Tsai, C.-T., Wu, J.-L., Lin, Y.-T., Yeh, M. K.-C. (2022). Design and Development of a Blockchain-Based Secure Scoring Mechanism for Online Learning. *Educational Technology & Society, 25* (3), 105-121.

Design and Development of a Blockchain-Based Secure Scoring Mechanism for Online Learning

Cheng-Ting Tsai[1], Ja-Ling Wu[2], Yu-Tzu Lin[3*] and Martin K.-C. Yeh[4]

[1]Department of Electrical Engineering, National Taiwan University, Taiwan // [2]Department of Computer Science, National Taiwan University, Taiwan // [3]Graduate Institute of Information and Computer Education / Institute for Research Excellence in Learning Sciences, National Taiwan Normal University, Taiwan // [4]Information Sciences and Technology, Pennsylvania State University Brandywine, USA // chtitsai@gmail.com // wjl@cmlab.csie.ntu.edu.tw // linyt@ntnu.edu.tw // kqy1@psu.edu

[*]Corresponding author

ABSTRACT: With the rapid increase of online learning and online degree programs, the need for a secure and fair scoring mechanisms in online learning becomes urgent. In this research, a secure scoring mechanism was designed and developed based on blockchain technology to build transparent and fair interactions among students and teachers. The proposed scoring mechanism was implemented by employing the Ethereum blockchain and its three autonomous smart contracts. The robustness and feasibility of the system was then verified with experiments. The resulting system is shown to be superior to existing online learning systems because it prevents answer tampering. In addition, fairness can be improved with blockchain protocols and a collaborative scoring policy. Lastly, this system helps manage interactions among students and teachers during the process of educational assessment, and encourages all on-chain members to trust the online learning process. These advantages improve peer evaluation and self-directed learning that are essential for a student-centered and collaborative learning environment.

Keywords: Blockchain, Ethereum, Cryptography, Online learning, Online assessment

1. Introduction

Over the last few years, researchers and application developers took blockchain technology more seriously since its most well-known realization, Bitcoin, was introduced by Satoshi Nakamoto in 2008 (Nakamoto, 2008). The success of Bitcoin shows that blockchain techniques do contribute to the stability and liveliness of a system where data and executive activities are decentralized—supervised and maintained by all members of the chain. This decentralization feature makes each interaction immutable, secure, and transparent. This explains why blockchain technology has been applied to various fields, such as profit sharing and credit scoring (Jain et al., 2019), where members are treated equally, and when the legitimacy of transferred information has to be considered seriously. In the field of education, security issues for teacher-student interaction were often not discussed in a traditional learning environment. However, as online learning and online degree programs are growing rapidly (Porter, 2015), a reliable and secure scoring mechanism for learning management systems is required to prevent possible cheating and guarantee fair and accurate assessment results. In fact, COVID-19 has accelerated teaching mode from physical to online format (Pavlov & Katsamakas, 2021), the need of secure and private learning systems becomes urgent.

Traditional online learning platforms might suffer from security vulnerability due to the lack of security mechanisms and unequal privileges. Existing research focused more on preventing cheating during online examinations by applying biometrics technologies (Apampa et al., 2010; Traoré et al., 2017; Sabbah, 2017) or multi-factor authentication (Urosevic, 2019) to increase the security during examination, introducing a live-remote human proctor for exam monitoring (Lilley et al., 2016), or proposing a conceptual framework to provide guidelines for online examinations (Ngqondi et al., 2021). However, answer tampering after tests or subjective scoring biases have not been addressed. Apparently, if the on-platform activities are not traceable, the system may not be trustworthy in ensuring fair and unforged teacher-student interactions. Another common challenge in assignment/examination scoring is scoring open-ended questions (e.g., essay questions and calculation problems). Teachers might have different opinions and biases, which leads to disagreement about the assessment results. There is still limited research studying how to develop security models for online learning. Although previous research has tried to develop an architecture of trustworthy web services for secure assessment for collaborative learning (Caballé et al., 2017), such architecture was built for grid infrastructure. To solve the problem of scoring biases, collaborative scoring is a possible solution because it can include various opinions from different scorers, which is common in collaborative and project-based learning. It, however, might cause the bandwagon effect if there is no proper scoring mechanism. Therefore, it is required to design a secure scoring mechanism for fair and effective scoring. Moreover, with the increasing number of online courses, more and more educational records will be stored and shared virtually over an array of networks. Invariably, we are facing the risks associated with hackers and other unethical actors. Blockchain technology can help secure and protect data in this new education model for its ability to combine information security and share data virtually to conduct learning among a wide range of networks.

In this paper, a blockchain-based assignment scoring mechanism is implemented to achieve fair and transparent teacher-student interactions during assessment, with which on-chain members are anonymous and their interactive activities are immutably traceable. To demonstrate its feasibility and applicability, we implemented the system on the Ethereum architecture along with multiple cryptography algorithms. Our teacher-student interaction model was designed to make all members equal and remove flaws in the scoring system, such as biases (by teachers or teaching assistants) and answer tampering (by students). Teachers can only uncover the students' identities at the end of the course to ensure the fairness of scoring. Further, three autonomous smart contracts were designed to guarantee the fairness and the efficiency of assessment. Finally, the proposed mechanism was implemented and the feasibility and robustness were examined by experiments.

2. Literature review

2.1. Online Learning Management System

An online Learning Management System (LMS) is a platform providing services of administration, assessment, reporting, automation, and delivery of educational courses, on which interactions among teachers and students might affect the effectiveness of online learning (Wright, 2014). With the rapid growth of online education, assessment for online assignments/examinations becomes a significant issue. However, the architecture of the LMS might not be secure enough to prevent misconducts, and fairness of assessment might be affected by student and teacher perceptions. Previous research has investigated possible solutions to increase security and fairness of online learning. Some studied how to strengthen authentication and identification systems to increase examination security by employing biometrics technologies (Apampa et al., 2010; Traoré et al., 2017; Sabbah, 2017) or multi-factor authentication (Urosevic, 2019). Examination monitoring is a solution to ensure fairness, which can be implemented by a live-remote human proctor (Lilley et al., 2016). Instead of only considering cheating prevention, some systems focus on improving the security of the LMS architecture, including an architecture of secure assessment by trustworthy web services (Caballé et al., 2017), a conceptual framework to provide guidelines for online examinations (Ngqondi et al., 2021), and a secure assessment management system based on cryptography protocols (Castella-Roca et al., 2006). However, the methods are either too complex to implement in a general LMS or only suitable for specific infrastructure. In addition, some illegal behaviours, such as answer tampering after the examination, are ignored in those studies.

2.2. Blockchain in education

Blockchain can be used to carry and transfer any valuable assets, such as currency, copyrighted materials, knowledge, and records. In education, there are many valuable information, including research data, experimental records, scores, credits and certificates of degrees whose management, security and fairness are necessary and extremely important for all stakeholders. Therefore, blockchain might be a suitable vehicle to bring benefits to educations (Chen et al., 2018; Skiba, 2017; Hernandez-de-Menendez et al., 2020) and makes management of all the students' and educators' information fairly and efficiently. For online education platforms, such as MOOCs, where students and educators come from different places of the world to achieve their own educational goals. Then the learning environment becomes more diverse, establishing trust between each member becomes a significant and challenging task.

The report of the European Commission's Joint Research Centre (JRC) suggests that issuing certificates is an important application for education (Grech & Camilleri, 2017), which involves tracking learning data (e.g., portfolio and achievements) to approve certificates (Raimundo & Rosário, 2021). Many studies focused on using blockchain to manage, share, and verify degrees/certificates and credits (Sharples & Domingue, 2016; Turkanović et al., 2018), or research results and data (Hoy, 2017). Some research studied secure assessment mechanisms for online learning (Lam & Dongol, 2020; Sudaryono et al., 2020). However, most of the works paid attention to managing the "post stage" of educational activities such as recording and sharing certificates, diplomas, and grades between institutions to protect the "results" of learning and assessment processes. As illustrated in Table 1, although recent research has proposed frameworks and algorithms for secure certificate verification or grade management, there is still limited research exploring effective algorithms for security and privacy "during" learning, not to mention the consideration of pedagogical features (e.g., collaborative learning or scoring bias). Additionally, implementation of the system and its performance evaluation are still lacking.

Table 1. Research of blockchain in education

Educational applications	Research	Features
Research results and data management	Document management (Das et al., 2021)	Use smart contracts to track, manage, and store documents to facilitate approval flows and apply public-key cryptography to facilitate data confidentiality and integrity
Certificate validation and management	Certification for e-learning (Li et al., 2019)	Store e-learning data in a Merkle tree and manage credits using a public blockchain
	Higher education credit management (Turkanović et al., 2018)	Use the DPoS consensus protocol to achieve globally unified viewpoint for students and higher education institutions
	Storing and managing degree information (Nazare et al., 2016)	Store certificate data in a Merkle tree while preserving the ability for individual users to access their own certificates
Assessment	Automate assessment for e-learning (Lam & Dongol, 2020)	Use smart contract to send test files for automated marking and grade calculation and storage
	Grade management (Sudaryono et al., 2020)	Prevent modification of grades by recording all processes in the blockchain
	Grades storage and calculation for e-learning (Li et al., 2019)	Manage grades by allocating e-learning voucher to ensure the credibility based on a private blockchain

2.3. Ethereum

Ethereum is one of the blockchain architectures introduced between 2013 and 2014, devoting to establish a global and most completed blockchain system. Ethereum is very popular and considered to be a huge breakthrough in blockchain technology. One of the important contributions of Ethereum is its introduction of Smart Contract, a computerized transaction protocol that executes the terms of a contract and is written by a specific programming language, such as Solidity (Dannen, 2017). Smart Contract can be independently and autonomously executed by nodes on an Ethereum network using virtual machines, which are called Ethereum Virtual Machines (EVMs). The Turing-completeness of Smart Contract allows Ethereum blockchain to be applied to many complex tasks, such as funding, supply chaining, bidding, and even signing another contract. These features transformed blockchain technology from a purely distributed system that can only send transactions (Jansen et al., 2019) to a completed decentralized architecture that can perform complex tasks and transfer virtual currencies. Ethereum is also open sourced so that everyone can join and research on it, or build his or her own designed private Ethereum-based chains. Therefore, if one wants to design a blockchain system to fulfill some complex use cases using Smart Contract, Ethereum is one of the best platforms. Considering its features, our work is realized based on the Ethereum architecture.

