of each node. A "read/edit" access level indicates that a node can read and partially edit the EAR as it has the symmetric key that is generated to encrypt the record when it is first added (i.e., edit access level means a student for example can update his address or change his profile picture). The temp-read level indicates that the node can temporarily read a record as it has the DBC address that will keep an encrypted copy of the record for a certain time to allow the temporary access by a trusted third party. The owner level is assigned to the university node that adds the record. It indicates that a node has full access and control of the ACC as it can add other nodes with the "read/edit" level, remove nodes from the AAC, and also alter the level for any existing nodes. The AAC also contains the "pstatus" field along with the lastupdate in order to notify participants when there is a change in their access level.

3.4.6. Deposit-Box Contract (DBC)

This contract is employed when there is a verified request from a third party to access an EAR for a student. This contract stores an encrypted EAR for a certain time to facilitate the record's access by a third party. The stored record will be encrypted via the public key of the third part before storing and will be kept only for a certain time specified by the Timer field.

4. Implementation of the Proposed UniChain System

4.1. Blockchain Initialization—Part I: Adding a New University Node to the Blockchain Network

In this stage, all universities that accept to join the blockchain network have to share their EARs and agree on: the rules of the proposed smart contracts, the proposed incentive mechanism, the frequency of updating the blockchain network and the process of generating, verifying, and appending a new block to the blockchain network. In practice, each university has an identification string or a public identifier (ID) that must be unique to the academic organizations. In addition, it should be assumed that the Ethereum addresses (which are equivalent to public keys) of all universities that agree to join the blockchain network have been received, and the software components have been installed.

The process of adding a new university node starts when the ID and the Ethereum address of the new node are sent to the NC in addition to the requested type role. Voter nodes in the NC validate and authenticate that received request by ensuring and confirming that the received request is related to a legitimate university that is not registered previously. If the request is accepted and validated, the NC updates its local memory with the Ethereum address of the new node, its ID and its role. The NC creates a new SRHC for the new node whose address will be sent to that university node.

4.2. Blockchain Initialization—Part II: Computing the Degree of a University Node

The process of computing the degree of a node is performed by a node that has a "university" role listed in the NC. In this stage, the REM installed in each university node computes the degree of the associated node within the network. Since an EAR is a student's asset and not a cryptocurrency or a digital currency to be exchanged, unlike previously proposed blockchain-based systems for EARs, a new incentive mechanism is proposed. The mechanism leverages the degree of university nodes from the perspective of EARs systems by measuring their efforts regarding maintaining academic records and creating new blocks. The degree of a node is calculated based on the quantity and the quality of the EARs stored in its database. Based on the purposes and the perspective of the designed system, various methods and several attributes could be used to define the quality of academic records. Generally, the quality of an academic record should be judged by whether it serves the purpose for which it was intended. To define a measurable standard for the quality in academic records, the proposal considers five key attributes that should be evaluated for each item included in the record. Thus, quality in academic records is defined as having the attributes of legibility, completeness, consistency, correctness, and non-redundancy. To demonstrate the structure of an EAR, Figures A1–A3 in Appendix A present the entity relation diagrams of the university registration office, student marks, and examination scheduling.

- Legibility (L): Any entry in an academic record has to be legible, dated, timed and authenticated by the university.
- Completeness (CM): An academic record is considered complete if it has all items that all universities have agreed on during the previous stage.
- Correctness (CR): The academic record's correctness refers to the accuracy of its collected data.
- Consistency (CN): An academic record is considered consistent when the included data are reliable, and the data integrity has not been corrupted regardless of how often or in what way the data have been retrieved, viewed, stored, or processed.
- Non-redundancy (NR): Redundancy in an academic record indicates that the data of a record may be repeated by several universities.

Thus, by taking the previous attributes into consideration, the degree of a university is defined as the total quality of all EARs for all students stored in the database of that university.

$$\delta_i = \sum_{EAR=1}^{m} Q_{EAR} \tag{1}$$

where the quality of an EAR is defined as the product of its L, CM, CR, CN and NR attributes.

To compute the Legibility indicator, each item in the EAR will be checked for whether it is considered as a legal item or not. The classification process will be performed via the REM component installed on the university node. Legal items will be tagged with $i1$, while illegal items will be tagged with $i2$. Thus, $L_{EAR} = 1$ if all items of the EAR are considered as legal items; otherwise,

$$L_{EAR} = \frac{\sum i1}{\sum i1 + i2} \tag{2}$$

For the NR indicator, $NR_I = 1$ if all data stored in an academic record are unique and not repeated by any other university, otherwise the non-redundancy indicator will be divided among the universities that share that item. For the correctness, completeness, and consistency indicators, each item in the EAR will be classified via the REM as: $n1$, correct element; $n2$, incorrect element; $n3$, missing element; $n4$, extra element; and, $n5$, conflict and reduction element. Equations (3)–(5) measure the completeness, correctness, and consistency of an EAR.

$$CM_{EAR} = \frac{\sum n1 + n2 + n5}{\sum n1 + n2 + n3 + n5} \tag{3}$$

$$CR_{EAR} = \frac{\sum n1}{\sum n1 + n2 + n4 + n5} \tag{4}$$

$$CN_{EAR} = 1 - \frac{\sum n5}{\sum n1 + n2 + n4 + n5} \tag{5}$$

Accordingly, the degree of a university i is computed by Equation (6):

$$\delta_i = \sum_{EAR=1}^{m} Q_{EAR} = \sum_{EAR=1}^{m} L_{EAR}\, CM_{EAR} CN_{EAR} CR_{EAR} NR_{EAR} \tag{6}$$

For illustration, assume that a university would like to join the blockchain network. Assume that all universities that would like to join the blockchain network have to agree on 35 important items that should be included in an EAR and thus be measured for all EARs stored in their databases. Suppose that a university has two EARs, as detailed in Table 1. Table 2 shows how to compute the quality of the two academic records whose details are presented in Table 1. In Table 2, the EAR attributes, the quality of a record, and the degree of a university node are computed. At the end of the initialization stage, each university will have its degree that will be dynamically stored in the NC. Thus, the NC will automatically update the average degrees of nodes within the network as well as voter nodes. Note that a node with a degree that is greater than the average degree of the blockchain will be considered as a voter node, while the node that has the least degree among the nodes in the network will be assigned the task of generating the next new block. The proposal rewards the "block's creator" an incentive that will added to its degree to decrease its probability of recreating the next block instead of just creating a digital currency, thus achieving a fairness among universities and ensuring the sustainability of the system.

Table 1. Description of two academic records in a university node.

Indicator	EAR1	EAR2
L	35 items are legal	34 items are legal and one item is not authorized
n_1	15 correct elements	13 correct elements
n_2	7 incorrect elements	6 incorrect elements
n_3	5 missing elements	7 missing elements
n_4	3 extra elements	3 extra elements
n_5	5 conflict elements	6 conflict elements
NR	30 items are unique 2 items are shared with other 3 universities 3 items are shared with other 2 universities	28 items are unique 4 items are shared with other 3 universities 3 items are shared with other 2 universities

Table 2. An example of computing the quality of two academic records.

