Hindawi
Security and Communication Networks
Volume 2021, Article ID 8241801, 18 pages
https://doi.org/10.1155/2021/8241801

Research Article

Based on Consortium Blockchain to Design a Credit Verifiable Cross University Course Learning System

Chin-Ling Chen,[1,2,3] **Tianyi Wang**,[1] **Woei-Jiunn Tsaur**,[4] **Wei Weng**,[1] **Yong-Yuan Deng**,[3] **and Jianfeng Cui**[5]

[1]*School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China*
[2]*School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China*
[3]*Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan*
[4]*Computer Center, National Taipei University, New Taipei 237303, Taiwan*
[5]*School of Software Engineering, Xiamen University of Technology, Xiamen 361024, China*

Correspondence should be addressed to Tianyi Wang; 2022031496@stu.xmut.edu.cn and Jianfeng Cui; jfcui@xmut.edu.cn

Received 1 October 2021; Revised 7 November 2021; Accepted 16 November 2021; Published 16 December 2021

Academic Editor: Jiewu Leng

In recent years, the attention of online cross-university courses has been increasing, and students in universities want to increase their knowledge and professional skills by taking online courses from different universities, which raises the issue of course credit verification. In the past, the credits obtained by students in online courses lack endorsement from the education department, and the students' learning process could not be verified. Therefore, the credits of online courses in one university could not be recognized by other universities. The education departments of some countries and regions implement credit conversion rules to convert the credits obtained by students in online courses into university credits or certificates endorsed by the education department. However, these schemes rely too much on the authority of the education department, and the process of students obtaining credits cannot be verified. In addition, the centralized storage method makes the data of education departments at risk of leakage or tampering. With the emergence of blockchain technology, some researchers have proposed the use of blockchain to store students' credits, making it possible to reach consensus among multiple parties on the blockchain while ensuring that credits are not tampered with, but these schemes cannot test the learning process of students and the recognition of credits still relies on the authority of the education department. To solve the above problems, this paper proposes a cross-university course learning system with verifiable credits based on Hyperledger Fabric consortium blockchain technology, and the consortium includes many universities. The credits obtained by students in the course and the hash value of the learning records are stored on the blockchain, and the data on the blockchain is jointly maintained by the universities in the system. One university can verify the homework and final examination of students to check the real ability of students, thus recognizing the credits from other universities, and at the same time, to protect the privacy of students, the important data of students are encrypted for transmission.

1. Introduction

1.1. Background. Education is a way for students to improve themselves, and higher education is directly related to students' development direction and employment prospects. Students want to learn more courses in their universities, or even take courses across universities, to expand their knowledge and skills. Currently, cross-university online courses are popular among students, and students can take courses from different universities online through the Internet. Especially in 2020, when the outbreak of COVID-19 spread worldwide and almost all universities around the world stopped offline courses, online courses ensured the sustainability of education [1]. By April 2021, Udemy, an

American online learning platform, has more than 40 million learners [2]. From September 2020 to July 2021, learners in just one country completed more than 140,000 courses on the Coursera platform of the United States [3]. However, there are limitations to cross-university online courses. Students can acquire more knowledge and skills through cross-university courses, but they lack simple and efficient ways to prove their ability. For a university who offers courses to students from other universities, it promotes the dissemination of professional knowledge and the reputation of the university, and the grades of students' online courses can be used as part of the entrance examination scores. Unfortunately, when it comes to the online courses from other universities, it is difficult for the university to recognize the grades even if students perform well. For teachers, by offering courses for online learning, they can show their research directions and attract more students to engage in research in related fields, but it becomes a challenge to check students' course learning outcomes because teachers do not know the capabilities of these students.

With the increase in the number of cross-university course learners, more and more students want to obtain credits for online courses as their learning credentials for further education or employment, and receiving credits or certificates means that the grades of online courses are recognized by the universities to which the courses belong. Indian Institute of Management (IIM-Kozhikode) announces partnership with Coursera to launch course certificates in business, strategy, marketing, and product management [4]. However, there exists a recognition problem with the credits obtained by taking courses across universities. The credits of online courses are jointly awarded by the learning platform and the university, lacking the verification of authoritative institutions. Although online courses are created by universities, it is difficult for the two universities to reach a consensus on the course content and learners' ability. Therefore, after completing the online course, the credits obtained by students in one university cannot be recognized by other universities.

To solve the problems of credit recognition, the European Credit Transfer and Accumulation System (ECTS) [5] helps students transfer credits between universities. In this system, the credits obtained by students in their universities can be converted into a certain number of ECTS credits, and if a student wants to transfer to another university, the ECTS credits of the student will be converted into the credits of the target university. Moreover, the Credit Bank System (CBS) [6] in Korea allows students to convert their learning achievements into credits and then convert credits into higher education degrees. However, the above schemes have some problems: Firstly, they rely on the management of the central educational institution, and it is because of the authority of the central educational institution that the credits in the system can be recognized. Secondly, the credit conversion rules lack verification for the learning process of students. Although the credit is recognized by all universities in the system, it does not fully reflect the student's ability. Finally, since the system data are stored in a centralized way, there is a risk of data loss or tampering.

The emergence of blockchain technology [7] has provided a new way to solve the credit verification problem [8]. The blockchain can be regarded as a distributed database where the data is tamper-evident, traceable [9–11], maintained by multiple parties, and can reach a consensus among multiple parties [12]. This paper proposes a decentralized cross-university course learning system based on consortium blockchain technology, where the consortium includes many universities. For students, they can take courses from any university in the system and get credits. The hash values of students' homework and final examinations are stored on the blockchain, and the data on the blockchain is maintained by all universities in the system. Since each university manages its teachers and students, the data on the blockchain can reach a consensus among all universities, teachers, and students without relying on a central institution, and one university recognizes course credits obtained by students from other universities by verifying students' homework and final examination.

1.2. Related Works. Currently, research in the education field focuses on storing students' credits, certificates, and learning records via blockchain technology and using distributed storage technology and cryptography to ensure that the data on the blockchain is tamper-evident, thus facilitating the sharing of data among multiple parties. The related works are listed in Table 1.

In 2016, Sharples and Domingue [13] proposed to use blockchain to store students' learning processes and achievements, thus enabling distributed storage of student-related data, but this paper did not introduce the system architecture.

In 2018, Turkanović et al. [14] proposed a blockchain education credit platform named EduCTX where educational institutions can award students credits that can be checked by third parties and students can transfer credits between different educational institutions. Unfortunately, this system can only check whether a student has obtained credits and has no way to verify the student's learning process.

In 2020, Zhao et al. [15] proposed a student portfolio management system that stores students' learning records and teachers' evaluations of students through blockchain technology. However, teachers' evaluations are subjective and cannot objectively reflect students' abilities. In addition, the article failed to provide a method to protect the privacy of students' learning records.

In 2021, Mishra et al. [16] proposed an Ethereum-based student credential sharing system, where universities encrypt the students' credentials before uploading them to the blockchain, and if a third party wants to check students' credentials, the students encrypt credentials before sending them to protect their privacy. Considering that students may not be able to afford the gas in Ethereum, the system has set up a fund organization to provide financial support for system members.

Based on the above research, it can be found that the recognition of students' credits still relies on authority and

Table 1: The related works survey.