3. Scope and assumptions

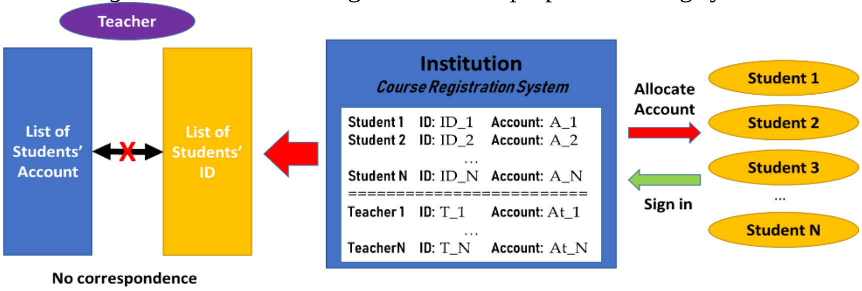
3.1. Regulations and account management

This study aims at developing the required system for a practical educational scenario. Therefore, the proposed assignment scoring mechanism is expected to operate properly under the supervision of an educational institution or an online learning system, where regulations are made to restrict both students and teachers from sabotaging the system. This may seem to centralize the system; however, the operations of the system are designed not to be interfered by the administrator. This means the system is decentralized running by the students and teachers who follow the regulations under the administrator's supervision. Furthermore, the administrator has to verify the status of students and teachers after they signed in the system and intervenes between students and teachers only when some disputes against the preset rules occurred.

In each quarter or semester, every qualified member, such as teacher, student, and teaching assistance, will respectively receive an address that points to the corresponding account used in the assignment/scoring system, from the administrator. After registration, the administrator gives teachers their student lists that contain student accounts associated with the corresponding classes (to prevent non-registered students from joining the courses without permission) and the students' IDs to identify that the students did take the classes at the end of the quarter/semester (see Section 4.4 for details). Note that the correspondence between accounts and students' ID remains in secret (see Figure 1). That is, the teacher will never know which student owns a specific account until the course is finished. The administrator uses the accounts to track and supervise members' behaviors to enhance the stability and liveliness of the system. Offenders are suspended or punished according to the regulations or even laws depending on the severity of violation.

To make the system highly reliable and functional, supervisions and regulations are necessary. However, the system will still operate in a decentralized manner due to the nature of blockchain. Once the system starts, it will be maintained and verified by every on-chain member and its operation will be almost impossible to interfere with or temper the data stored on it, not even by the administrator.

Figure 1. Account management for the proposed scoring system


Teacher
List of Students' Account
X
List of Students' ID
No correspondence
Institution
Course Registration System
Student 1 ID: ID_1 Account: A_1
Student 2 ID: ID_2 Account: A_2
...
Student N ID: ID_N Account: A_N
=========================
Teacher 1 ID: T_1 Account: At_1
...
TeacherN ID: T_N Account: At_N
Allocate Account
Sign in
Student 1
Student 2
Student 3
...
Student N

3.2. Computational power

In the purposed work, the computational power is assumed to be uniformly distributed among all involved members. That is, each member joins the consensus mechanism and has an equal chance and responsibility to create a new block and maintain the liveness of the system. Even though some members do have better computational power than others, it is assumed that no one member will gather enough computational power to conspire against or even sabotage the system.

4. Blockchain-based scoring mechanism

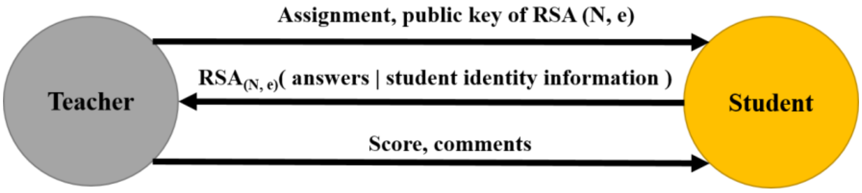
4.1. Basic member interaction models

In every scenario of education, an interaction between students and teachers is a must. Our simplest model aims at simulating the interaction between students and teachers through a blockchain architecture. To do so, the following three functional modules must be defined: assignment delivery and submission, scoring results delivery, and class information announcement. A teacher can deliver assignments or announce information to students by simply sending transactions with messages. Following the same principle, students can submit their assignment answers. However, it does not make sense to put all messages (e.g., answers to assignments) directly on a transaction because a blockchain is a transparent system, which means every on-chain member can see the content of any validated transaction. In short, submitting assignment answers in its plaintext form would result in exposing students' answers to everyone. Therefore, messages that are not suitable to be publicized must be encrypted before sending. There are various ways of encrypting messages to transmitted securely. In this paper, the Rivest–Shamir–Adleman (RSA) algorithm (Calderbank, 2007; Rivest et al., 1978), one of the most widely used encryption methods that is easy to implement and very hard to be cracked, is applied. To use this encryption algorithm (Figure 2), the teacher needs to generate a key pair (a public key and a private key) and sends the public key to the students along with the assignment.

Students then use the RSA algorithm with the shared public key to encrypt their answers and send the ciphertexts to the teacher. The messages include answers for the assignment and student's identity (Section 4.4) so that ciphertexts look different even if the answers are the same. Such design can be effective to prevent plagiarism.

Finally, the teacher can restore the students' assignment answers by decrypting the ciphertexts with the private key.

Figure 2. Interactions between teachers and students

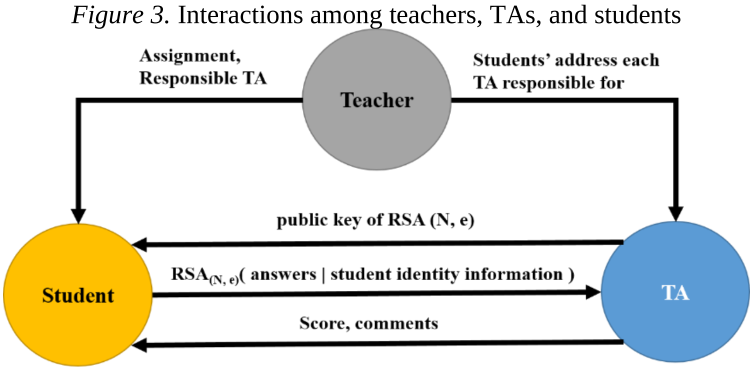

Assignment, public key of RSA (N, e)
Teacher
RSA$_{(N, e)}$(answers | student identity information)
Student
Score, comments

By this approach, our system not only keeps the messages in secret but also prevents answer tamping. All students and teachers put their trust in this model, every system player has fair rights and legal duties to maintain and interact with the model. All system activities are easy to be tracked and supervised; therefore, the deletion or modification of any content of the announcement, assignment, and assignment submissions is nearly impossible. Consequently, the proposed model can build a secure and fair online course platform by using blockchain.

4.2. The role of teaching assistants

Teaching assistants (TAs) are often recruited to help run large courses and distribute assignments. One of the most common tasks that a TA is required to do is to grade assignments and tests. Thus, our model is extended to take the interactions among students, TAs, and teachers into account. As shown in Figure 3, the bottom half of the new model is similar to Figure 2. The only difference is that grading submitted tasks is now done by TAs. In this case, teacher still needs to deliver assignments to students and assign a TA to each student and gives a proof to evidence that the assignment is indeed released by the teacher. After receiving the verified assignment, TAs and students interact with each other accordingly.