EAR No.	L	CM	CR	CN	NR	Q_{EAR}
1	$\frac{35}{35} = 1$	$\frac{15+7+5}{15+7+5+5} = 0.84$	$\frac{15}{15+7+3+5} = 0.50$	$1 - \frac{5}{15+7+3+5} = 0.83$	$\frac{30+\frac{2}{3}+\frac{3}{2}}{35} = 0.92$	0.32
2	$\frac{34}{35} = 0.97$	$\frac{13+6+6}{13+6+7+6} = 0.78$	$\frac{13}{13+6+3+6} = 0.46$	$1 - \frac{6}{13+6+3+6} = 0.79$	$\frac{28+\frac{4}{3}+\frac{3}{2}}{35} = 0.88$	0.24
						0.56

4.3. Adding a New Student Node

The procedures of adding a new student node begins when a request is sent by a university node. The university node sends the Ethereum address of the new student node, its ID and the requested role to the NC for validation. Similar to the process of validating a request sent for adding a university node, voter nodes validate and authenticate that received request by ensuring and confirming that the received request is related to a legitimate student and the non-existence of a registered student matching that received ID. If the request is accepted and validated, the NC updates its local memory with the student's ID, its Ethereum address and a "student" role. The NC creates a new SRHC for the new student node whose address will be forwarded to the university. The new student's account information will be sent to the student node from the university node that forms the request.

4.4. Registering a Student Node

The process of registering a student can be viewed as an example of generating a stewardship among two different nodes: one stores and manages the data for the other (i.e., a university node stores and manages the data for a student node). A student registration process is performed whenever a new student visits a university.

The process begins by ensuring and confirming that the student node already is a registered node in the blockchain system. The DB Manager of the university node, which provides an access interface to the existing database, sends the student's Ethereum address along with its "student" role to the NC for verification. The NC ensures that the registration process will be accomplished for a student node that already registered in the blockchain. The NC returns a Boolean value for confirmation. Otherwise, the process of adding a new student node has to be completed first. Upon confirmation, the university node sends the student information (i.e., the student's Ethereum address, and its ID) in a transaction to its SRHC. The SRHC confirms whether the student is a new student or not.

Upon confirmation, the SRHC of the university node requests to generate a new stewardship with the student who can accept or reject that request. The SRHC of the university node will generate a new entry with the Ethereum address of the student node, its ID, "waiting approval" status, and last update of the status. Similarly, the SRHC of the student node will create a new entry with the Ethereum address of the university node, its ID, "waiting approval" status, and last update of the status. When the student accepts the request, the status field of both the university and the student SRHC will be updated with "acknowledged student approval" along with the last_update field. Otherwise, the process will be canceled, the status field will be updated with "acknowledged student approval", and a notification will be sent to the university node. After accepting the request and updating the SRHC of the university, the university's SRHC creates a new RC for the new stewardship. The RC then accordingly fills the student's Ethereum address, his/her ID in the Owner field and all the university database related information. The RC sends its address to the SRHC of the university and the student nodes to update their "RC.add" data field for future reference.

4.5. Adding a New Academic Record Via a University Node

The procedures of adding a new academic record starts after establishing a stewardship between the university and the student nodes and thus having a shared RC. First, internal encryption in the university node begins the process of adding a new record. When a new record is created by a university node, that record will be transferred to the DB Manager. It creates an access link (AL) of a free location in the university existing database and hashes both the generated access link *h(AL)* and the record *h(EAR)*.

The DB Manager forwards the created access link, and the student record to the Cipher/Decipher Manager for encryption. The Cipher/Decipher Manager generates a symmetric key (SMK), encrypts the new record and link with that key and then encrypts that generated symmetric key with the public keys of the: university, student and set of proxies. The Cipher/Decipher Manager sends the encrypted

record to the DB Manager to store. In addition, all other encrypted data will be sent to the DB Manager to create a log indicating the creation of the new record since the history of all access will be stored in the blockchain to provide a full view of all events that happened to each record. The hash of the created log will be calculated and stored in the DB Manager for block verification later. Thus, ensuring the integrity of data since if any part of the data is changed, all involved nodes will notice the alteration. Then, the log will be sent to the Cipher/Decipher Manager for encryption with the public key of the university node.

The university node sends the student's ID to its SRHC that will return the associated RC address. The university node then sends the record information (filename of the record, hash value of the access, and hash value of the student's record, the encrypted symmetric keys $E_{pk}(SMK)$, and the log) to the RC. The RC stores the filename of the record, the hash value of the access link, and the hash value of the student's record. The RC then creates a new ACC for the record and forwards the encrypted symmetric keys $E_{pk}(SMK)$. The ACC auto-creates the access and permissions information for the record, i.e., student and university permissions, and then sends its address to the RC for its reference. On the other hand, the LC updates its entries with the received encrypted log, the associated Ethereum address of the university node, the "new log" status to indicate that the new log has not been added to the blockchain yet, and the timestamp of the last status update. At the end, the encrypted access link is sent to the student over HTTPS who will store that link in its Cipher/Decipher Manager and will be used when the student would like to read his/her record. Additionally, when the new record is created, the university node notifies the NC to updates the associated degree of the university node. The NC informs the REM to add the value of the added record to the node's degree and to return it to perform the update.

4.6. Editing a Record by a University Node

The university node sends the student's ID to its SRHC to retrieve the associated RC address. Upon receiving the RC address, the university node then sends the filename of the requested record and its Ethereum address to the RC. The RC forwards the request to the ACC to check whether that received Ethereum address has a permission (i.e., "owner" access level) on the requested record or not. If the university node has a permission, the AAC forwards the university's encrypted symmetric key $E_{pk_{university}}(SMK)$ to the RC. The RC in turns forwards the received key to the university node.

The Cipher/Decipher Manager in the university node first decrypts the received symmetric key using its private key and then decrypts the access link with that symmetric key. The DB Manager of the university node follows the related access link and then retrieves the encrypted EAR from the database for editing. Note that, when a record is modified, its hash value will also be changed [24]. Thus, the DB Manager, after modifying the record, calculates the new hash of the modified record $h(EAR)$. The DB Manager sends the student's ID to its SRHC to retrieve the associated RC address. The new hash value will be sent to the RC for updating. Moreover, the DB Manager creates a log indicating the process of record editing, hashes the log and then forwards the log to the Cipher/Decipher Manager for encryption. The encrypted log will then be forwarded to the LC. The LC adds a new entry with the received encrypted log, a "new log" status to indicate that the new log has not been added to the blockchain yet, and a timestamp indicating the last status update. Additionally, when the editing the student's record, the university node notifies the NC to updates the associated degree of the university node. The NC informs the REM to reevaluate the value of the record and thus updating the node's degree. The REM performs the calculations and returns the new degree to the NC for updating.