Authors	Year	Objective	Technologies	Merits	Demerits
Sharples et al. [13]	2016	A distributed system for the educational record, reputation, and reward	Blockchain	Realize distributed storage of student-related data	No system architecture is proposed
Turkanović et al. [14]	2018	Higher education credit platform	Public blockchain	Student credits can be awarded and transferred	Students' learning process cannot be checked
Zhao et al. [15]	2020	System for student e-portfolio assessment	Consortium blockchain	Realize storage of students' learning records and teachers' evaluation	Security analysis is not sufficient
Mishra et al. [16]	2021	System for sharing students' credentials	Ethereum	Student certificates are encrypted before being uploaded to the system	Consumption of tokens is inevitable
Jeong et al. [17]	2021	The multilateral personal portfolio authentication system	Hyperledger fabric	Detailed system implementation based on hyperledger fabric	Unable to protect the privacy of learning records

the process of obtaining credits cannot be verified. Therefore, this paper proposed a cross-university course learning system with credits verifiable, where a student needs to complete homework and final examination to obtain credit from the university, and other universities can verify the student's homework and final examination to recognize the credit. This system encrypts students' homework and final examinations, thus effectively protecting students' privacy. Moreover, the system adopts consortium blockchain architecture with no token consumption, which improves operational efficiency compared to public blockchain architecture.

The remainder of this paper is organized as follows: Section 2 briefly introduces the preliminary. Section 3 shows the proposed system structure and an application scenario. The paper gives the security and feature analysis in Section 4 and presents the discussion in Section 5. Finally, Section 6 concludes this paper.

2. Preliminary

2.1. Elliptic Curve Cryptography ECDSA. In the blockchain, elliptic curve cryptography [18] is used for digital signature. If a member A in the system wants to send a message M to a member B, the member A needs to digitally sign the message. The process of the signature algorithm is as follows.

2.1.1. Determine Parameters and Generate Keys. The system will first determine the parameters a and b of the elliptic curve $y^2 = (x^3 + \mathrm{ax} + b) \bmod p$, the base p, and the origin G, and then the system will generate private keys d_A and d_B for members A and B, respectively, and generate the public key $Q_A = d_A G$ for the member A.

2.1.2. Generate Signature. Member A selects a random number k, then calculates $H = \mathrm{hash}(M)$, $(x, y) = \mathrm{kG}$, $r = x \bmod p$, $s = k^{-1}(H + \mathrm{rd}_A) \bmod p$, and sends the ECDSA signature pair (r, s) together with the message M to the member B.

2.1.3. Verify Signature. After receiving the signature pair (r, s) and the message M, the member B calculates $H' = \mathrm{hash}(M)$, $u = H' s^{-1} \bmod p$, $v = \mathrm{rs}^{-1} \bmod p$, $(x', y') = \mathrm{uG} + \mathrm{vd}_A G$. If $x' \bmod p = r$, member B confirms that the signature pair (r, s) and the message M sent by the member A are correct.

2.2. Smart Contract. In 1996, Nick Szabo first proposed the concept of the smart contract [19], which digitized contracts in the real world. In a smart contract, both parties will agree on the content of the contract in advance, and the contract will be executed automatically when the conditions are met, without the need for supervision by a third party. Blockchain has the characteristics of nontampering of data and decentralization. The emergence of blockchain technology provides a platform to support the execution of smart contracts, which are jointly executed by the nodes of the whole blockchain network, and the results of their execution become impossible to tamper with after the whole network reaches consensus. Blockchain and smart contract technologies provide new solutions to existing problems in many industries [20], and this paper aims to solve the problem of credit verification across universities.

2.3. Hyperledger Fabric. Hyperledger Fabric was proposed by IBM in 2018 [21], which is an open-source blockchain platform and one of the most popular consortium blockchains so far [22]. Unlike public blockchain systems such as Bitcoin and Ethereum, Hyperledger Fabric uses consortium blockchain technology where system members reach a consensus on transactions without consuming tokens. Different from Bitcoin and Ethereum where the data is publicly accessible [23], all members who join the Hyperledger Fabric network need to be authenticated to ensure that unrelated people cannot join the network and get the data on the blockchain. Since the identities of all members in the network are known, the nodes of Hyperledger Fabric do not need to reach a consensus through Proof of Work (POW) [24], and the number of transactions generated in

Hyperledger Fabric can reach 3500 per second, which is much higher than 3.5 of Bitcoin and 5.4 of Ethereum [25].

In addition to Hyperledger Fabric, there are currently many consortium blockchain platforms in the market, such as Ethereum [26], Corda [27], Quorum [28], and MultiChain [29]. Compared with the above platforms, Hyperledger Fabric has higher throughput and shorter latency [30], and it has wide interest and application in many industries (including finance, IoT, supply chain, manufacturing, and technology) [31]. Therefore, this paper chooses to design a credit verifiable cross-university course learning system based on Hyperledger Fabric.

There are mainly the following components in the Hyperledger Fabric network:

(1) Certificate authority (CA): certificate authority provides an identity authentication mechanism for system users. Before a user can interact with the blockchain network, he or she needs to connect to a CA server, which provides the user with identity information as well as the public and private keys.

(2) Orderer: an orderer node collects the endorsed transactions sent by users from client nodes and packages them into blocks, then sends these blocks to the peer nodes.

(3) Peer: peer nodes are divided into endorse peer, leader peer, and anchor peer, where the endorse peer calls chaincode to simulate the execution of a transaction and endorse the transaction, the leader peer broadcasts the block received from the orderer node to all the peer nodes in the organization, and the anchor peer exchange data with other anchor peers between different organizations. All the peer nodes can be considered as committer peers who check each transaction in the received block and update the ledger after the check is completed, and the ledger consists of a blockchain that stores all the transactions and the World State that stores the state data of all the members in the system.

(4) Client: the client node is operated by a system member, which must be connected to one of the peer nodes or orderer nodes to communicate with the blockchain network. Firstly, the client node sends a transaction proposal to the peer in the organization for endorsement. Once the client node has received a sufficient number of signed proposal responses from endorse peers, the client node sends the transaction containing endorsed transaction proposal responses to the orderer node, and the orderer node orders the transactions into blocks.

(5) Channel: channel can achieve isolation of different services and there is only one blockchain in a channel. Users need to get certificates from the CA node firstly, then they can communicate with the peer node or orderer node through the channel.

(6) Organization: organizations represent entities such as enterprises and institutions in the blockchain network. Each organization contains endorse peer, leader peer, and anchor peer that store the ledger, and each member in the system belongs to an organization.

(7) Chaincode: chaincode can be regarded as the smart contract in Hyperledger Fabric, which is written in some language and is deployed on every peer node, and users can achieve query and modification of data on the blockchain by invoking chaincode.

3. System Model

3.1. System Architecture. This research proposed a cross-university course learning system based on Hyperledger Fabric where the consortium includes many universities. The main members of the system include university administrators and users, and users include teachers and students. Teachers, students, and university administrators in the same university form an organization, and each university has administrators to manage its users. The system architecture is shown in Figure 1.

The members of the system are described as follows.

(1) Certificate authority (CA): CA nodes provide certificates, public and private keys for users who want to join the system, and each university has its own CA node.

(2) University: a university is managed by university administrators, and there are many teachers and students and some administrators in each university.

(3) University administrator: a university administrator creates the channel, joins the channel with the peers in his or her organization, installs the chaincode on each peer node and initializes the chaincode, reviews the identity of users, and creates system accounts for them, reviews courses created by the teachers and checks students' grades. By invoking chaincode, administrators add course information to teachers' accounts, award credits to students, and verify students' homework and final examinations.

(4) Teacher: teachers apply to their universities for teaching courses and grade students' homework and final examinations. By invoking chaincode, teachers add course information and grades to students' accounts.

(5) Students apply to the teachers for learning courses, submit homework and final examinations to the teachers for grades and apply to their universities for course credits. By invoking chaincode, students add the hash values of their homework and final examinations to their accounts.

3.2. Application Scenario. Figure 2 shows the application scenario where Student A of University A wants to learn Course B of Teacher B who comes from University B, and University C wants to verify the credit obtained by Student A.