Figure 3. Interactions among teachers, TAs, and students

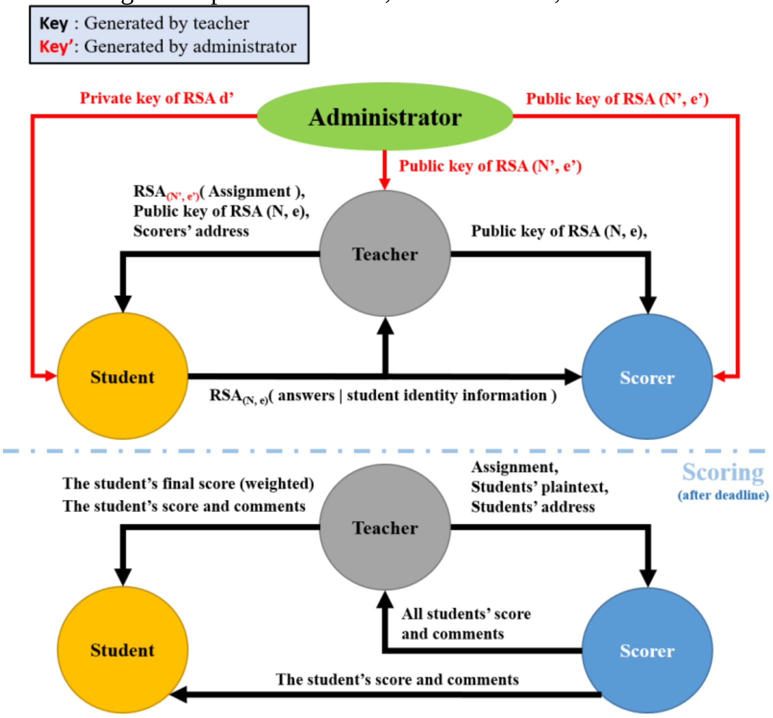


4.3. Collaborative scoring

In real online learning, teachers sometimes design open-ended questions (e.g., essays) to assess students' understanding. It is challenging to assess this type of questions. On the one hand grading using one scorer could possibly cause a bias. On the other hand, if more scorers are included, there might be disagreements among scorers with various perspectives. Therefore, in many high-stake examinations, more than one scorer is involved in scoring to avoid biases. In this study, we also design a secure scoring mechanism for collaborative scoring.

In our collaborative scoring system (shown in Figure 4), students, the responsible teacher, and multiple scorers are invited in the assessment process. The assignments still need to be given by the responsible teacher to prove its legality. Students have to submit their answers in ciphertexts to all scorers. After the deadline, the teacher shares the assignments, students' answers, and sometimes the teacher's remarks (this is optional), to other scorers, while scorers use the ciphertexts and the public key received from the teacher to verify the plaintexts to ensure the plaintexts have not been tempered. Finally, scorers will send their scores to the teacher and corresponding students, so that the final scores can be calculated based on a preset weighting. The teacher sends two scores to each student, one is the teacher's score as a judgement and the other is the final (collected and weighted) score sent to be recorded and verified. Once again, because messages are trackable in blockchain, forging scores become very difficult.

Figure 4. Interactions among the responsible teacher, invited scorers, and students in collaborative scoring


Key : Generated by teacher
Key': Generated by administrator
Private key of RSA d'
Administrator
Public key of RSA (N', e')
Public key of RSA (N', e')
RSA(N', e')(Assignment),
Public key of RSA (N, e),
Scorers' address
Teacher
Public key of RSA (N, e),
Student
Scorer
RSA(N, e)(answers | student identity information)
Scoring
(after deadline)
The student's final score (weighted)
The student's score and comments
Teacher
Assignment,
Students' plaintext,
Students' address
Student
All students' score
and comments
Scorer
The student's score and comments

Another issue is regarding the timing of receiving students' submissions by the scorers. To prevent the scorers from discussing submissions with others, which may introduce the scoring bias, the assignments should be kept secure during the submission stage. However, data on blockchain are transparent to all on-chain members. Therefore, an RSA public-private key-pair can be employed to make the assignments secure from scorers at the submission stage. But the process might be inefficient: all students generate their own key-pairs and the teacher encrypts the assignment for each student with the student's key individually. To improve the usability and simplify the process for both teacher and students, we chose to have a system administrator generate the key-pairs and distribute the public keys to the teacher and the private keys to the students. The teacher then can send encrypted assignments and only the corresponding students can decrypt. When the scoring stage starts, the scorers receive the plaintext of the assignment and students' answers for scoring. The scorers then can use the public key to verify and ensure that the answers are not tempered.

By this approach, the scores for opened questions will be more reliable. Additionally, the scorers in this model are anonymous, so that each scorer can judge the quality of answers without being affected by others (e.g., the owner of the answers). As a result, the proposed approach can improve quality and fairness. Specifically, for extremely high-stake assessments such as examinations that are directly affecting the issuing of certificates or the college entrance qualifications, the abovementioned method is believed to be a trustworthy way for establishing credibility in scoring.

4.4. Authentication

On a blockchain, every member is identified by a hash string and their activities are thus anonymized. In other words, this property allows students to take courses without giving up their identity. It also ensures that teachers treating their students equally. However, the teacher needs to recover students' IDs to give final scores. To achieve this, a Shamir's Secret Sharing algorithm (Shamir, 1979) and Chaotic Cryptography algorithm (Kocarev & Lian, 2011) based authentication scheme are used and are discussed in the rest of this section.

In the beginning of the course, by using Chaotic Cryptography, each student generates his/her secret codes by encrypting his/her student ID with the chosen password and segments his/her secret codes into secret pieces by using the Shamir's Secret Sharing algorithm (Figure 5). Students then send each one of their secret pieces together with their submitted assignments to the teacher so that the teacher can eventually find their secret codes out (Figure 6). The *t*-out-of-*N*, (*N*,*t*)-Threshold Shamir's Secret Sharing algorithm is adopted. This algorithm initially segments the secret into *N* pieces, and the secret can later be recovered if at least *t* out of the *N* pieces are retrieved. In our system, the parameter *N* is set to the total number of assignments in a course, and *t* is the least number of assignments that a student has to submit, which is determined by the teacher.

Figure 5. Student generates secret pieces

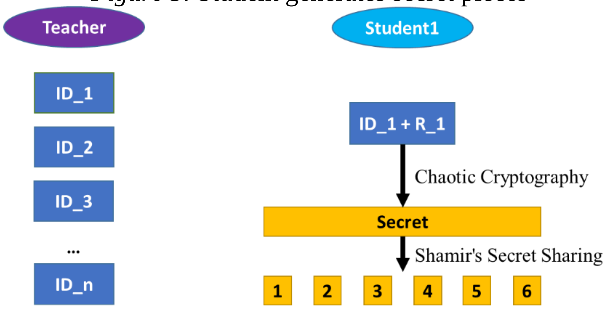


Figure 6. Teacher recovers a student's secret code by retrieving secret pieces sent together with the submitted assignment

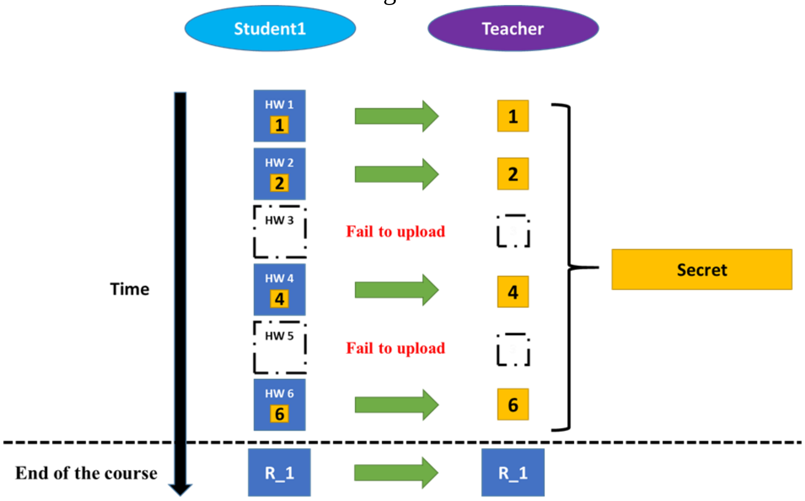
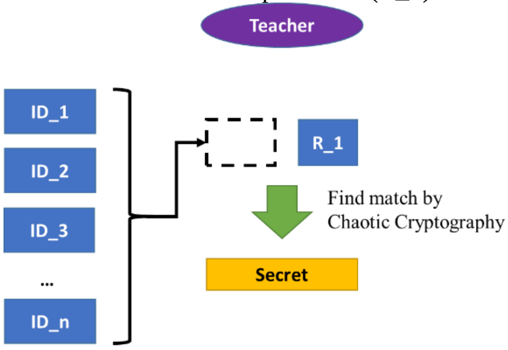


Figure 7. Teacher identifies a student with the password (R_1) and the associated secret code



In considering the fact that it is not realistic to expect every student to submit every assignment on time due to unpredicted reasons such as missing the deadline or cannot complete the assignment, the teacher should still be able to recover the secret codes with only a part of secret pieces. In some cases, students are not asked to turn in all assignments but at least a minimum number of assignments. If a student fails to fulfill the minimum requirement, the teacher won't be able to recover the secret codes to identify the student, and thus no final score will be given to the student. In contrast, if a student can prove the efforts that they put to the course, the final score should still be given even some of the submissions are missing.

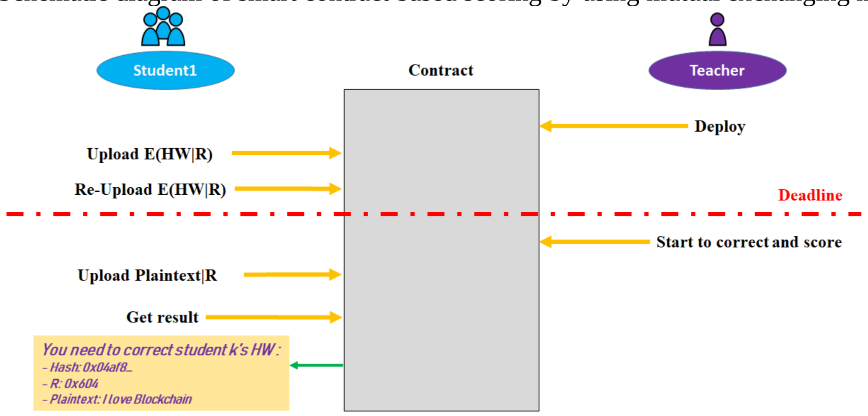
Integrating the secret sharing algorithm with the system makes it closer to the needs of real application scenarios. However, the adopted (N,t)-Threshold scheme also imply that the teacher can obtain students' identity before the course is completed. To solve this problem, Chaotic Cryptography is applied to protect students' privacy. A teacher can never find out the student ID within the secret codes without knowing the password set by the student. That is, a student's identity will remain in secret before the student sent out the final key information, i.e., the password, to the teacher, at the end of the course (Figure 7). Additionally, because there is no student ID shared on the blockchain, a student's identity behind a given account is safe and remains unknown to the other members. By combining the above schemes, a teacher can identify the students enrolled in the course and set some rules (such as the values of N and t) for the course while the students' privacy is well protected by their privately set passwords.