4.7. Access a Record from Student Node

A student node sends the university's ID to its SRHC to retrieve the associated RC address. Upon receiving the RC address, the student node then sends the filename of the requested record and Ethereum address of the student to the RC. The RC forwards the request to the ACC to check whether that received Ethereum address has a permission (i.e., "read/edit" access level) on the requested record

or not. If the student node has a permission, the AAC forwards the student's encrypted symmetric key $E_{pk_{student}}(SMK)$ to the RC. The RC in turns forwards the received key with the database access information to the student node.

The Cipher/Decipher Manager in the student node first decrypts the received symmetric key using its private key and then decrypts the access link with that symmetric key. The DB Manager of the student node follows the related access link and retrieves the encrypted EAR from the university's database. Since students can access their nodes via online wallets, records access can be performed by any device with Internet connection, thus improving the interoperability of EARs. Moreover, the DB Manager creates a log indicating the process of reading the record, hashes the log and then forwards the log to the Cipher/Decipher Manager for encryption. The encrypted log will then be forwarded to the LC. The LC adds a new entry with the encrypted received log, a "new log" status to indicate that the new log has not been added to the blockchain yet, and a timestamp indicating the last status update.

4.8. Generating a Student Transcript

A student node sends a request to generate a transcript to a university node. The university node sends the student's ID to its SRHC to retrieve the associated RC address. Upon receiving the RC address, the university node then sends the filename of the requested record and the student Ethereum address to the RC. The RC forwards the request to the ACC to check whether that received Ethereum address has a permission on the requested record or not. If the student node has a permission, the AAC forwards the university's encrypted symmetric key $E_{pk_{university}}(SMK)$ to the RC. The RC in turns forwards the received key to the university node.

The Cipher/Decipher Manager in the university node first decrypts the received symmetric key using its private key and then decrypts the access link with that symmetric key. The DB Manager of the university node follows the related access link and then retrieves the encrypted EAR from the database to create the transcript. The created transcript will be forwarded to the Cipher/Decipher Manger that in turn will sign the transcript with the private key of the university and then encrypt the signed transcript with the public key of the student. In addition, the DB Manager creates a log indicating the process of creating a transcript, hashes the log and then forwards the log to the Cipher/Decipher Manager for encryption. The encrypted log then will be forwarded to the LC. The LC adds a new entry with the received encrypted log, a "new log" status to indicate that the new log has not been added to the blockchain yet, and a timestamp indicating the last status update. The university node forwards the encrypted signed transcript to the student node via HTTPS. The student node can retrieve the transcript by the use of the student private key and ensure it comes from the desired university node by the use of the university public key.

4.9. A University Node Reads a Record from Another University Node

The process of reading a record that is stored in a university node from another university node utilizes the timed-based deposit-box mechanism to increase both the accessibility and the security of EARs systems.

Suppose that there are two university nodes, a and B, where University B would like to read a specific student's record from University A. University B generates a request to read the record first, signs that generated request by its private key for authorization, and then encrypts the signed request with the public key of University A. Over HTPPS, the encrypted signed request will be sent to University A. Upon receiving the request, University A decrypts the request with its private key and then decrypts it with the public key of University B to ensure that the university is the one that it claims to be.

First, Node A will send the ID of the student to its SRHC to return the associated RC address. University Node A then sends the filename of the record to the RC to retrieve the associated ACC address. University Node A then sends the Ethereum address and the access level request to the ACC. The ACC then forwards the Ethereum address of University B and its request for the NC to

verify whether University B is an authorized university registered on the system. Upon receiving the verification from the NC, the ACC generates a new entry with the Ethereum address of Node B, the Access Level, and the "request new level" status, and the timestamp of the last status update. The ACC requests a change in the access level from the file owner (i.e., student), and updates both its status field to "waiting approval" and last-update. If the student accepts the request, the ACC updates the access level with "temp-read" for the applicable file with the "approved" status and the last-update. Once the request has been approved, the ACC sends a notification to University Node B indicating that a new access level is assigned to it. The ACC also notifies the RC to create a new entry in the DPC with the Ethereum address of University B.

The RC then sends the Ethereum address of University B, and the file name to University Node A. The DB Manager retrieves the record and forwards the record with the received public key to the Cipher/Decipher Manager for encryption. The DBC will be updated by storing the encrypted file for a certain time specified by the Time field. Over HTTPS, University A sends the DPC address to University B to access the requested record by decrypting it using its private key. The DB Manager of Node B will update the LC entry with an encrypted log indicating the process of reading the record. The LC updates its status with "new log" to indicate that the new log has not been added to the blockchain yet, and the timestamp of the last status update.

4.10. Generating, Verifying and Appending a New Block

The process of generating a new block begins by selecting the university that is responsible for performing this computational task. Based on the degree that each university node owns, the selection process is performed. According to the selection method in the proposed incentive mechanism, universities with more degrees maintain more academic records or records with higher values. Thus, they are less likely to be selected. The NC assigns this task to the university with the lowest degree among other universities in the system.

Assume that University Node A is selected for the task of generating a new block. University A sends a request to the LC. The LC forwards the request to the NC to verify that University A is an authorized university on the system and it is selected for performing the task. The NC returns a verification to the LC. Upon receiving the verification, the LC sends all encrypted Logs whose status is "new log" to University A. After collecting all logs, University A creates a new block that includes all logs, broadcasts to the blockchain network about the new block and calls for verification.

Each involved university node verifies its logs in the new block. Each involved node decrypts the log with its private key, computes the hash value of the log after decryption and compares the computed hash with the value stored in its DB Manager. Each node then sends a signed proof to the University Node A. Upon receiving all the signed proofs, University A notifies the NC to update the degree of University A by adding the incentive value c to its current degree. According to the proposed incentive mechanism, the university that will be chosen to generate a new block will get an incentive c as a reward upon successfully verifying the block by other universities. The value of c depends on the size of the blockchain network and the distribution of universities degrees. Thus, c will be defined as a fraction of the average degrees in the network. University A then broadcasts to all universities to append the new block. After appending the new block, the LC automatically updates the status field for all logs which are added to the chain as "appended" and the last_update field. The process of generating, verifying, and appending a new block is summarized in Figure 6.