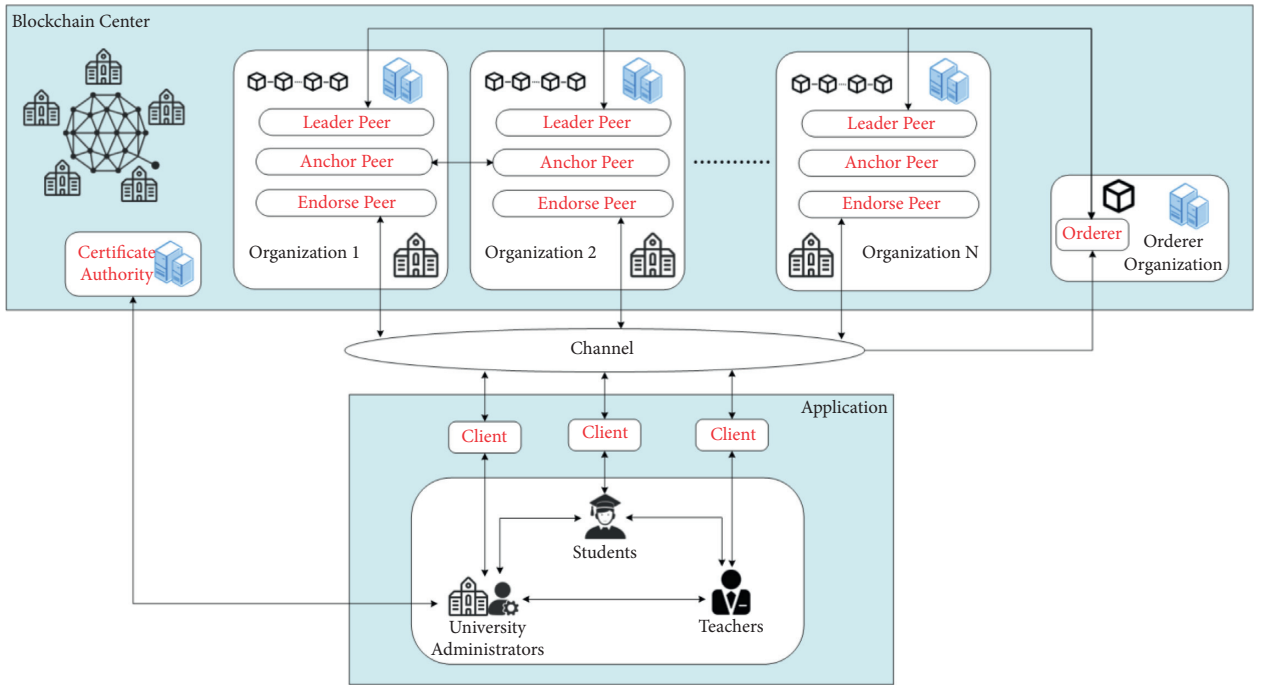

Figure 1: System structure.

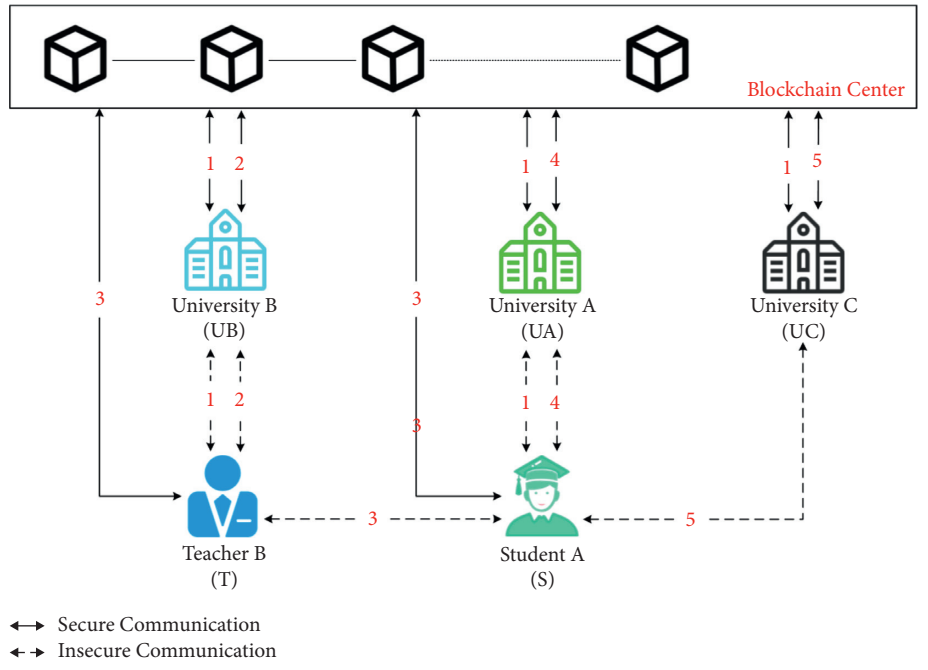

Figure 2: Application scenario.

Step 1: the administrator of a university first creates a channel, then the administrator of each university in the system joins the channel with the peers in his or her organization, installs the chaincode on each peer node, and initializes the chaincode. All users (including teachers and students) need to apply for registration with the administrators of their universities first to obtain a system account.

Step 2: Teacher B applies to University B for teaching Course B, which needs to be reviewed by his university. After the course is approved, University B adds the course information to the account of Teacher B by invoking chaincode.

Step 3: Student A applies to Teacher B for learning Course B, and Teacher B adds the course information to the account of Student A by invoking chaincode.

During the course learning period, Student A needs to upload the hash value of homework and final examination and send them to Teacher B for grading, then Teacher B will upload the grade of Student A by invoking chaincode.

Step 4: Student A applies to his university for obtaining the credit of Course B after receiving the grade from Teacher B. By invoking chaincode, University A checks the grade of Student A and awards credit if the grade is qualified.

Step 5: Student A wants to transfer from University A to University C, and University C and, therefore, wants to verify the credits of Student A. University C submits the credit verification request to Student A, and Student A sends the homework and final examination to University C. Before reviewing the content of homework and final examination, University C needs to calculate the hash value of the homework and final examination, then compares it with the hash value uploaded by Student A to ensure that the homework and final examination have never been tampered with.

3.3. Initial Phase. In the initial phase, the university administrator creates a channel, then installs, and initializes the chaincode for each peer node in the organization.

Step 1: the university administrator logs in to the system through an application program, then starts a client node and creates a channel.

Step 2: the university administrator connects to each peer node in the organization in turn through the channel, then installs the chaincode on each peer node and initializes the chaincode. The chaincode is shown in Algorithm 1.

3.4. User Registration Phase. In this phase, User X submits registration application and identity information to his or her university. After verifying the identity, the university administrator connects to the CA node to generate the certificate, the public key, and the private key of User X. Figure 3 shows the flowchart of the user registration phase.

Step 1: User X generates the registration application M_{Request} and identity information M_{Identity}, then transmits $M_{\text{Request}}, M_{\text{Identity}}$ to his or her university

Step 2: University administrators verify M_{Identity}, then transmit M_{Identity} to the CA node if M_{Identity} is valid. CA generates the private key d_X, the public key Q_X, and the certificate Cert_X of User X based on M_{Identity}

Step 3: The application program generates the system account and ID_X of User X based on d_X, Q_X, Cert_X and M_{Identity}, then sends (d_X, Q_X, ID_X) to User X

3.5. Course Registration Phase. In this phase, Teacher B sends Course B to his university for review. If Course B is valid, the administrator of University B adds the course information to the account of Teacher B by invoking chaincode. Figure 4 shows the flowchart of the course registration phase.

Step 1: Teacher B wants to add Course B to the system, first he generates M_{Course} and the course teaching request M_{Request}, then chooses a random number k_1, calculates $H_1 = \text{hash}(M_{\text{Course}}, \text{ID}_T, M_{\text{Request}}, \text{TS}_T)$, $(x_1, y_1) = k_1 G$, $r_1 = x_1 \bmod p$, $s_1 = k_1^{-1}(H_1 + r_1 d_T) \bmod p$, and sends $M_{\text{Request}}, \text{ID}_T, M_{\text{Course}}, \text{TS}_T, \text{Cert}_T, (r_1, s_1)$ to University B.