5. Smart contracts

Smart contract is a crucial feature of blockchain that uses designed protocol to autonomously run on decentralized networked nodes for achieving various complex tasks. Once deployed, a smart contract acts as a fair and transparent arbiter to deal with every request from its users. In education, there are many complex situations that a smart contract can be applied to make things easier. For example, it can be used to collect group lists or act as a billboard to announce information. It is worth mentioning that a well-designed smart contract can also replace TAs for completing tasks that follow unambiguous rules, such as scoring assignments. Smart contracts guarantee tasks can be done objectively comparing with TAs who may have specific personal opinions on certain students. To ensure the fairness and transparency of scoring, three approaches to score assignments or examinations by a smart contract are proposed: *peer evaluation*, *automatic scoring by a smart contract*, and *collaborative scoring*.

As mentioned in Section 4.1, sending answers in plaintext equals sharing answers to everyone on the blockchain, which is certainly not ideal. However, using a smart contract to decrypt a ciphertext is very difficult and costly due to the complexity of crypto algorithms. What is worse is uploading a private key to the blockchain not only has to pay the cost for storing large random numbers but also reveal the private key to all on-chain members. Practically, it is not trivial to avoid mistakes when embedding a huge-size message into a transaction. Therefore, a better way to protect information security is to use the smart contract to directly verify the ciphertext with the aid of various commitment schemes instead of decrypting it back to plaintext and then score. As illustrated in Figure 8, when an assignment is announced, students need to upload their answers in a ciphertext form before the deadline. The secure hash algorithm used to obtain the ciphertext should also be supported by the smart contract to truly optimize the efficiency. In our work, the keccak256 hash algorithm, which is a callable function to Solidity language, is adopted.

Figure 8. Schematic diagram of smart contract based scoring by using mutual exchanging mechanism



Similar to the basic model, the plaintext should contain an extra message, which is denoted as R in Figure 8, to prove the student's identity at the end of the course and prevent answer-tamping or assignment-copying flaws to ensure every student will get a unique hash value even if their answers are the same. After the deadline, students upload their answers together with message R. The integrity of answers can be proved by the smart contract via checking if the hash of the plaintext matches the uploaded ciphertext. Once it is confirmed the answers in plaintext can be scored manually or automatically by the smart contract.

5.1. A smart contract to support peer evaluation

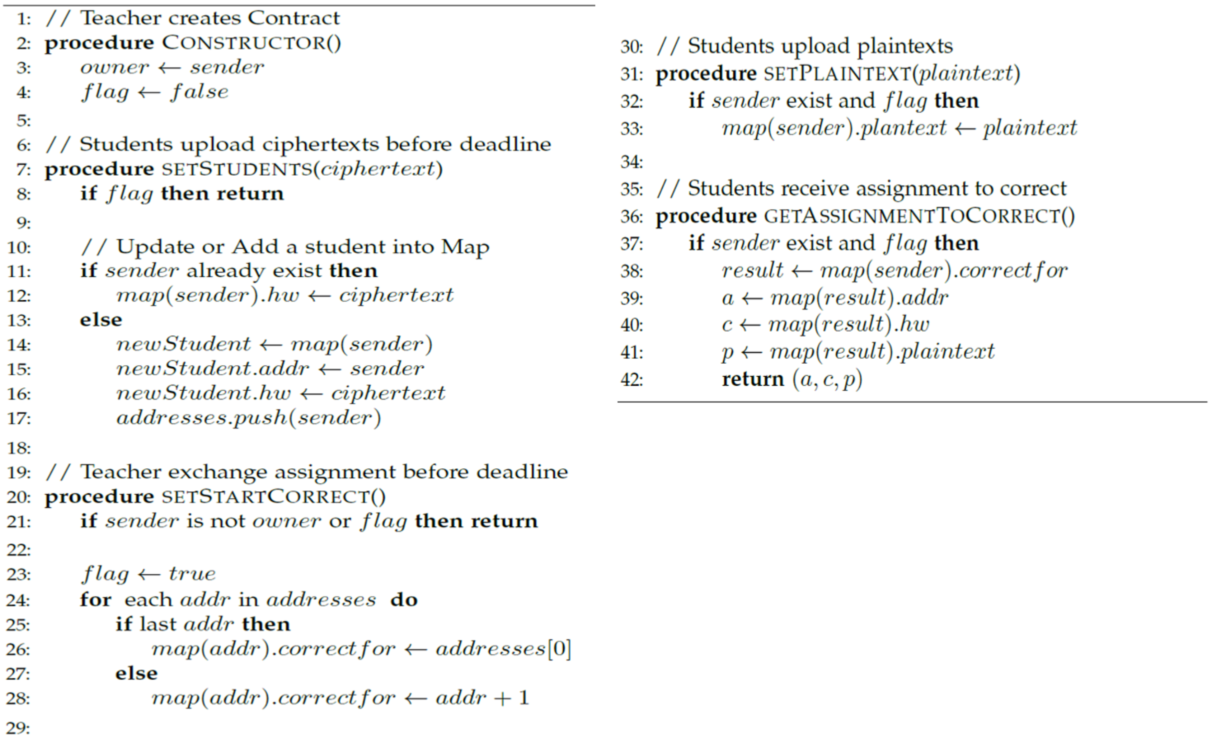
Peer evaluation is to let students score other student's assignments. On a blockchain, no student is able to know the owner of other addresses. That is, a student does not know whose assignment he or she is grading, and therefore, will reduce the chance of cheating. In addition, the smart contract is designed to make the assignment of peer evaluation randomly. Even if a student shared his or her address with friends, there is no guarantee that they will be paired, especially when there are many students enrolled in the course.

The proposed smart contract requires four basic functions for peer evaluation to work: (1) submit ciphertext, (2) start scoring, (3) submit plaintext, and (4) fetch the assignment that needs to be corrected/scored. As shown in Table 2, initially, only function (1) is activated for students to submit their ciphertext (with commitment) while functions (3) and (4) remain disabled until the teacher calls function (2) and uploads the solutions or the rubrics after the assignment deadline. At this point, function (1) is also disabled to prevent students from submitting new answers. The algorithm of scoring by exchanging is illustrated in Figure 9.

Table 2. Status of all functions when function (2) is called

Functions	(1) Submit ciphertext	(2) Start scoring	(3) Submit plaintext	(4) Fetch assignment
After (2) is called	Disable	Disable	Enable	Enable

Figure 9. Algorithm: Scoring by mutual exchanging

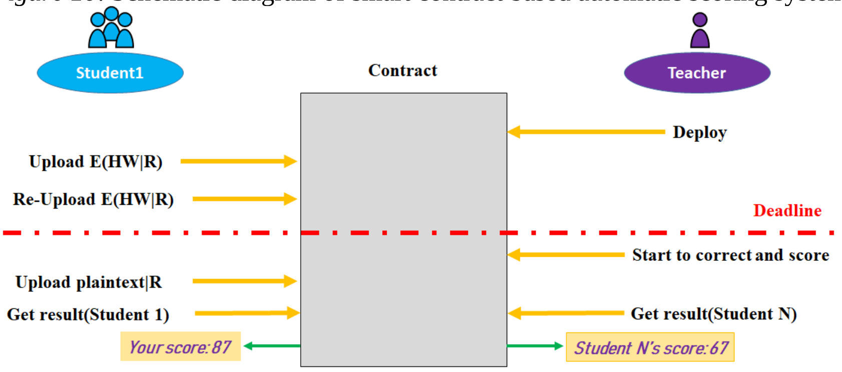


```
// Teacher creates Contract
procedure CONSTRUCTOR()
    $owner \leftarrow sender$
    $flag \leftarrow false$

// Students upload ciphertexts before deadline
procedure SETSTUDENTS($ciphertext$)
    if $flag$ then return

    // Update or Add a student into Map
    if $sender$ already exist then
        $map(sender).hw \leftarrow ciphertext$
    else
        $newStudent \leftarrow map(sender)$
        $newStudent.addr \leftarrow sender$
        $newStudent.hw \leftarrow ciphertext$
        $addresses.push(sender)$

// Teacher exchange assignment before deadline
procedure SETSTARTCORRECT()
    if $sender$ is not $owner$ or $flag$ then return

    $flag \leftarrow true$
    for each $addr$ in $addresses$ do
        if last $addr$ then
            $map(addr).correctfor \leftarrow addresses[0]$
        else
            $map(addr).correctfor \leftarrow addr + 1$


// Students upload plaintexts
procedure SETPLAINTEXT($plaintext$)
    if $sender$ exist and $flag$ then
        $map(sender).plantext \leftarrow plaintext$

// Students receive assignment to correct
procedure GETASSIGNMENTTOCORRECT()
    if $sender$ exist and $flag$ then
        $result \leftarrow map(sender).correctfor$
        $a \leftarrow map(result).addr$
        $c \leftarrow map(result).hw$
        $p \leftarrow map(result).plaintext$
        return $(a, c, p)$
```