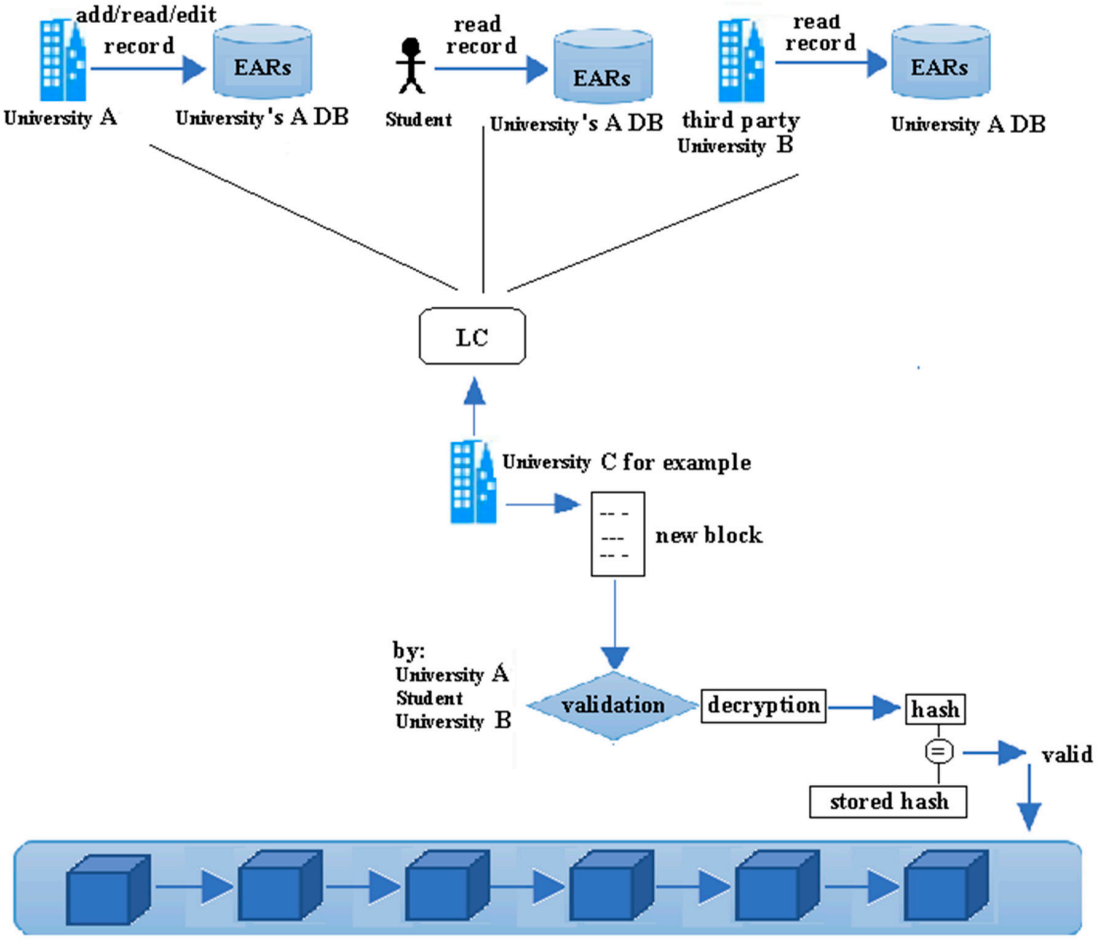

Figure 6. Generating/validating/appending a new block.

5. Evaluation and Discussion

5.1. Experimental Setting

For evaluation, experiments were performed on a computer system with an Intel Core i7-5557U 3.10 GHz processor, 16 GB memory, and Windows 10 (64 bit) operation system. The Ethereum platform, which is an open source platform featuring smart contract (scripting) functionalities, was used for implementing the proposed system. The smart contracts were written in Solidity and deployed with Truffles with no capacity restrictions on the stored data size. To allow the interaction with an Ethereum node using HTTPS, the web3.js library was employed. The open source Apache JMeter was used as a functional and performance measurement tool for testing the services provided on the web. The experimental parameters and their values are given in Table 3.

Table 3. Parameters used in the experiments.

Description	Values
The submitted Queries	1000–10,000
The stored EARs	10,000–100,000
Number of nodes	1000–10,000

Several tests based on two parameters were performed: the number of submitted queries and the size of the stored academic records. The measurement of the performance was based on the following

metrics: the average response time, the throughput, and the communication overhead or the average number of messages sent per a node. Only one parameter was changed each time so that any changes in the performance would be based solely on this parameter. In fact, results achieved from these tests were used to study the behavior of the proposal for: (1) random systems with different number of nodes and roles; and (2) systems with different academic records' size.

To study the effects of changing the distribution of the submitted queries on the average response time, the throughput, and the communication overhead, theses queries were varied from 1000 to 10,000 query units, and the distribution of the submitted queries among the nodes were carried in the following manner.

- 25% variations: Similar request distributions among nodes.
- 50% variations: The intermediate situation where the majority of queries are submitted to 50% of nodes.
- 75% variations: The advanced intermediate situation where the majority of queries are submitted to 75% of nodes.

To study the effects of increasing the number of academic records stored in the university databases on the average response time, the throughput, and the communication overhead, the number of stored records were varied from 10,000 to 100,000, and the distribution of the records among the nodes were carried in the following manner.

- 25% variations: Similar records distributions among nodes.
- 50% variations: The intermediate situation where the majority of records are stored in 50% of nodes.
- 75% variations: The advanced intermediate situation where the majority of records are stored in 75% of nodes.

5.2. Results and Discussion

The results show that the average response time and the average number of messages sent per node increased as the total submitted queries was increased, as shown in Figures 7 and 8.

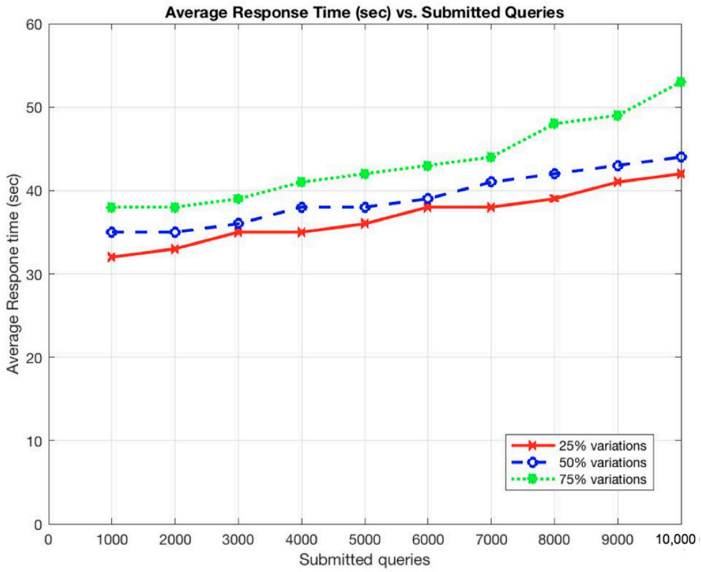

Figure 7. Average response time (s) vs. submitted queries.