Step 2: after receiving M_{Request} from Teacher B, the administrator of University B first uses $\text{TS}_{\text{NOW}} - \text{TS}_T \le \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_T of teacher B by ID_T, verifies Cert_T by Q_T, and calculates $H_2 = \text{hash}(M_{\text{Course}}, \text{ID}_T, M_{\text{Request}}, \text{TS}_T)$, $u_1 = H_2 s_1^{-1} \bmod p$, $v_1 = r_1 s_1^{-1} \bmod p$, $(x_2, y_2) = u_1 G + v_1 Q_T$, check $x_2 \bmod p \stackrel{?}{=} r_1$. If the signature verification is passed, the university administrator reviews Course B, generates the result M_{Result}, and invokes the chaincode CheckCourse. The chaincode is shown in Algorithm 2.

3.6. Course Learning Phase. If Student A wants to learn Course B from University B, he will first apply for course learning to Teacher B and Teacher B adds the course information to his account by invoking chaincode. During the period of learning Course B, Student A uploads the hash value of his homework and final examination by invoking chaincode and sends them to Teacher B for grading, then Teacher B uploads the grade of Student A by invoking chaincode. Figure 5 shows the flowchart of the course learning phase.

Step 1: Student A selects Course B that he wants to learn, then generates the course learning application M_{Request}, chooses a random number k_3, calculates $H_5 = \text{hash}(M_{\text{Request}}, \text{ID}_C, \text{ID}_S, \text{TS}_S)$, $(x_5, y_5) = k_3 G$, $r_3 = x_5 \bmod p$, $s_3 = k_3^{-1}(H_5 + r_3 d_S) \bmod p$, and sends $M_{\text{Request}}, \text{ID}_C, \text{ID}_S, \text{TS}_S, \text{Cert}_S, (r_3, s_3)$ to Teacher B.

Step 2: after receiving M_{Request} from Student A, Teacher B first uses $\text{TS}_{\text{NOW}} - \text{TS}_S \le \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_S of Student A by ID_S, verifies Cert_S by Q_S, and calculates $H_6 = \text{hash}(M_{\text{Request}}, \text{ID}_C, \text{ID}_S, \text{TS}_S)$, $u_3 = H_6 s_3^{-1} \bmod p$, $v_3 = r_3 s_3^{-1} \bmod p$, $(x_6, y_6) = u_3 G + v_3 Q_S$, $x_6 \bmod p \stackrel{?}{=} r_3$. If the signature verification is passed, Teacher B invokes the chaincode LearnCourse. The chaincode is shown in Algorithm 3.

Then, Student A chooses a random number k_4, calculates $H_8 = \text{hash}(M_{\text{Work}}, \text{ID}_C, \text{ID}_S, \text{TS}_S)$, $(x_7, y_7) = k_4 G$, $r_4 = x_7 \bmod p$, $s_4 = k_4^{-1}(H_8 + r_4 d_S) \bmod p$, and sends $C_{\text{Work}}, \text{ID}_C, \text{ID}_S, \text{TS}_S, \text{Cert}_S, (r_4, s_4)$ to Teacher B.

Step 4: after receiving C_{Work} from Student A, Teacher B first uses $\text{TS}_{\text{NOW}} - \text{TS}_S \le \Delta T$ to confirm whether the timestamp is valid, then verifies Cert_S by Q_S, decrypts C_{Work} with his private key d_T, generates $M_{\text{Work}} = D_{d_T}(C_{\text{Work}})$ and calculates $H_9 = \text{hash}(M_{\text{Work}},$

```
(1) type Chaincode struct {
(2) }
(3) type Teacher struct {
(4)     Name string ‘json:”name”’
(5)     University string ‘json:”university”’
(6)     Course string ‘json:”course”’
(7) }
(8) type Student struct {
(9)     Name string ‘json:”name”’
(10)    University string ‘json:”university”’
(11)    Hash string ‘json:”hash”’
(12)    Grade int ‘json:”grade”’
(13)    Credit int ‘json:”credit”’
(14) }
(15) func (t *Chaincode) Init(stub shim.ChaincodeStubInterface) peer.Response {
(16)    return shim.Success(nil)
(17) }
(18) func (t *Chaincode) Invoke(stub shim.ChaincodeStubInterface) peer.Response {
(19)    function, args: = stub.GetFunctionAndParameters()
(20)    if function = = “CheckCourse” {
(21)        return t.CheckCourse(stub, args)
(22) } else if function = = “LearnCourse” {
(23)        return t.LearnCourse(stub, args)
(24) } else if function = = “WorkUpload” {
(25)        return t.WorkUpload(stub, args)
(26) } else if function = = “AddGrade” {
(27)        return t.AddGrade(stub, args)
(28) } else if function = = “CheckStudent” {
(29)        return t.CheckStudent(stub, args)
(30) } else if function = = “AwardCredit” {
(31)        return t.AwardCredit(stub, args)
(32) }
(33) return shim.Error(“Invalid Smart Contract function name. “)
(34) }
```

Algorithm 1: Chaincode used for initialization.

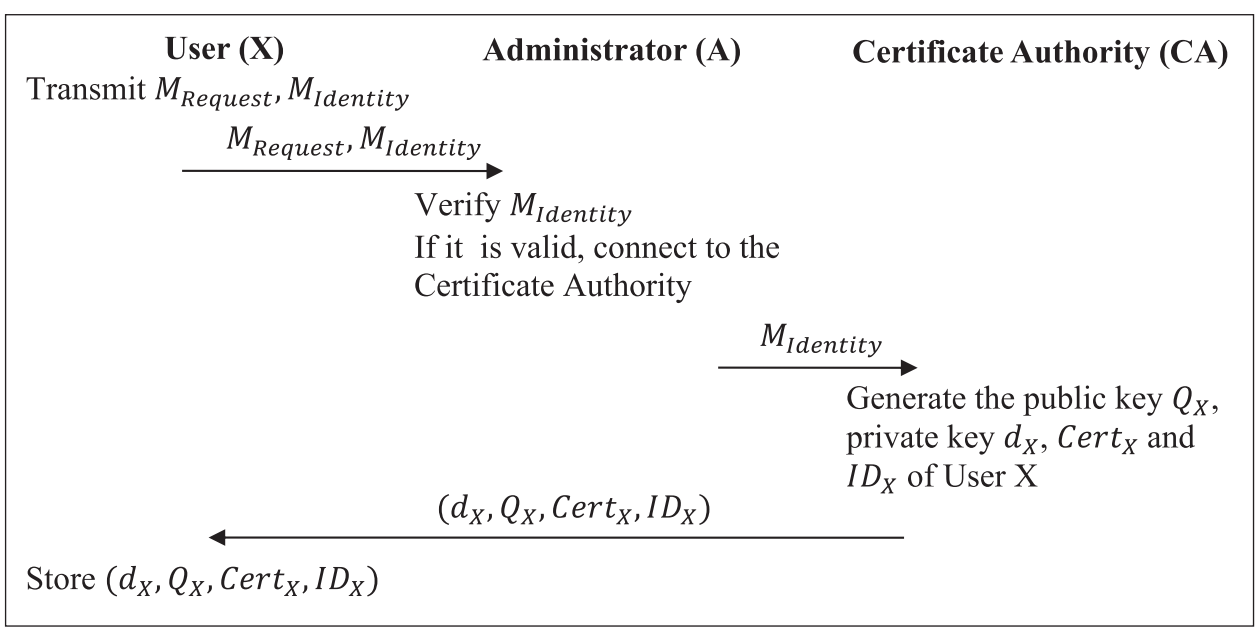

Figure 3: User registration phase.