5.2. A smart contract to support automatic scoring

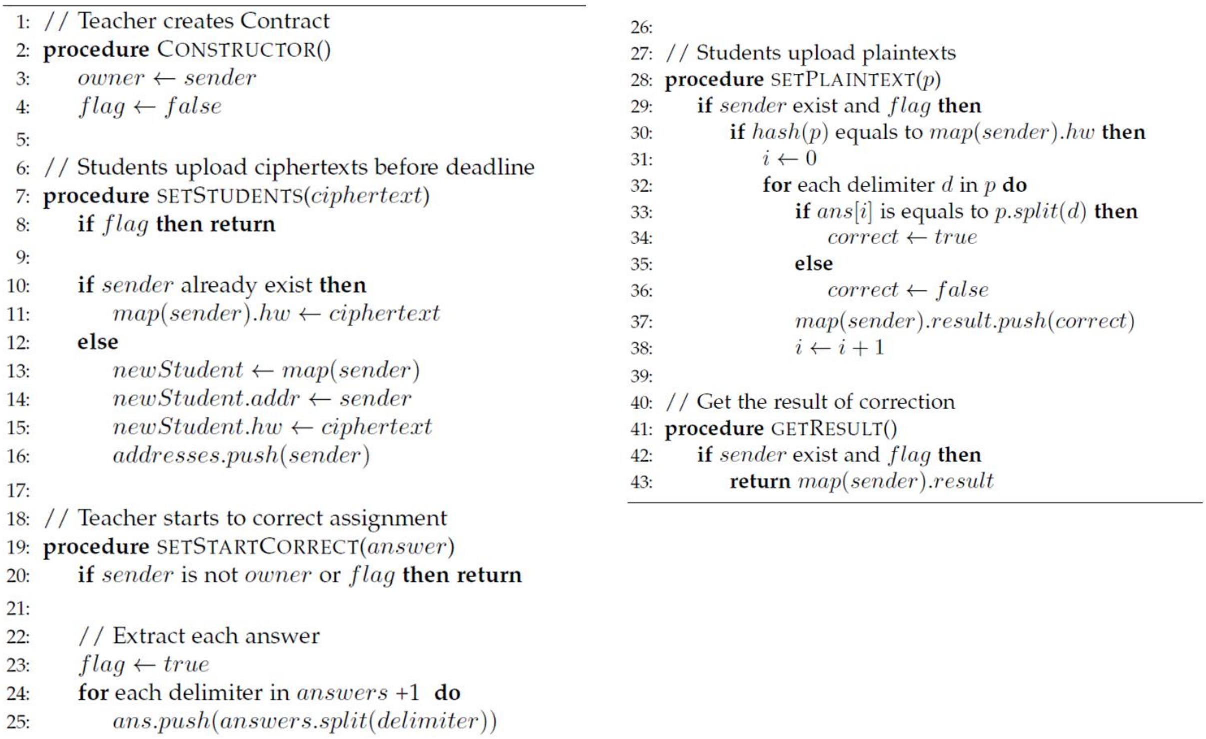
Another way to make scoring fairly to every student is to let the smart contract grades the submissions, as illustrated in Figure 10. This kind of smart contracts also have four basic functions: (1) submit ciphertext, (2) start scoring, (3) submit plaintext, and (4) fetch scoring results. When calling function (2), the teacher receives the actual answers and changes the status of the other three functions as shown in Table 2. The smart contract then scores the answer once the student calls function (3) and uploads the plaintext that matches the verified ciphertext uploaded by function (1). Finally, function (4) allows all enrolled students to see the results of their assignments.

Figure 10. Schematic diagram of smart contract based automatic scoring system



Comparing with the smart contract proposed in Section 5.1, this approach simplifies students’ work loads and guarantees fairness to all students because the smart contract autonomously grades every submission. However, to make this method feasible, both the solutions and their forms in the plaintext domain must be fixed to make sure that the smart contract can match or extract correct solutions from the plaintexts. For this reason, the teacher must upload answer keys rather than guidelines or rubrics. Thus, scoring essay is hard to achieve using this approach. The automatic scoring algorithm is illustrated in Figure 11.

Figure 11. Algorithm: Automatic scoring



```
// Teacher creates Contract
procedure CONSTRUCTOR()
    $owner \leftarrow sender$
    $flag \leftarrow false$

// Students upload ciphertexts before deadline
procedure SETSTUDENTS($ciphertext$)
    if $flag$ then return

    if $sender$ already exist then
        $map(sender).hw \leftarrow ciphertext$
    else
        $newStudent \leftarrow map(sender)$
        $newStudent.addr \leftarrow sender$
        $newStudent.hw \leftarrow ciphertext$
        $addresses.push(sender)$

// Teacher starts to correct assignment
procedure SETSTARTCORRECT($answer$)
    if $sender$ is not $owner$ or $flag$ then return

    // Extract each answer
    $flag \leftarrow true$
    for each delimiter in $answers$ +1 do
        $ans.push(answers.split(delimiter))$

// Students upload plaintexts
procedure SETPLAINTEXT($p$)
    if $sender$ exist and $flag$ then
        if $hash(p)$ equals to $map(sender).hw$ then
            $i \leftarrow 0$
            for each delimiter $d$ in $p$ do
                if $ans[i]$ is equals to $p.split(d)$ then
                    $correct \leftarrow true$
                else
                    $correct \leftarrow false$
                $map(sender).result.push(correct)$
                $i \leftarrow i + 1$

// Get the result of correction
procedure GETRESULT()
    if $sender$ exist and $flag$ then
        return $map(sender).result$
```

5.3. A smart contract to support collaborative scoring

To make scoring of open-ended questions more convincing by allowing the answers be judged by different scorers, a smart contract is designated to implement a collaborative scoring framework. This smart contract consists of seven basic functions (Figure 12): (1) submit ciphertext, (2) start grading, (3) submit plaintext, (4) register scorer, (5) get assignment, (6) score, and (7) get result. Once again, students need to upload their ciphertexts using function (1) as the commitments, submit the assignment answer plaintexts by calling function (3), and after the deadline or after the teacher starts the correction/scoring process applying function (2). Notice that function (2) plays only the role of locking and unlocking functions, as listed in Table 3, without asking for standard procedures of scoring to ensure scorers following their own opinions. Function (4) allows the teacher (contract owner) to add scorers to the smart contract at any moment and the scorers can then apply function (5) to see student’s information (ciphertext, plaintext and address) they need for scoring the submissions. The scorers can upload their scoring results using function (6), whereas the students and the teacher can find the scoring results using function (7). The scoring results include each score given to the student by different scorers and a weighted final score.

This smart contract design provides an efficient method for collaborative scoring. It helps manage tasks and integrate information into one simple platform while still ensures all scorers’ and students’ anonymity so that each judgement can be made without interference by other factors. With this smart contract, scores are given trustworthily and faithfully so that the final scores can reflect the true learning outcome and thus the certificate of the course or the achievements accomplished in the course can be more convincing. The algorithm is shown in Figure 13.

Figure 12. Schematic diagram of the proposed smart contract based autonomous collaborative scoring system