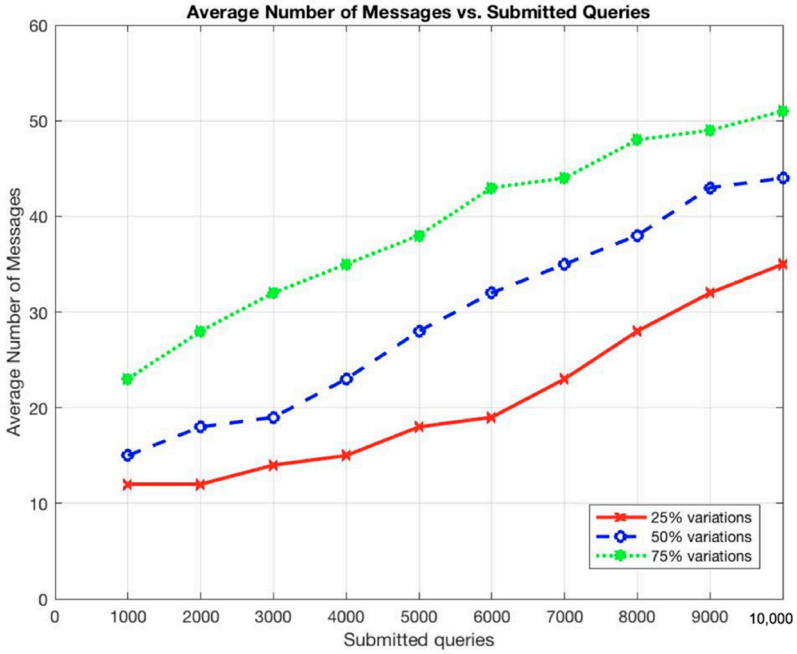

Figure 8. Average number of messages vs. submitted queries.

Similarly, the average response time and the average number of messages sent per node increased as the total number of stored records was increased, as shown in Figures 9 and 10.

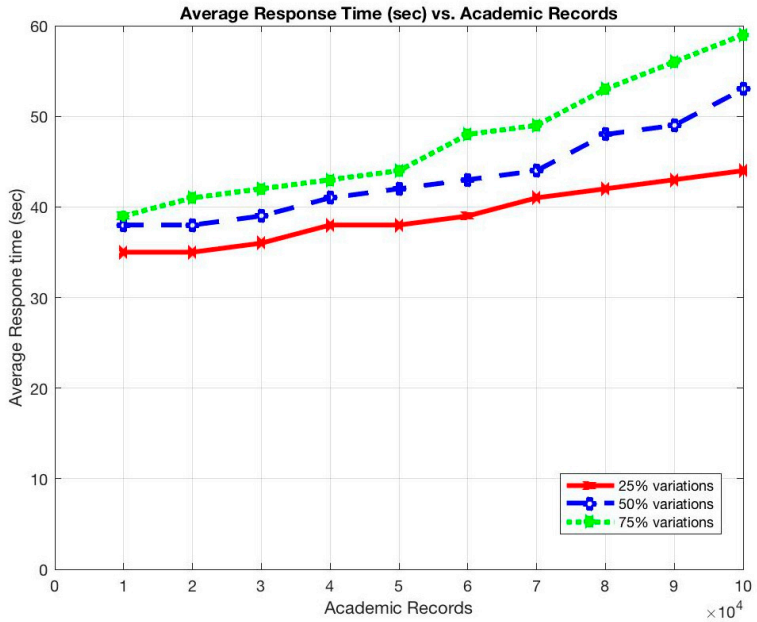

Figure 9. Average response time (s) vs. academic records.

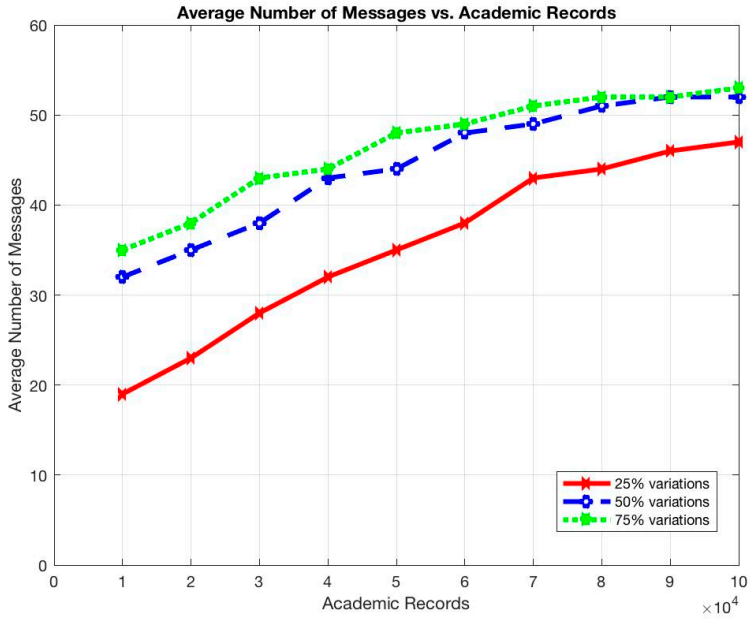

Figure 10. Average number of messages vs. academic records.

These situations are expected as the more queries to be submitted, the longer it takes for a query to be completed and the more communications among participant nodes to be occurred. In addition, the more records to be stored, the more participants to use the system, and, thus, the more queries to be submitted on the system. However, the throughput of the system remains constant even with the increments of the submitted queries or the stored records (Figures 11 and 12).

The stability of the system's throughput even when increasing the number of stored records and the number of submitted queries prove the ability of the system to handle and process a large dataset with high frequency at low latency as in EARs systems.

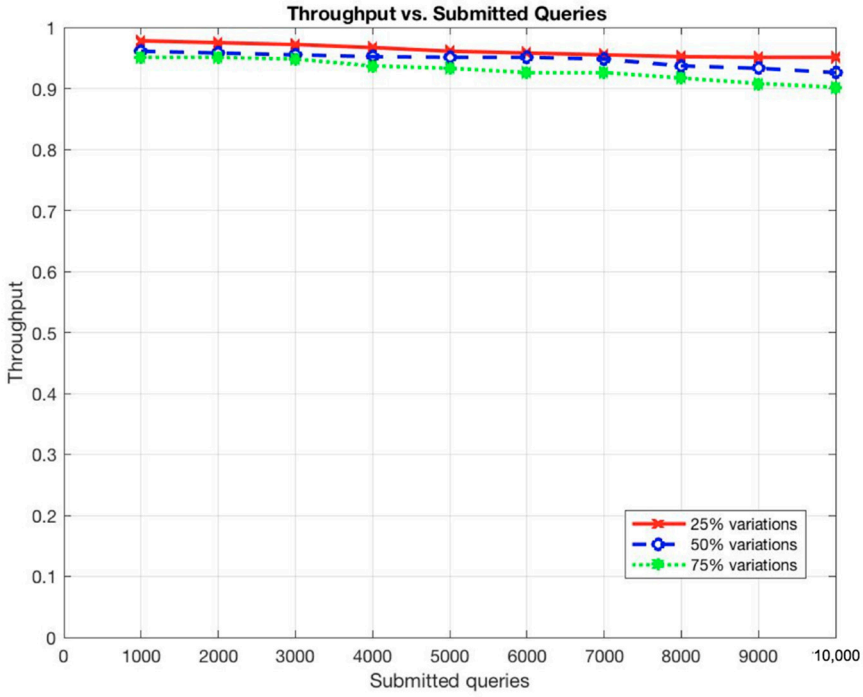

Figure 11. Throughput vs. Submitted queries.