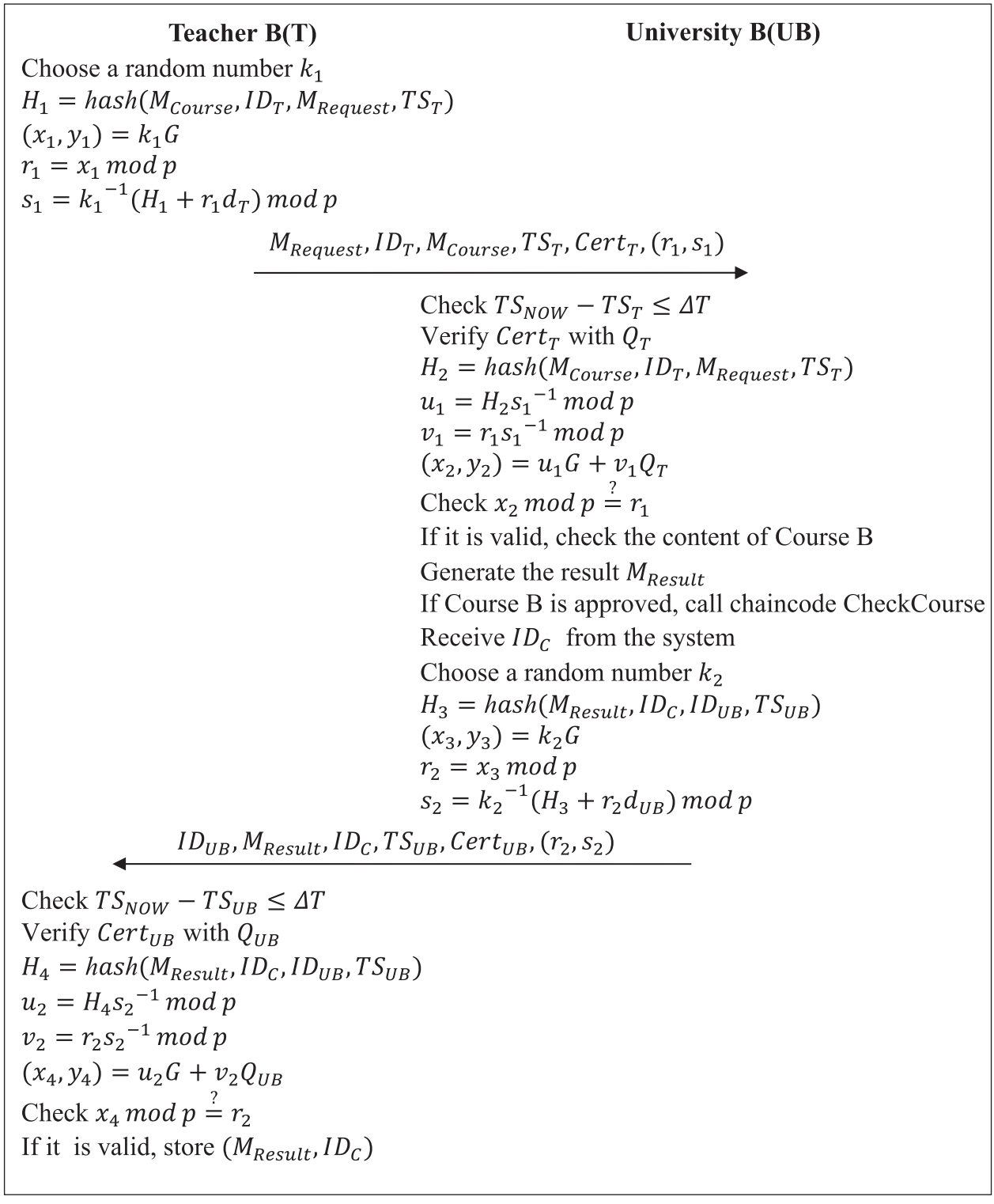

Figure 4: Course registration phase.

ID_C, ID_S, TS_S), $u_4 = H_9 s_4^{-1} \text{mod} p$, $v_4 = r_4 s_5^{-1} \text{mod} p$, $(x_8, y_8) = u_4 G + v_4 Q_S$, $x_8 \text{mod} p \stackrel{?}{=} r_4$. If the signature verification is passed, Teacher B reviews M_{Work} and generates the grade g, then he uploads the grade by invoking the chaincode AddGrade. The chaincode is shown in Algorithm 5.

3.7. Credit Application Phase. In this phase, Student A applies to his university for the credit of Course B. If the grade of Student A is qualified, the administrator of University A awards him the credit by invoking the chaincode. Figure 6 shows the flowchart of the credit application phase.

```
(1) func (t *Chaincode) CheckCourse(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2) if len(args) ! = 4 {
(3)     return shim.Error("nu Incorrect mber of arguments. Expecting 4″)
(4) }
(5) var teacher = Teacher{Name: args [1], University: args [2], Course: args [3]}
(6) teacherAsBytes, _: = json.Marshal(teacher)
(7) APIstub.PutState(args[0], teacherAsBytes)
(8) return shim.Success(nil)
(9) }
```

Step 3: after receiving $\mathrm{ID}_{\mathrm{UB}}, M_{\mathrm{Result}}, \mathrm{ID}_C, \mathrm{TS}_{\mathrm{UB}}, \mathrm{Cert}_{\mathrm{UB}}, (r_2, s_2)$ from University B, Teacher B first uses $\mathrm{TS}_{\mathrm{NOW}} - \mathrm{TS}_{\mathrm{UB}} \le \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_{UB} of University B by $\mathrm{ID}_{\mathrm{UB}}$, verifies $\mathrm{Cert}_{\mathrm{UB}}$ by Q_{UB}, and calculates $H_4 = \mathrm{hash}(M_{\mathrm{Result}}, \mathrm{ID}_C, \mathrm{ID}_{\mathrm{UB}}, \mathrm{TS}_{\mathrm{UB}})$, $u_2 = H_4 s_2^{-1} \bmod p$, $v_2 = r_2 s_2^{-1} \bmod p$, $(x_4, y_4) = u_2 G + v_2 Q_{\mathrm{UB}}$, $x_4 \bmod p \stackrel{?}{=} r_2$.

Algorithm 2: Chaincode for the university to check courses.

Step 1: Student A generates the credit application M_{Request} and sends $M_{\mathrm{Request}}, \mathrm{ID}_C, \mathrm{ID}_S$ to University A.

Step 2: after receiving $M_{\mathrm{Request}}, \mathrm{ID}_C, \mathrm{ID}_S$ from Student A, the administrator of University A invokes the chaincode CheckStudent to check the grade g of Student A. The chaincode is shown in Algorithm 6. If the grade g is qualified, the administrator adds the credit to the account of Student A by invoking the chaincode AwardCredit. The chaincode is shown in Algorithm 7.

3.8. Credit Verification Phase. Student A wants to transfer from University A to University C, and thus University C wants to verify the credits of Student A. University C first applies for credit verification, then Student A sends his homework and final examination which are encrypted to University C. After decrypting the message and ensuring that the data is not tampered with, the administrator of University C reviews the content of the homework and final examination. Figure 7 shows the flowchart of the credit verification phase.

Step 1: University C generates the credit verification application M_{Request}, chooses a random number k_5, calculates $H_{10} = \mathrm{hash}(M_{\mathrm{Request}}, \mathrm{ID}_C, \mathrm{ID}_{\mathrm{UC}}, \mathrm{TS}_{\mathrm{UC}})$, $(x_9, y_9) = k_5 G$, $r_5 = x_9 \bmod p$, $s_5 = k_5^{-1}(H_{10} + r_5 d_{\mathrm{UC}}) \bmod p$, and sends $M_{\mathrm{Request}}, \mathrm{ID}_C, \mathrm{ID}_{\mathrm{UC}}, \mathrm{TS}_{\mathrm{UC}}, \mathrm{Cert}_{\mathrm{UC}}, (r_5, s_5)$ to Student A.