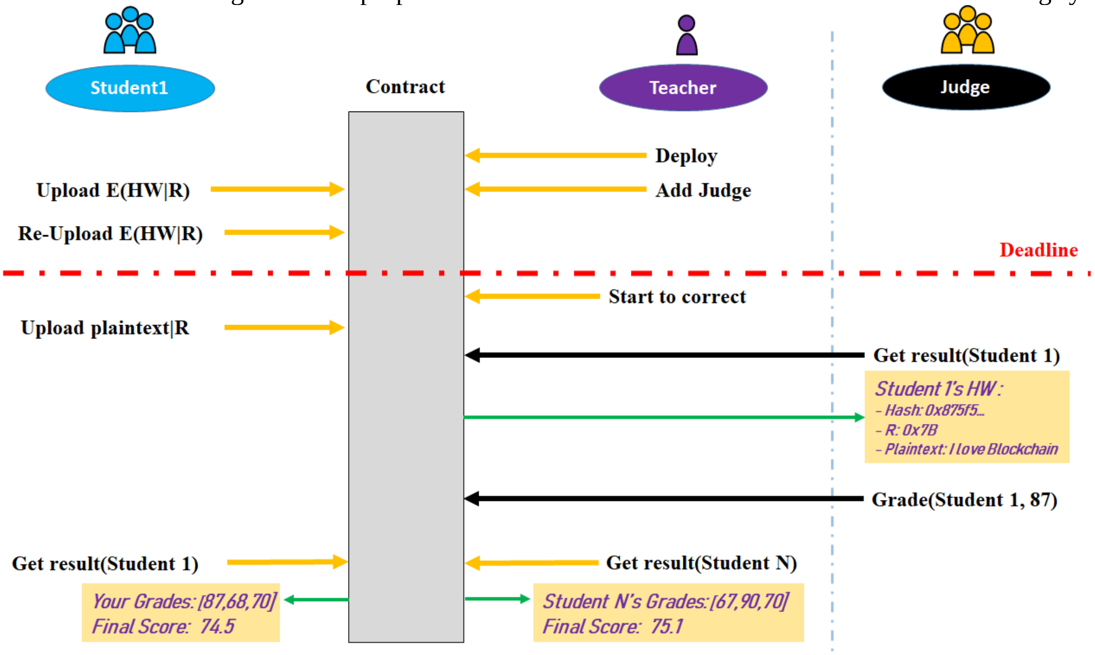
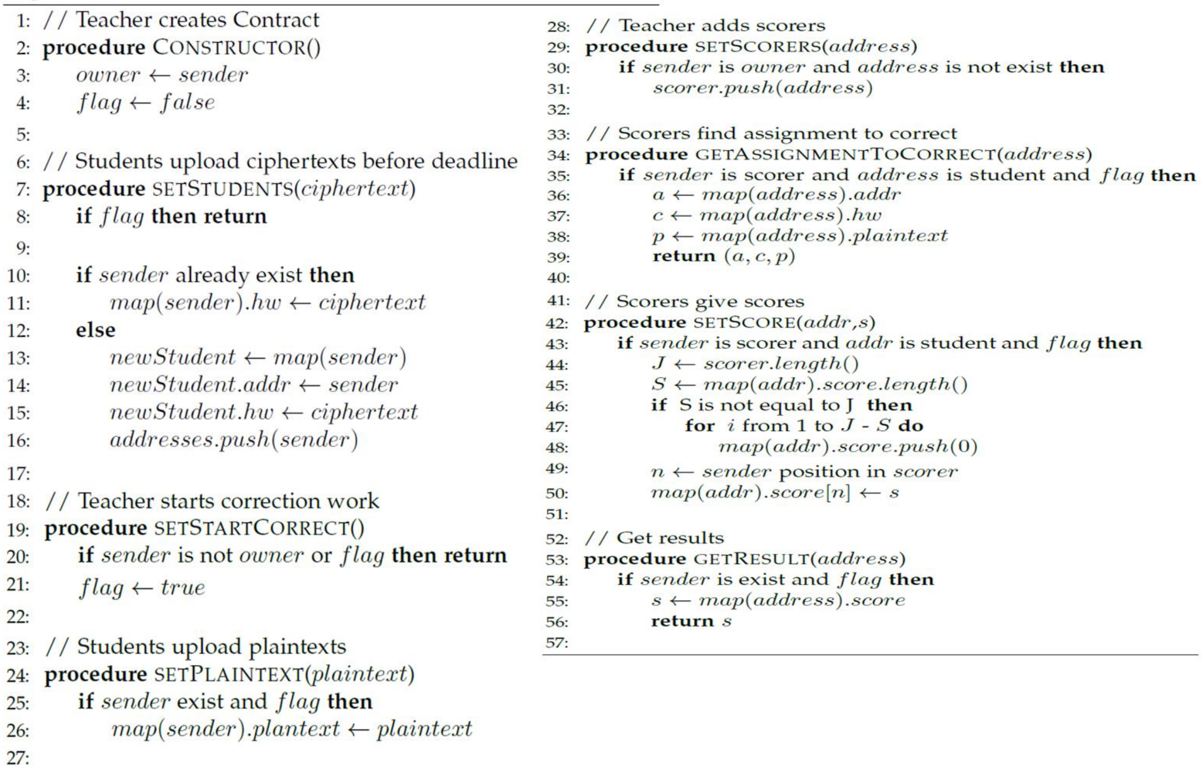


Table 3. Statuses of all functions when function (2) is called

Functions	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Deployment	En	En	Dis	En	Dis	Dis	Dis
After (2) is called	Dis	Dis	En	En	En	En	En
Note	(1): Submit ciphertext (2): Start scoring (3): Submit plaintext (4): Register scorer				(5): Get assignment (6): Score (7): Get result		

Note. En: Enabled, Dis: Disabled.

Figure 13. Algorithm: Collaborative scoring



```
// Teacher creates Contract
procedure CONSTRUCTOR()
    $owner \leftarrow sender$
    $flag \leftarrow false$

// Students upload ciphertexts before deadline
procedure SETSTUDENTS($ciphertext$)
    if $flag$ then return

    if $sender$ already exist then
        $map(sender).hw \leftarrow ciphertext$
    else
        $newStudent \leftarrow map(sender)$
        $newStudent.addr \leftarrow sender$
        $newStudent.hw \leftarrow ciphertext$
        $addresses.push(sender)$

// Teacher starts correction work
procedure SETSTARTCORRECT()
    if $sender$ is not $owner$ or $flag$ then return
    $flag \leftarrow true$

// Students upload plaintexts
procedure SETPLAINTEXT($plaintext$)
    if $sender$ exist and $flag$ then
        $map(sender).plantext \leftarrow plaintext$

// Teacher adds scorers
procedure SETSCORERS($address$)
    if $sender$ is $owner$ and $address$ is not exist then
        $scorer.push(address)$

// Scorers find assignment to correct
procedure GETASSIGNMENTTOCORRECT($address$)
    if $sender$ is scorer and $address$ is student and $flag$ then
        $a \leftarrow map(address).addr$
        $c \leftarrow map(address).hw$
        $p \leftarrow map(address).plaintext$
        return $(a, c, p)$

// Scorers give scores
procedure SETSCORE($addr$,$s$)
    if $sender$ is scorer and $addr$ is student and $flag$ then
        $J \leftarrow scorer.length()$
        $S \leftarrow map(addr).score.length()$
        if S is not equal to J then
            for $i$ from 1 to $J$ - $S$ do
                $map(addr).score.push(0)$
        $n \leftarrow sender$ position in $scorer$
        $map(addr).score[n] \leftarrow s$

// Get results
procedure GETRESULT($address$)
    if $sender$ is exist and $flag$ then
        $s \leftarrow map(address).score$
        return $s$
```

6. Experiment

6.1. Implementation

The purposed work is realized on the Ethereum blockchain network with designed application tools to integrate all the mechanisms introduced in Section 4 and 5. For simplicity and re-producibility, the proposed blockchain system is built based on the Ethereum source code (https://github.com/ethereum/goethereum), programmed in Go language. The application tools are important keys to make the realized assignment scoring system much more user friendly. They cover all complex procedures for the users (students and teachers) so that everyone can use the system with ease by few simple selections without the need to understand the principles and theories of blockchain beforehand, which is in fact a desired scenario in real usage.

The application tools include three main modules: the cryptography module, the blockchain module, and the student identity module. Two versions of the application tools are created: one for the students and the other for the teachers. To summarize the tools, we used Nodejs for blockchain interactions, Go for Chaotic Cryptography (Amigo et al., 2007) and Python for user-interface, RSA (Shand & Vuillemin, 1993) and Secret Sharing (Shamir, 1979) for encryption/decryption. The procedures and the user interface of the application are illustrated in Figure 14 and Figure 15, respectively.

Figure 14. Procedures of the Teacher-Student Interaction by the applications

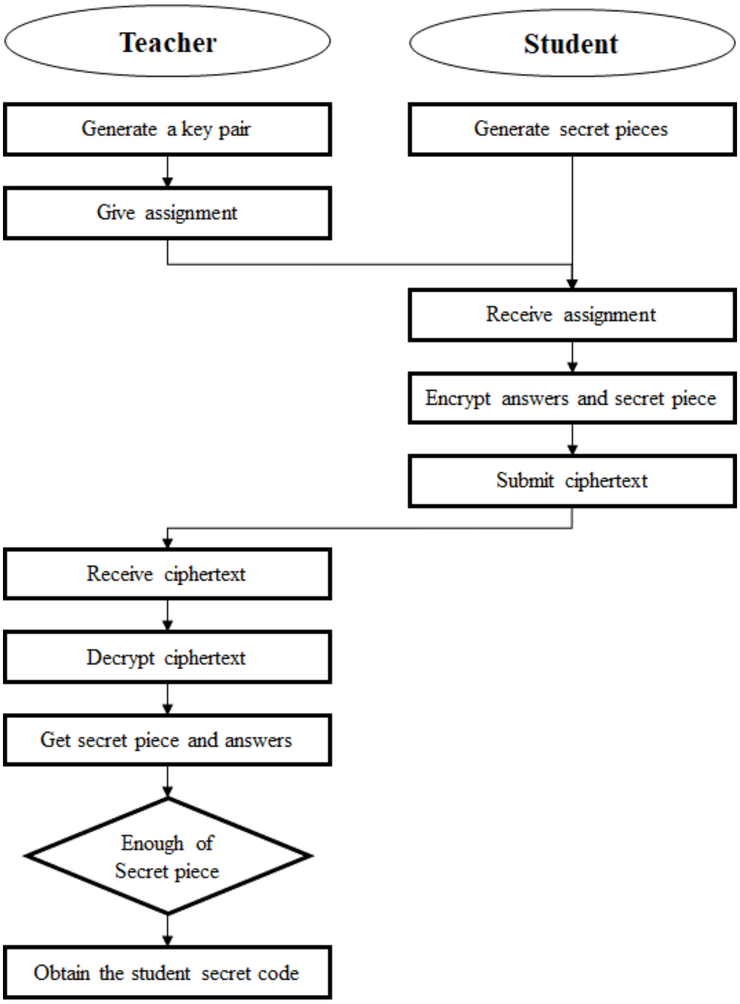
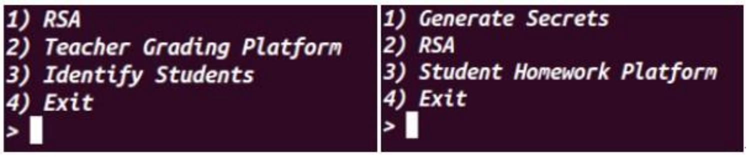


Figure 15. The application user interface for teachers (left) and students (right)



```
1) RSA
2) Teacher Grading Platform
3) Identify Students
4) Exit
>
```

```
1) Generate Secrets
2) RSA
3) Student Homework Platform
4) Exit
>
```

6.2. Performance test: randomness of the chaotic random number

In our work, a chaotic map based random number generating module is used to hash the inputted plaintexts for protecting the students’ anonymity. Thus, the security of the Chaotic Cryptography module is directly correlated to the randomness of the generated random numbers. The experiment results show the high randomness of the generated random numbers by observing their distributions and comparing the randomness between two generated results with two seeds differed in a tiny difference.