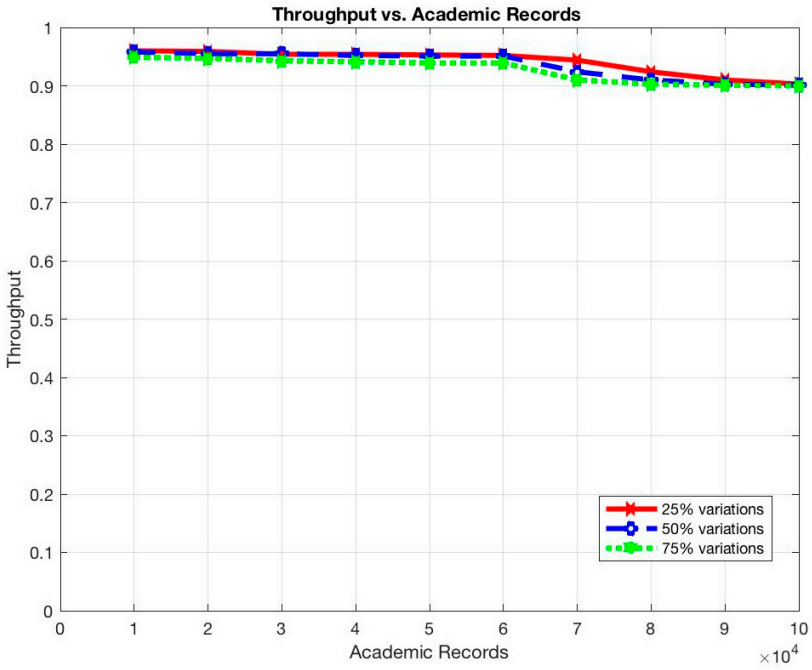

Figure 12. Throughput vs. Academic Records.

Actions in the proposed system are classified into off-blockchain and on-blockchain actions. The off-blockchain actions involve computing the degree of university nodes during the blockchain initialization, creating an access link, calculating its hash value and encrypting it when adding a new record. In addition, database storage and retrieval procedures are off-blockchain actions. The on-blockchain actions include retrieving and storing data values in smart contracts, sending internal transactions to link different contracts, and spawning new contracts using other contracts. The diversification of the system modules and the balance between the on-chain and the off-chain actions involve increasing the features of the proposed system while maintaining its performance.

The proposed system adopts the PoA consensus algorithm, which plays a significant role in the system performance as it can handle more transactions per second compared with the PoW and the PoS. The PoA minimizes the intensive of computations and increases the system performance as it provides lower transaction acceptance latency and steady time intervals for issuing blocks. According to the proposal, two rounds are required for the generation and the validation of the new block. The block's creator node first sends the created block to only involved nodes (i.e., no need to send the block to all participants in the network). These nodes validate their logs and then reply with a verification. A block will be appended when receiving all associated verifications. By adopting PoA, the number of messages sent for block generation and validation will be decreased, thus ensuring the effectiveness of the system. The adoption of timed-based smart contracts ensures a reasonable period for the transactions and the computations performed on the data. The loss of connectivity resets the timers to zero and the data are destroyed using instructions stored in the smart contracts.

5.3. The Real Impact of UniChain in Managing EARs

Generally, an EAR is a record in a database that stores academic information about students in a digital format. There are two ways to issue an academic record: it could be created digitally or be digitized from existing paper records. Record digitalization refers to the process of transforming different materials of the records into digital format while preserving the record characteristics of reliability, usability, authenticity and integrity [25]. An EARs management system is a digital tool with a web and/or mobile interface that enables access to the record content while maintaining its important characteristics. Relational databases are used to implement the existing EARs management systems.

Currently, a student may visit more than one institution during his/her study, for instance, exchange-students or credit-mobile students. Credit-mobile students refer to "study-abroad" or registering in one university while taking courses at another, such as those in the EU's Erasmus program. The number of credit-mobile students are increasing and their destinations diversifying [26]. These students remain enrolled in their home countries while receiving a small number of credits from foreign institutions. There were at least 1.6 million students from abroad who were undertaking tertiary level studies across the EU in 2016 [27]. In fact, the EAR will be stored in the database of the university that issued the record, which will be the only eligible university for editing it. In other words, the EARs of a student are placed in different institutions databases, thus providers cannot have a comprehensive overview of all the records of a single student. Academic provider databases are partly open to students and other academic providers with different specified permissions. Students with access rights could query their EARs from different universities. Universities with access rights could query EARs of a common student from other university when there is a need. These situations cause a lack of coordinated data management and exchange. Hence, academic records are isolated and fragmented, rather than cohesive.

Moreover, as academic providers are solely maintaining the records in which they had issued, there is a difficulty to confirm data integrity when a malicious entity modifies that single copy of the record or even when a record is removed from a provider database. The need for multiple access to the EARs had raised the interoperability challenges between students and academic providers which pose additional barriers to effective data sharing.

Additionally, as technology is constantly evolving, several advanced techniques are developed to violate digital privacy and security. Unfortunately, academic records are considered as major targets for information theft since they include private and sensitive information, e.g., the students' names, identity numbers, contacts info and addresses. In 2018, the hacking of Australian National University resulted in the theft of 19 years' worth of data. Electronic records protection is a complex and massive undertaking [28]. According to the identity intelligence company 4iQ [29], the number of data breaches increased more than 400% in 2018, exposing almost 15 billion records, and the average cost of a security breach is $17 million.

The following points summarize the problems addressed by current EARs management systems:

- the lack of coordinated data management and exchange, as EARs are fragmented and isolated, rather than cohesive;
- the need for multiple access to the EARs raises the interoperability challenges between students and universities;
- the inability to transfer or access EARs and testimonials across multiple institutions, making it difficult to achieve effective data sharing; and
- the lack of protection and control of private information by data owners.

On the other hand, the proposed UniChain system will be built above the existing academic provider databases to facilitate the integration with the existing systems. To reduce the requirements of storing EARs in the blockchain and to utilize the existing systems, EARs will be continuously stored in the provider databases. As universities currently maintain and manage the EARs, while students can only read data, provider nodes in the proposed design will be responsible for the maintenance of the blockchain. According to an article that was published in the Applied Sciences Journal in June 2019 [30], a blockchain could bring several benefits to education: (1) improving security that includes data protection, privacy, and integrity; (2) providing better control on how students' data are accessed and by whom; (3) enhancing accountability and transparency; (4) enhancing trust between all included parties and ease the communication between them; (5) lowering the cost as the nature of blockchain technology can help in reducing the unnecessary cost associated with the transactions and storage of data; (6) authenticating students' identities as well as their digital certificates; (7) the efficiency of data

exchange and the management of students records; (8) enhancing interoperability; and (10) supporting future career.