Step 2: after receiving M_{Request} from University C, Student A first uses $\mathrm{TS}_{\mathrm{NOW}} - \mathrm{TS}_{\mathrm{UC}} \le \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_{UC} of University C by $\mathrm{ID}_{\mathrm{UC}}$, verifies $\mathrm{Cert}_{\mathrm{UC}}$ by Q_{UC}, and calculates $H_{11} = \mathrm{hash}(M_{\mathrm{Request}}, \mathrm{ID}_C, \mathrm{ID}_{\mathrm{UC}}, \mathrm{TS}_{\mathrm{UC}})$, $u_5 = H_{11} s_5^{-1} \bmod p$, $v_5 = r_5 s_5^{-1} \bmod p$, $(x_{10}, y_{10}) = u_5 G + v_5 Q_{\mathrm{UC}}$, $x_{10} \bmod p \stackrel{?}{=} r_5$. If the signature verification is passed, Student A generates M_{Result}, encrypts M_{Work} with the public key Q_{UC}, generates $C'_{\mathrm{Work}} = E_{Q_{\mathrm{UC}}}(M_{\mathrm{Work}})$, chooses a random number k_6, calculates $H_{12} = \mathrm{hash}(M_{\mathrm{Result}}, M_{\mathrm{Work}}, \mathrm{ID}_C, \mathrm{ID}_S, \mathrm{TS}_S)$, $(x_{11}, y_{11}) = k_6 G$, $r_6 = x_{11} \bmod p$, $s_6 = k_6^{-1}(H_{12} + r_6 d_S) \bmod p$, and sends $M_{\mathrm{Result}}, C_{\mathrm{Work}}, \mathrm{ID}_C, \mathrm{ID}_S, \mathrm{TS}_S, \mathrm{Cert}_S, (r_6, s_6)$ to Student A.

Step 3: after receiving M_{Result} from Student A, the administrator of University C first uses $\mathrm{TS}_{\mathrm{NOW}} - \mathrm{TS}_S \le \Delta T$ to confirm whether the timestamp is valid, then searches for the public key Q_S of Student A by ID_S, verifies $Cert_S$ by Q_S, decrypts C'_{Work} with the private key d_{UC}, generates $M_{\mathrm{Work}} = D_{d_{\mathrm{UC}}}(C_{\mathrm{Work}})$ and calculates $H_{13} = \mathrm{hash}(M_{\mathrm{Result}}, M_{\mathrm{Work}}, \mathrm{ID}_C, \mathrm{ID}_S, \mathrm{TS}_S)$, $u_6 = H_{13} s_6^{-1} \bmod p$, $v_6 = r_6 s_6^{-1} \bmod p$, $(x_{12}, y_{12}) = u_6 G + v_6 Q_S$, $x_{12} \bmod p \stackrel{?}{=} r_6$. If the signature verification is passed, the administrator checks H_7 which was uploaded by Student A by invoking the chaincode CheckStudent and calculates $H_{14} = \mathrm{hash}(M_{\mathrm{Work}})$. If $H_{14} = H_7$, it means that M_{Work} sent by Student A to University C are the same as the homework and final examination which were submitted by Student A in the course learning phase. Having ensured that the homework and final examination are not tampered with, the administrator of University C reviews the content of M_{Work}.

4. Security and Feature Analysis

4.1. Mutual Authentication. In this paper, BAN logic [32] was used for identity authentication. The notation of BAN logic is described as below.

(1) $< X >_K$ The message X is combined with a key K
(2) $P| \equiv XP$ believes X
(3) $P \overset{K}{\leftrightarrow} QP$ and Q use a shared key K to communicate
(4) $\#(X)$ The message X is fresh
(5) $P| \Rightarrow XP$ has jurisdiction over X
(6) $P \triangleleft XP$ sees X
(7) $P| \sim XP$ once said X

The main goals of the scheme are to authenticate the identity between User X and User Y.

(1) G1: $X| \equiv X \overset{x_{Y-X}}{\leftrightarrow} Y$
(2) G2: $X| \equiv Y| \equiv X \overset{x_{Y-X}}{\leftrightarrow} Y$
(3) G3: $Y| \equiv X \overset{x_{X-Y}}{\leftrightarrow} Y$
(4) G4: $Y| \equiv X| \equiv X \overset{x_{X-Y}}{\leftrightarrow} Y$
(5) G5: $X| \equiv \mathrm{ID}_Y$

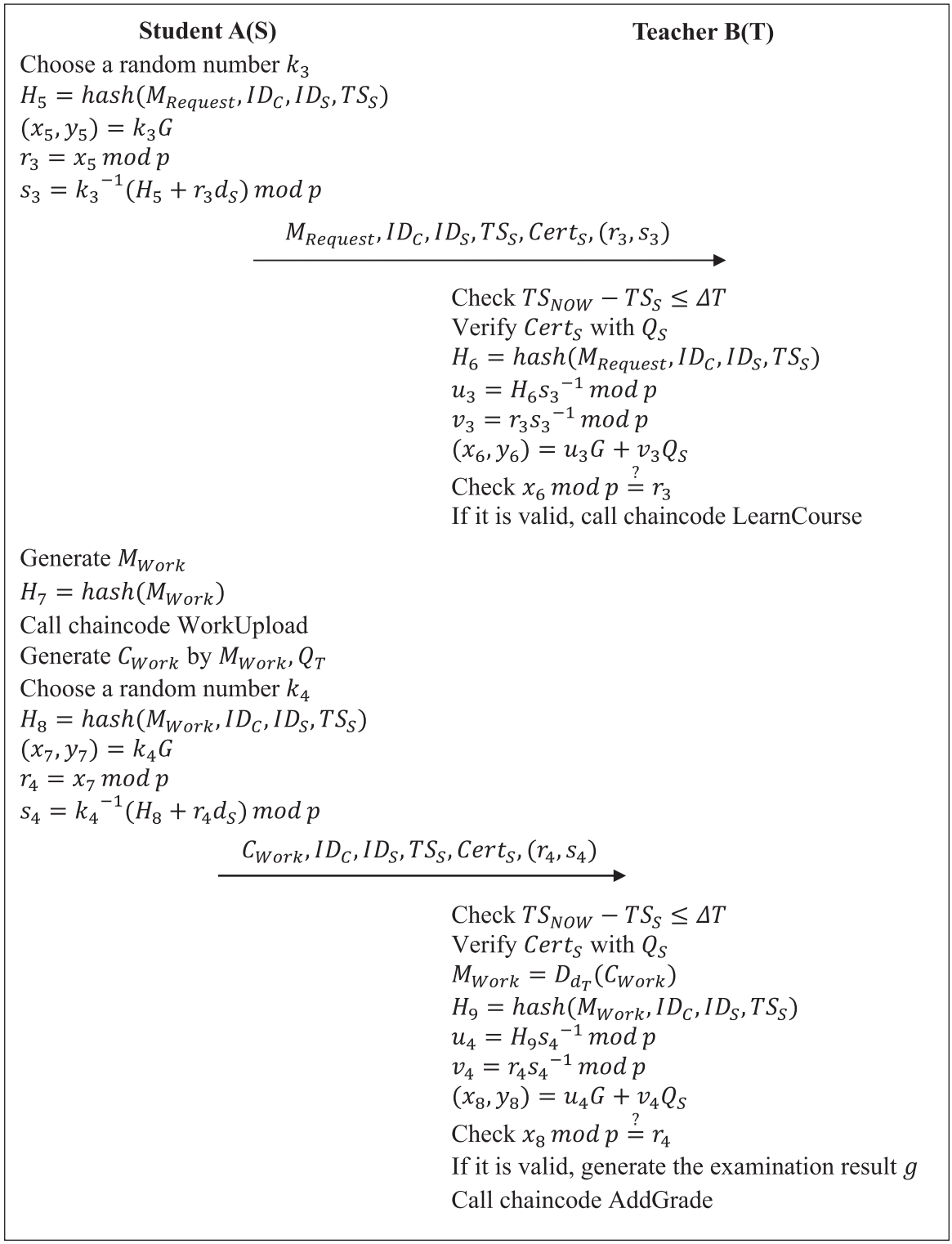

Figure 5: Course learning phase.