Figure 16. The noisy image generated with the seed value 12345678

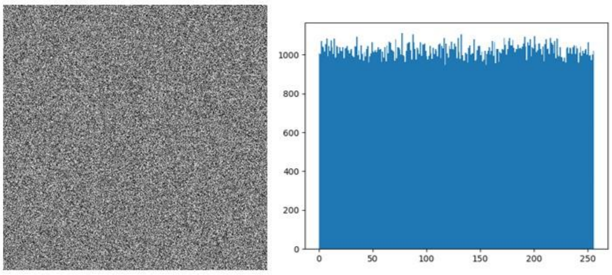


Figure 17. The noisy image generated with the seed value 12345677

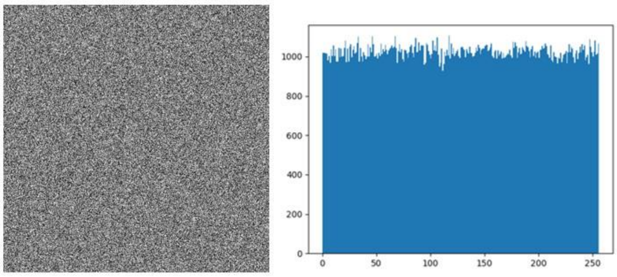
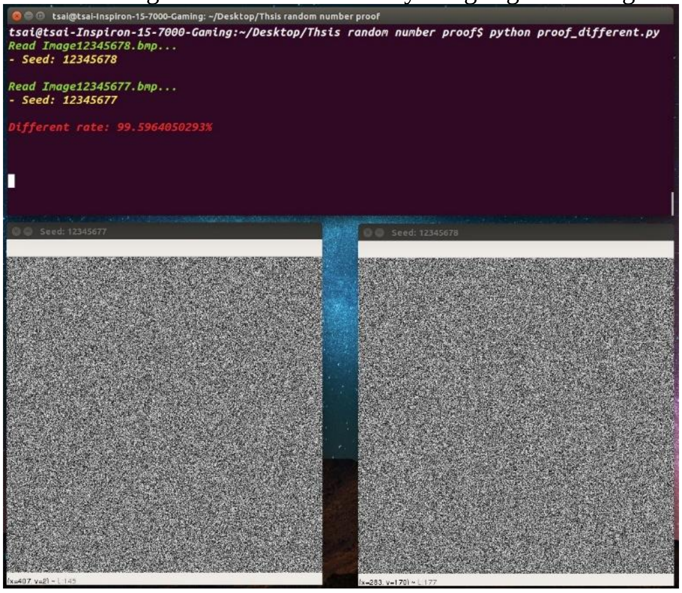


Figure 18. The difference image between the two noisy images given in Figure 14 and Figure 15


tsai@tsai-Inspiron-15-7000-Gaming: ~/Desktop/Thsis random number proof
Seed: 12345677
Seed: 12345678
(x=407, y=2) ~ L:145
(x=283, y=170) ~ L:177

```
tsai@tsai-Inspiron-15-7000-Gaming:~/Desktop/Thsis random number proof$ python proof_different.py
Read Image12345678.bmp...
- Seed: 12345678

Read Image12345677.bmp...
- Seed: 12345677

Different rate: 99.5964050293%
```

Figure 16 (left) shows the 512×512 noise image corresponding to the generated random numbers, with the given seed 12345678 and the associated histogram (Figure 16 (right)) verifies that their corresponding distribution is very close to the uniform one. Figure 17 (left) shows another 512×512 noise image which is created with the seed value of 12345677 and as shown in Figure 17 (right) the resultant histogram is once again very close to uniform distribution. Although there is only a single digit difference between the two seeds, the comparison given in Figure 18 shows that there is a huge amount of pixel changes (99.6%) between the two images. Therefore, the outcomes of the adopted chaotic random number generating module are highly unpredictable and will bring large benefits to students’ privacy and security.

6.3. Comparison with related work

Comparing our work with a popular centralized online course platform (https://ceiba.ntu.edu.tw/) called CEIBA used at first author's university, our system has advantages in system transparency and fairness to students (as shown in Table 4). In the centralized platform, information can be uploaded or deleted without being recorded, that is, those actions cannot be tracked by involved members (mostly students). Thus, students or teachers could miss some deleted information and result in dissensions. Additionally, if students use their true identities to interact with their teachers, this may result in teachers treating each student unequally due to an implicit stereotype. Therefore, using blockchain properties to openly track every information and activity will make the system much more ***transparent*** to avoid lots of unnecessary disputes between students and teachers by treating every on-chain member ***equally***. Those smart contracts introduced in Section 5 not only help teachers distribute some heavy workloads but also ensure better ***fairness*** to every student. Moreover, the crypto system used in this work can prevent students from cheating.

Moreover, most of previous research focused on managing post stage of educational activities, for example, recording and sharing students' certificates and degrees between colleges to protect the "results" of assessment processes. In contrast, our work realized a design that is aiming at managing information security to ***protect the "procedure" of assessment***, including assignment submissions and scoring, to ensure the ***transparency*** and ***fairness*** of educational assessment.

Table 4. Comparison of the purposed work and the traditional online course platform, CEIBA

	The purposed work	Traditional educational online platform
Decentralized	Decentralized	Centralized
Transparency	Blockchain property	Centralized
Fairness	Smart Contract, Blockchain property	Depend on teachers
Prevent cheating	Cryptography, Blockchain property	-
Speed	Latency due to Encryption/Decryption	-
Data Preservation	Maintained by all members	Centralized
Liveness, Stability	Depend on all members	Depend on administrator

7. Discussion

The design of the proposed system focuses mostly on realization of a transparent and fair assignment scoring platform based on the blockchain technology. Both ***system performance*** (***security and stability***) and ***pedagogical feasibility*** are considered:

Regarding the ***performance of the proposed system***, the ***security*** properties described in this work was tested and proved to be stable (low latency without error) using 20 nodes equipped with a 2.80GHz CPU and 16GB RAM. The ***stability*** of the blockchain seems to be able to reliably handle lots of users because it is Ethereum-based blockchain, which has been used by millions of users. Besides, our system uses the crypto-hash function and the RSA algorithm, whose security depends on the hash function and the key size. National Institute of Standards and Technology (NIST) provided the estimated maximum-security (Barker & Dang, 2016) and the 1024-bit RSA we used can achieve 80-bits security strength which keeps an acceptable trade-off between security and encoding speed. In the future, longer key length could be used when computing power improves to the increase security level. Moreover, a larger scale real test should be performed before the system is ready to be deployed in practical usage. To increase the practical value of our current system, the user interface should be more user-friendly. Some procedures of the proposed system can also be automated to make scoring more effective. For example, automatically decrypting ciphertext once the teacher's account received a certain amount of submissions from students.

Regarding the ***pedagogical feasibility***, the proposed system provides many features that are aligned with the emerging trends in education. The "***peer evaluation***" smart contract relies on the submission order of students' submitted assignments and could be exploited when a group of students conspired to upload their assignments (in ciphertext form) at the same time. Using hash functions to generate random numbers can make the exchanging behavior more unpredictable. But, again, this may result in too high of computational cost when there are too many students enrolled in the course. Besides, the random numbers are predictable by those who decide the seeds of the hash functions (e.g., the responsible teachers) or those who generate the blocks (i.e., the miners). Additionally, blockchain provides a more ***student-centered environment***, students have an easy way to store and manage their portfolio, projects, credits, and degrees, which contributes to self-directed learning. The system also allows educators, universities, and institutes to manage student-related affairs, share their information with other universities, and track students' learning histories and outcome. It can also prevent improper activities, such as cheatings or forgeries, with the aid of blockchain, a decentralized and transparent system where every activity can be verified and supervised by all involved members. With blockchain, a student can apply for the entrances to colleges without printing mass of diplomas or certificates of programs learnt; instead, colleges can find student's information. This will not only save resources and time, but also establish fairness, transparency, and security of information flow.

Although it is expected to establish an efficient way for supervising scoring-related activities and ensuring fairness to all members, the latency caused by the involved encryption processes becomes the major obstacle to its adaption in practice. The most obvious latency is caused by the RSA module, which takes approximately one minute to encrypt a plaintext with just 100 words. Fortunately, this comes from the huge time cost from programming implementation, which can be solved by optimization techniques.