The UniChain system employs the blockchain technology, which is a collection of techniques (cryptography and hash functions) to create a chain of data where each new piece of data is linked to the previous ones by a cryptographic hash function. Therefore, it significantly increases the difficulty of attack and improves the privacy and security of EARs. All accesses to the EARs will be performed through the blockchain, and accordingly the history of those accesses will be stored in the blockchain to provide a full view of all events occurred to EARs. Thus, it ensures the integrity of data and prevents misuse of a student EAR. All logs details in addition to the record ownership metadata will be added to the chain.

Sensitive information that are placed on the blockchain are encrypted to decrease the possibility of being accessed by unauthorized entity. UniChain system increases the level of data obfuscation by separating sensitive information via the use of SRHC, RC and ACC. The use of deposit-box technique is employed to solve the problem of transferring encrypted messages among nodes with no need to share symmetric key.

The proposed framework employs the hashing methods, i.e., SHA-256, to ensure data integrity. UniChain keeps a hash value of the link that will be created during the record's issue to access the EAR in the blockchain instead of keeping the link itself. To access a record, the encrypted query link will be sent over HTTPS to the associated participant who has access rights. Therefore, its hash value stored in the blockchain ensures that no alterations have been made outside the blockchain during the transfer as the value of the hash is unique to the original document. For further security, UniChain will store the query link, the key and the EARs in different locations.

Privacy is maintained in the UniChain by employing timed-based smart contracts for governing transactions. Security and access control are maintained by the adoption of advanced encryption and authentication techniques throughout the blockchain. Interoperability, auditability, and accessibility are provided by the use of smart contracts and comprehensive logs.

For creating, validation, and appending new block, the proposed system employs a new incentive mechanism integrated with the Proof of Authority (PoA) consensus algorithm. The proposal leverages the degree of providers nodes from the perspective of EARs systems by measuring their efforts regarding maintaining academic records and creating new blocks. Provider nodes with fewer degrees are more likely to be selected for creating the new block. The proposal rewards the "block's creator" an incentive that will added to its degree to decrease its probability of recreating the next block instead of just creating a digital currency, thus achieving a fairness among providers and ensuring the sustainability of the system.

6. Conclusions

In this paper, a design of a blockchain-based system, namely UniChain, for managing EARs is proposed. UniChain is designed to be compatible with the existing EARs' databases and to provide interoperable, secure, and efficient access to EARs by universities, students and third parties, while maintaining the students' privacy. In UniChain, the blockchain maintenance including creation, verification and appending of new blocks is the responsibility of universities, while allowing students to securely control accesses their EARs. Privacy is maintained in the UniChain by employing timed-based smart contracts for governing transactions and monitoring the computations performed on the EARs through the enforcement of the acceptable usage policies. The adoption of hashing techniques ensures the integrity of data. Security and access control are maintained by the adoption of advanced encryption and authentication techniques throughout the blockchain. Interoperability, auditability, and accessibility are provided by the use of comprehensive logs. The proposal is independent of any specific system, and its variations can potentially accommodate other similar systems with multiple access for electronic records. This work proposes a new incentive mechanism integrated with the PoA for mining. It leverages the degree of universities regarding their efforts on maintaining academic

records and creating new blocks. The proposed mechanism rewards the "block's creator" an incentive to be added to its degree and accordingly decreasing its probability of recreating the next block instead of just creating a digital currency, thus achieving the fairness and the equality among universities and ensuring the sustainability of the system. Extensive experiments were conducted to evaluate the UniChain performance on different aspects, including response time, throughput, and communication overhead. The results indicate the efficiency of the proposal in handling a large dataset at low latency.

Author Contributions: Conceptualization, E.-Y.D. and Y.-A.D.; methodology, E.-Y.D.; formal analysis, E.-Y.D. and Y.-A.D.; writing—original draft preparation, E.-Y.D., Y.-A.D. and S.-M.Y.; writing—review and editing, Y.-A.D. and S.-M.Y.; supervision, S.-M.Y; and project administration, S.-M.Y.

Funding: This research received no external fund.

Acknowledgments: We thank Palestine Technical University—Kadoori, Bank of Palestine, and Taawon association for their support.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

The entity relation diagrams of the university registration office, student marks, and examination scheduling are shown in Figures A1–A3.

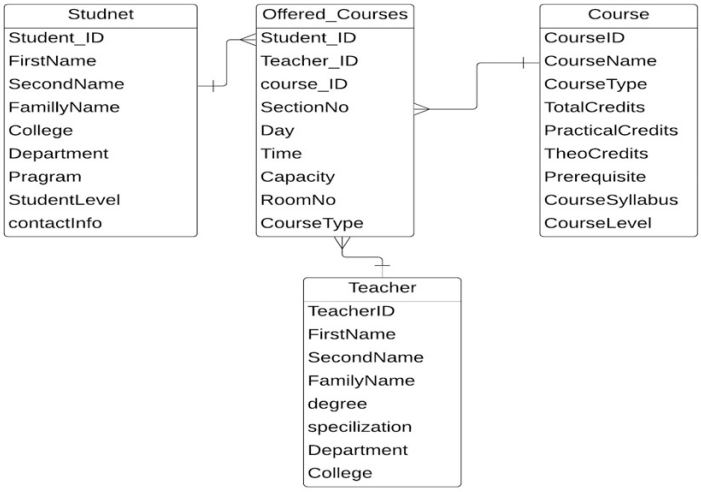

Figure A1. The ERD of the university registration office [31].

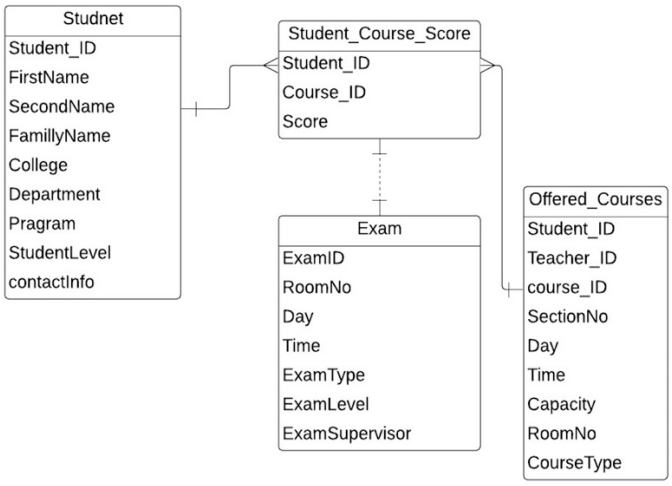

Figure A2. The ERD of student score [32].

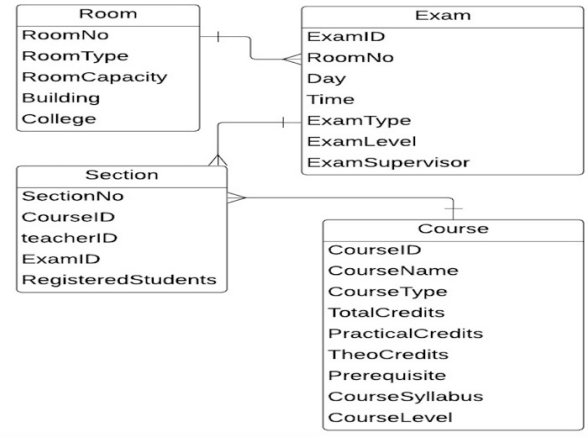

Figure A3. The ERD of examination scheduling [33].