(6) G6: $X| \equiv Y| \equiv \mathrm{ID}_Y$

(7) G7: $Y| \equiv \mathrm{ID}_X$

(8) G8: $Y| \equiv X| \equiv \mathrm{ID}_X$

BAN logic is used for producing an idealized form as follows:

(1) M1: $(<\mathrm{hash}(\mathrm{ID}_X, M_X, \mathrm{TS}_X)>_{x_{X-Y}})$

(2) M2: $(<\mathrm{hash}(\mathrm{ID}_Y, M_Y, \mathrm{TS}_Y)>_{x_{Y-X}})$

It is necessary to make the following assumptions before analyzing the proposed scheme:

(1) A1: $X| \equiv \#(\mathrm{TS}_X)$

```
(1) func (t *Chaincode) LearnCourse(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)     if len(args) != 6 {
(3)         return shim.Error("Incorrect number of arguments. Expecting 6")
(4)     }
(5)     var student = Student{Name: args [1], University: args [2], Hash: args [3], Grade: args [2], Credit: args [3]}
(6)     studentAsBytes, _: = json.Marshal(student)
(7)     APIstub.PutState(args[0], studentAsBytes)
(8)     return shim.Success(nil)
(9) }
```

Step 3: When Student A finishes his homework and final examination M_{Work}, he searches for the public key Q_T of Teacher B by ID_T, encrypts M_{Work} with the public key Q_T, generates $C_{\text{Work}} = E_{Q_T}(M_{\text{Work}})$, calculates $H_7 = \text{hash}(M_{\text{Work}})$, and uploads H_7 by invoking the chaincode WorkUpload. The chaincode is shown in Algorithm 4.

Algorithm 3: Chaincode for the teacher to approve students to join the course.

```
(1) func (t *Chaincode) WorkUpload(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)     if len(args) != 2 {
(3)         return shim.Error("Incorrect number of arguments. Expecting 2")
(4)     }
(5)     studentAsBytes, _: = APIstub.GetState(args[0])
(6)     student:= Student{}
(7)     json.Unmarshal(studentAsBytes, &student)
(8)     student.Hash = args[3].
(9)     studentAsBytes, _ = json.Marshal(student)
(10)    APIstub.PutState(args[0], studentAsBytes)
(11)    return shim.Success(nil)
(12) }
```

Algorithm 4: Chaincode for the student to upload the hash value of his homework and examination.

```
(1) func (t *Chaincode) AddGrade(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)     if len(args) != 2 {
(3)         return shim.Error("Incorrect number of arguments. Expecting 2")
(4)     }
(5)     studentAsBytes, _: = APIstub.GetState(args[0])
(6)     student:= Student{}
(7)     json.Unmarshal(studentAsBytes, &student)
(8)     student.Grade = args[4].
(9)     studentAsBytes, _ = json.Marshal(student)
(10)    APIstub.PutState(args[0], studentAsBytes)
(11)    return shim.Success(nil)
(12)}
```

Algorithm 5: Chaincode for the teacher to add course grades to the student account.

(2) A2: $Y| \equiv \#(\text{TS}_X)$
(3) A3: $X| \equiv \#(\text{TS}_Y)$
(4) A4: $Y| \equiv \#(\text{TS}_Y)$
(5) A5: $X| \equiv Y| \Rightarrow X \overset{x_{Y-X}}{\leftrightarrow} Y$
(6) A6: $Y| \equiv X| \Rightarrow X \overset{x_{X-Y}}{\leftrightarrow} Y$
(7) A7: $X| \equiv Y| \Rightarrow \text{ID}_Y$
(8) A8: $Y| \equiv X| \Rightarrow \text{ID}_X$

According to these assumptions, the main proof of the authentication is as follows:

4.1.1. User Y Authenticates User X. By M1 and the seeing rule, derive

$$Y \triangleleft \left(< \text{hash}(\text{ID}_X, M_X, \text{TS}_X) >_{x_{X-Y}} \right). \tag{1}$$

By A2 and the freshness rule, derive

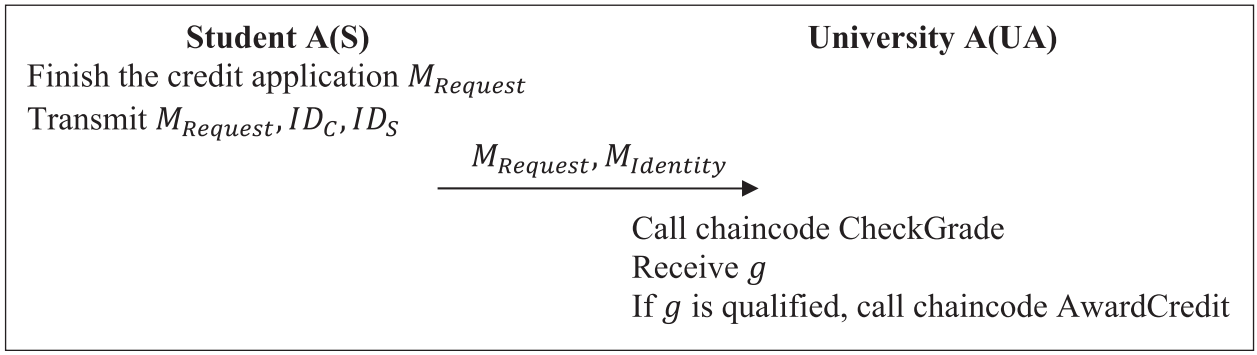

Figure 6: Credit application phase.

```
(1) func (t *Chaincode) CheckStudent(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)     if len(args) != 1 {
(3)         return shim.Error("Incorrect number of arguments. Expecting 1")
(4)     }
(5)     studentAsBytes, _ := APIstub.GetState(args[0])
(6)     return shim.Success(studentAsBytes)
(7) }
```

Algorithm 6: Chaincode for university and teachers to check student's information on the blockchain.

```
(1) func (t *Chaincode) AwardCredit(APIstub shim.ChaincodeStubInterface, args []string) peer.Response {
(2)     if len(args) != 2 {
(3)         return shim.Error("Incorrect number of arguments. Expecting 2")
(4)     }
(5)     studentAsBytes, _ := APIstub.GetState(args[0])
(6)     student:= Student{}
(7)     json.Unmarshal(studentAsBytes, &student)
(8)     student.Credit = args[5].
(9)     studentAsBytes, _ = json.Marshal(student)
(10)    APIstub.PutState(args[0], studentAsBytes)
(11)    return shim.Success(nil)
(12) }
```

Algorithm 7: Chaincode for the university to award student credit.

$$Y| \equiv \#\left(< \text{hash}\left(\text{ID}_X, M_X, \text{TS}_X\right) >_{x_{X-Y}} \right). \tag{2}$$

By A6, statement 1, and the message meaning rule, derive

$$Y| \equiv X| \sim \left(< \text{hash}\left(\text{ID}_X, M_X, \text{TS}_X\right) >_{x_{X-Y}} \right). \tag{3}$$

By statement 2, statement 3, and the nonce verification rule, derive

$$Y| \equiv X| \equiv \left(< \text{hash}\left(\text{ID}_X, M_X, \text{TS}_X\right) >_{x_{X-Y}} \right). \tag{4}$$