In the future, combining our work with other related works to integrate the merits of blockchain technology into higher level education usage, such as sharing and maintaining students' certificate and learning results between institutions and colleges, is of great interest. Although this goal is currently difficult to achieve since it requires the support of cross-chain techniques. However, the cross-chain system integration might build a complete blockchain-based educational system, from information sharing between institutions, basic interactions between teachers and students, to establish a true transparent and fair educational system for all students, teachers, and administration staffs.

8. Conclusions

This paper presents a design of blockchain-based assignment scoring mechanism for online learning. Our goal is to take advantages of blockchain properties and cryptography algorithms to build a transparent and secure teacher-student interaction system for online assessment. The fairness of scoring can be guaranteed by anonymity of the proposed blockchain architecture and the collaborative scoring policy. Although the online learning system will pay extra computational cost and related administrative procedures need to be made to use the proposed scoring mechanisms, our work is one initial step in designing and developing a feasible scoring mechanism to achieve fairer and more secure assessment for the rapid-growing online learning. The trend of education is moving toward online model. The proposed methodology can contribute to the high-quality assessment for online learning. In the future, empirical studies could be conducted by embedding the proposed mechanism in a real online learning platform such that its effectiveness in real educational applications could be examined. In addition, big data solutions and the architecture design (e.g., using proof-of-stake to reduce computational power and carbon footprint) can also be considered to enhance the feasibility of the proposed scoring mechanism. More advanced algorithms can also be applied to improve the performance of the proposed scoring system. For example, picking another random number generator with higher randomness and efficiency with less computational cost is an important task for smart contract designer to provide more unpredictability, and thus, achieving real fairness for all members.

References

Amigo, J. M., Kocarev, L., & Szczepanski, J. (2007). Theory and practice of chaotic cryptography. *Physics Letters A*, *366*(3), 211-216. http://doi.org/10.1016/j.physleta.2007.02.021

Apampa, K. M., Wills, G., & Argles, D. (2010). User security issues in summative e-assessment security. *International Journal of Digital Society (IJDS)*, *1*(2), 1-13. http://doi.org/10.20533/ijds.2040.2570.2010.0018

Barker, E., & Dang, Q. (2016). *NIST special publication 800-57 part 1, revision 4*. *Recommendation for Key Management Part 1: General.* National Institute of Standards and Technology Special Publication (NIST). http://dx.doi.org/10.6028/NIST.SP.800-57pt1r4

Caballé, S., Miguel, J., Xhafa, F., Capuano, N., & Conesa, J. (2017). Using trustworthy web services for secure e-assessment in collaborative learning grids. *International Journal of Web and Grid Services*, *13*(1), 49-74. http://doi.org/10.1504/IJWGS.2017.082059

Calderbank, M. (2007). *The RSA cryptosystem: History, algorithm, primes*. Math. Uchicago. Edu. https://www.math.uchicago.edu/~may/VIGRE/VIGRE2007/REUPapers/FINALAPP/Calderbank.pdf

Castella-Roca, J., Herrera-Joancomarti, J., & Dorca-Josa, A. (2006). A Secure e-exam management system. In *Proceedings of the First International Conference on Availability, Reliability and Security (ARES'06)*. IEEE. http://doi.org/10.1109/ARES.2006.14

Chen, G., Xu, B., Lu, M., & Chen, N. S. (2018). Exploring blockchain technology and its potential applications for education. *Smart Learning Environments*, *5*(1), 1-10. http://doi.org/10.1186/s40561-017-0050-x

Dannen, C. (2017). *Introducing Ethereum and solidity*. Apress. http://doi.org/10.1007/978-1-4842-2535-6_4

Das, M., Tao, X., & Cheng, J. C. P. (2021). A Secure and distributed construction document management system using Blockchain. In Toledo Santos, E., & Scheer, S. (Eds,), *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering* (pp. 850–862). https://doi.org/10.1007/978-3-030-51295-8_59

Grech, A., & Camilleri, A. (2017). *Blockchain for education*. Publications Office of the European Union. http://doi.org/10.2760/6064910.1007/s11191-017-9891-5

Hernandez-de-Menendez, M., Escobar Díaz, C., & Morales-Menendez, R. (2020). Technologies for the future of learning: State of the art. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, *14*(2), 683-695. http://doi.org/10.1007/s12008-019-00640-0

Hoy, M. B. (2017). An Introduction to the blockchain and its implications for libraries and medicine. *Medical Reference Services Quarterly*, *36*(3), 273-279. http://doi.org/10.1080/02763869.2017.1332261

Jain, N., Agrawal, T., Goyal, P., & Hassija, V. (2019). A Blockchain-based distributed network for secure credit scoring. In *2019 5th International Conference on Signal Processing, Computing and Control (ISPCC)* (pp. 306-312). IEEE. http://doi.org/10.1109/ISPCC48220.2019.8988510

Jansen, M., Hdhili, F., Gouiaa, R., & Qasem, Z. (2019). Do smart contract languages need to be Turing complete? In *International Congress on Blockchain and Applications* (pp. 19-26). Springer, Cham. http://doi.org/10.1007/978-3-030-23813-1_3

Kocarev, L., & Lian, S. (Eds.). (2011). *Chaos-based cryptography: Theory, algorithms and applications* (Vol. 354). Springer Science & Business Media. http://doi.org/10.1007/978-3-642-20542-2

Lam, T. Y., & Dongol, B. (2020). A Blockchain-enabled e-learning platform. *Interactive Learning Environments*, 1-23. http://doi.org/10.1080/10494820.2020.1716022

Li, C., Guo, J., Zhang, G., Wang, Y., Sun, Y., & Bie, R. (2019). A Blockchain system for E-learning assessment and certification. In *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)* (pp. 212-219). IEEE. http://doi.org/10.1109/SmartIoT.2019.00040

Lilley, M., Meere, J., & Barker, T. (2016). Remote live invigilation: A Pilot study. *Journal of Interactive Media in Education*, *2016*(1). http://doi.org/10.5334/jime.408

Nakamoto, S. (2008). *Bitcoin: A Peer-to-peer electronic cash system*. Bitcoin. https://bitcoin.org/bitcoin.pdf

Nazare J., Duffy K., & Schmidt J. P. (2016). *What we learned from designing an academic certificates system on the blockchain*. MIT Media Lab. https://medium.com/mit-media-lab/what-we-learned-from-designing-an-academic-certificates-system-on-the-blockchain-34ba5874f196

Ngqondi, T., Maoneke, P. B., & Mauwa, H. (2021). A Secure online exams conceptual framework for South African universities. *Social Sciences & Humanities Open*, *3*(1), 100132. http://doi.org/10.1016/j.ssaho.2021.100132

Pavlov, O. V., & Katsamakas, E. (2021). COVID-19 and financial sustainability of academic institutions. *Sustainability*, *13*(7), 3903. http://doi.org/10.3390/su13073903

Porter, S. (2015). *To MOOC or Not to MOOC: How can online learning help to build the future of higher education?* Chandos Publishing. http://doi.org/10.1080/00049670.2016.1183469

Raimundo, R., & Rosário, A. (2021). Blockchain system in the higher education. *European Journal of Investigation in Health, Psychology and Education*, *11*(1), 276-293. http://doi.org/10.3390/ejihpe11010021

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A Method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120-126. http://doi.org/10.1145/359340.359342

Sabbah, Y. W. (2017). Security of online examinations. In *Data Analytics and Decision Support for Cybersecurity* (pp. 157-200). Springer. http://doi.org/10.1007/978-3-319-59439-2_6

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, *22*(11), 612-613. http://doi.org/10.1145/359168.359176

Shand, M., & Vuillemin, J. (1993). Fast implementations of RSA cryptography. In *Proceedings of IEEE 11th Symposium on Computer Arithmetic* (pp. 252-259). IEEE. http://doi.org/10.1109/ARITH.1993.378085

Sharples, M., & Domingue, J. (2016). The Blockchain and kudos: A Distributed system for educational record, reputation and reward. In *European conference on technology enhanced learning* (pp. 490-496). Springer, Cham. http://doi.org/10.1007/978-3-319-45153-4_48

Skiba, D. J. (2017). The Potential of blockchain in education and health care. *Nursing education perspectives*, *38*(4), 220-221. http://doi.org/10.1097/01.NEP.0000000000000190

Sudaryono, S., Aini, Q., Lutfiani, N., Hanafi, F., & Rahardja, U. (2020). Application of blockchain technology for iLearning student assessment. *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, *14*(2), 209-218. http://doi.org/10.22146/ijccs.53109

Traoré, I., Nakkabi, Y., Saad, S., Sayed, B., Ardigo, J. D., & de Faria Quinan, P. M. (2017). Ensuring online exam integrity through continuous biometric authentication. In *Information Security Practices* (pp. 73-81). Springer, Cham. http://doi.org/10.1007/978-3-319-48947-6_6

Turkanović, M., Hölbl, M., Košič, K., Heričko, M., & Kamišalić, A. (2018). EduCTX: A Blockchain-based higher education credit platform. *IEEE access*, *6*, 5112-5127. http://doi.org/10.1109/ACCESS.2018.2789929

Urosevic, K. A. (2019). *Student authentication framework for online exams outside of school* (Unpublished master thesis). Laurea University of Applied Sciences, Vantaa, Finland. https://urn.fi/URN:NBN:fi:amk-201902061979

Wright, R. D. (Ed.). (2014). *Student-teacher interaction in online learning environments*. IGI Global. http://doi.org/10.4018/978-1-4666-6461-6