References

1. Bessa, E.E.; Martins, J.S.B. A Blockchain-based Educational Record Repository. In Proceedings of the 7th International Workshop on ADVANCEs in ICT Infrastructures and Services, Praia, Cape Vert, 21–22 January 2019.
2. Nguyen, T. Gradubique: An Academic Transcript Database Using Blockchain Architecture. Master's Thesis, San Jose State University, San Jose, CA, USA, 14 December 2018.
3. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 14 November 2019).
4. Szabo, N. The Idea of Smart Contracts. Nick Szabo's Pap. Concise Tutor. 1997. Available online: https://nakamotoinstitute.org/the-idea-of-smart-contracts/ (accessed on 14 November 2019).
5. Liu, X.; Muhammad, K.; Lloret, J.; Chen, Y.-W.; Yuan, S.-M. Elastic and cost-effective data carrier architecture for smart contract in blockchain. *Future Gener. Comput. Syst.* **2019**, *100*, 590–599. [CrossRef]
6. Modi, R. *Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain*; Packt Publishing Ltd.: Birmingham, UK; Mumbai, India, 2018; ISBN 978-1-78883-138-3.
7. Rooksby, J.; Dimitrov, K. Trustless education? A blockchain system for university grades. In *New Value Transactions: Understanding and Designing for Distributed Autonomous Organisations*; Workshop DIS: Edinburgh, UK, 2017.
8. Ocheja, P.; Flanagan, B.; Ueda, H.; Ogata, H. Managing lifelong learning records through blockchain. *Res. Pract. Technol. Enhanc. Learn.* **2019**, *14*, 4. [CrossRef]
9. Grech, A.; Camilleri, A.F. GrBlockchain in Education. *Education* **2018**. [CrossRef]
10. Sharples, M.; Domingue, J. The blockchain and kudos: A distributed system for educational record, reputation and reward. In Proceedings of the 11th European Conference on Technology Enhanced Learning, Lyon, France, 13–16 September 2016.
11. Chen, G.; Chen, B.X.; Lu, M. Nian-Shing Exploring blockchain technology and its potential applications for education. *Smart Learn. Environ.* **2018**, *5*. [CrossRef]
12. Yumna, H.; Khan, M.M.; Ikram, M.; Ilyas, S. Use of Blockchain in Education: A Systematic Literature Review. In Proceedings of the Asian Conference on Intelligent Information and Database Systems, Yogyakarta, Indonesia, 8–11 April 2019.
13. Turkanović, M.; Hölbl, M.; Košič, K.; Heričko, M.; Kamišalić, A. EduCTX: A Blockchain-Based Higher Education Credit Platform. *IEEE Access* **2018**, *6*, 5112–5127. [CrossRef]
14. Nazaré, J.; Duffy, K.H.; Schmidt, J.P. What We Learned from Designing An Academic Certificates System on the Blockchain. MIT Media Lab. 2016. Available online: https://medium.com/mit-media-lab/what-we-learned-from-designing-an-academic-certificates-system-on-the-blockchain-34ba5874f196#.luos3nzc5 (accessed on 14 November 2019).
15. Blockcerts Blockcerts Universal Verifier. Available online: https://www.blockcerts.org (accessed on 14 November 2019).

16. Mattingly, J. A Virginia University Is Now Issuing Degrees on Blockchain. Available online: https://www.richmond.com/news/local/education/a-virginia-university-is-now-issuing-degrees-on-blockchain/article_1ec71f80-0beb-5bdf-9ee3-8af80df2f32f.html (accessed on 14 November 2019).
17. Sutton, M. UAE University launches Blockchain Records App. Available online: https://www.arabianindustry.com/technology/news/2019/feb/20/uae-university-launches-blockchain-records-app-6042569/ (accessed on 14 November 2019).
18. Awaji, B.; Solaiman, E. Online Education using Blockchain and Smart Contracts. In Proceedings of the 11th International Conference on Computer Supported Education, Crete, Greece, 2–4 May 2019.
19. GitHub PoA Private Chains. 2018. Available online: https://github.com/paritytech/wiki/blob/master/Proof-of-Authority-Chains.md (accessed on 14 November 2019).
20. Anita, A.; Flora, A.; Giovanni, C.; Francesco, G.; Nicla, I.; Antonino, M. A Study on Textual Features for Medical Records Classification. *Stud. Health Technol. Inform.* **2014**, *207*, 370–379.
21. Ethereum Clients. Available online: http://www.ethdocs.org/en/latest/ethereum-clients/index.html (accessed on 14 November 2019).
22. JSON RPC. Available online: https://github.com/ethereum/wiki/wiki/JSON-RPC#json-rpc-endpoint (accessed on 14 November 2019).
23. Prusty, N. *Building Blockchain Projects: Building Decentralized Blockchain Applications with Ethereum and Solidity*; Packt Publishing Ltd.: Birmingham, UK; Mumbai, India, 2017; ISBN 978-1-78712-214-7.
24. Stallings, W. *Cryptography and Network Security: Principles and Practice*, 4th ed.; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2006; ISBN 978-0-13-187316-2.
25. ISO Technical Committees. Information and Documentation Records Management—Part 1: Concepts and Principles. 2016. Available online: https://www.iso.org/standard/62542.html (accessed on 14 November 2019).
26. International Students. Available online: https://migrationdataportal.org/themes/international-students (accessed on 22 October 2019).
27. Learning Mobility Statistics. Available online: https://ec.europa.eu/eurostat/statistics-explained/index.php/Learning_mobility_statistics (accessed on 22 October 2019).
28. The Guardian Australian National University Hit by Huge Data Breach. Available online: https://www.theguardian.com/australia-news/2019/jun/04/australian-national-university-hit-by-huge-data-breach (accessed on 14 November 2019).
29. 4iQ. The Changing Landscape of Identities in the Wild the Long Tail of Small Breaches. 2019. Available online: https://4iq.com/2019-4iq-identity-breach-report-the-changing-landscape-of-identities-in-the-wild-the-long-tail-of-small-breaches/ (accessed on 14 November 2019).
30. Alammary, A.; Alhazmi, S.; Almasri, M.; Gillani, S. Blockchain-Based Applications in Education: A Systematic Review. *Appl. Sci.* **2019**, *9*, 2400. [CrossRef]
31. University Registration Office. Available online: https://circle.visual-paradigm.com/university-registration-office/ (accessed on 11 August 2019).
32. Student Score. Available online: https://circle.visual-paradigm.com/student-score/ (accessed on 11 August 2019).
33. Examination Scheduling. Available online: https://circle.visual-paradigm.com/examination-scheduling/ (accessed on 11 August 2019).