By statement 4 and the belief rule, derive

$$Y| \equiv X| \equiv X \overset{x_{X-Y}}{\leftrightarrow} Y. \tag{5}$$

By A6, statement 5, and the jurisdiction rule, derive

$$Y| \equiv X \overset{x_{X-Y}}{\leftrightarrow} Y. \tag{6}$$

By statement 6 and the belief rule, derive

$$Y| \equiv X| \equiv \text{ID}_X. \tag{7}$$

By A8, statement 7, and the jurisdiction rule, derive

$$Y| \equiv \text{ID}_X. \tag{8}$$

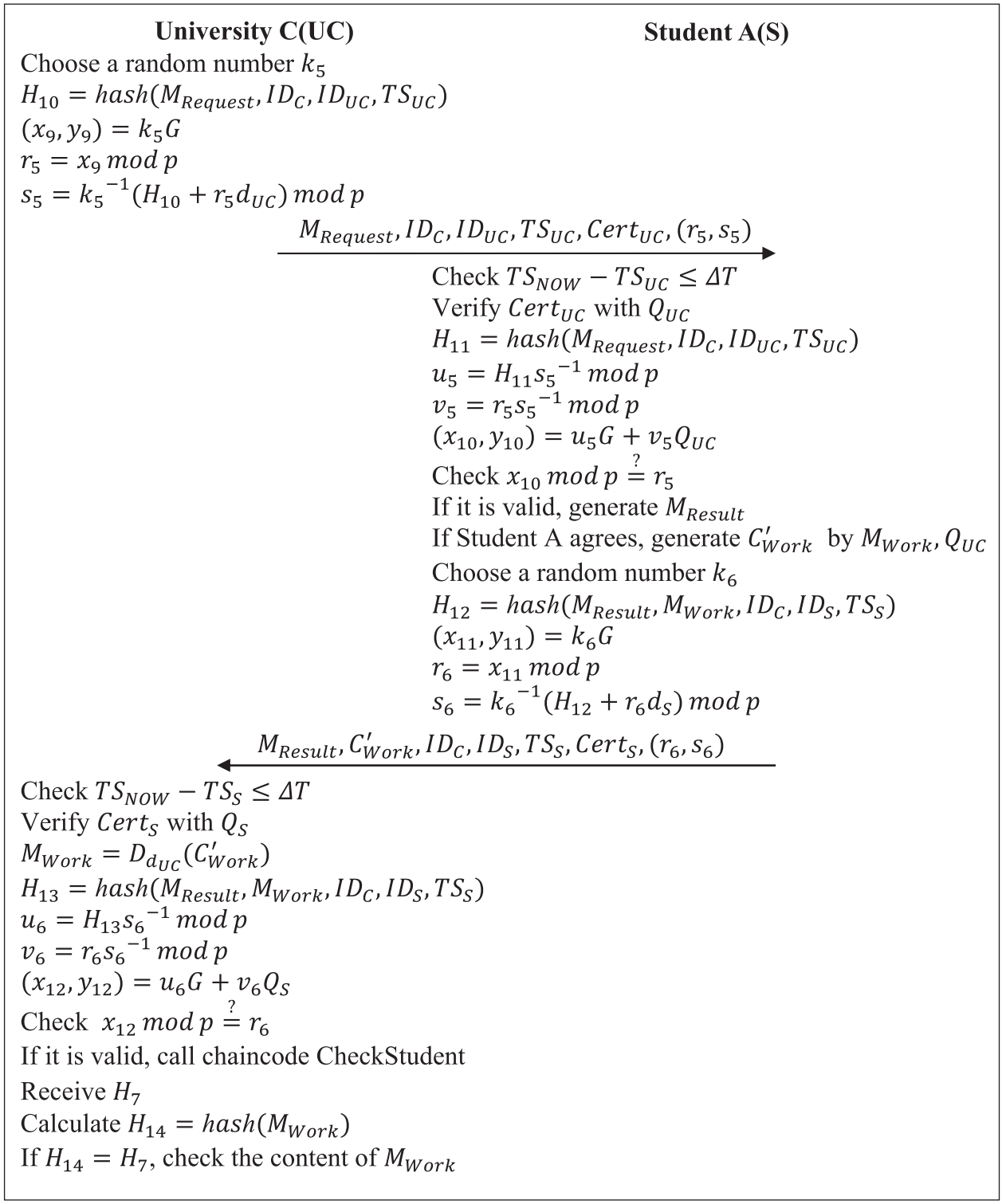

Figure 7: Credit verification phase.

4.1.2. User X Authenticates User Y. By M2 and the seeing rule, derive

$$X \triangleleft \left(< \text{hash}\left(\text{ID}_Y, M_Y, \text{TS}_Y\right) >_{x_{Y-X}} \right). \tag{9}$$

By A3 and the freshness rule, derive

$$X| \equiv \#\left(< \text{hash}\left(\text{ID}_Y, M_Y, \text{TS}_Y\right) >_{x_{Y-X}} \right). \tag{10}$$

By A5, statement 9, and the message meaning rule, derive

$$X| \equiv Y| \sim \left(< \text{hash}\left(\text{ID}_Y, M_Y, \text{TS}_Y\right) >_{x_{Y-X}} \right). \tag{11}$$

By statement 10, statement 11, and the nonce verification rule, derive

$$X| \equiv Y| \equiv \left(< \text{hash}\left(\text{ID}_Y, M_Y, \text{TS}_Y\right) > _{x_{Y-X}} \right). \tag{12}$$

By statement 12 and the belief rule, derive

$$X| \equiv Y| \equiv X \overset{x_{Y-X}}{\leftrightarrow} Y. \tag{13}$$

By A5, statement 13, and the jurisdiction rule, derive

$$X| \equiv X \overset{x_{Y-X}}{\leftrightarrow} Y. \tag{14}$$

By statement 14 and the belief rule, derive

$$X| \equiv Y| \equiv \text{ID}_Y. \tag{15}$$

By A7, statement 15, and the jurisdiction rule, derive

$$X| \equiv \text{ID}_Y. \tag{16}$$

By statement 5, statement 6, statement 7, statement 8, statement 13, statement 14, statement 15, and statement 16, it can be proven that User X and User Y authenticate each other.

4.2. Data Integrity. In this paper, Elliptic Curve Digital Signature Algorithm (ECDSA) is used to ensure data integrity, and any important data transmitted in the system need to be signed by the sender. In the process of data transmission, the sender first calculates the hash value of the data and then generates the digital signature by the hash value and his private key. After receiving the data and the digital signature from the sender, the receiver also calculates the hash value of the data and then searches for the public key of the sender. If the sender's signature can be generated by the receiver's hash value and the sender's public key, then it means that the data transmitted in the system has not been tampered with or lost.

Scenario: for example, Student A wants to send the course learning application M_{Request} of Course B to Teacher B. During data transmission, the data may be lost, or someone wants to maliciously tamper with the content of the application, then Teacher B receives M'_{Request} which is different from M_{Request}.

Analysis: after receiving $M'_{\text{Request}}, \text{ID}_C, \text{ID}_S, \text{TS}_S, \text{Cert}_S, (r_3, s_3)$ from Student A and verifying the timestamp and Cert_S, Teacher B calculates $H' = \text{hash}(M'_{\text{Request}}, \text{ID}_C, \text{ID}_S, \text{TS}_S)$, $u' = H's_3^{-1} \bmod p$, $v_3 = r_3 s_3^{-1} \bmod p$, $(x', y') = u'G + v_3 Q_S$, if $x' \bmod p \neq r_3$ and then realizes that M'_{Request} was not generated by Student A.

4.3. Privacy Protection. In order to protect the students' privacy, the homework and final examinations of students in the system will be encrypted before transmission. When sending homework and final examination message, the student will encrypt the message with the receiver's public key, and the receiver decrypts the message with his or her private key before reviewing the content of the homework and final examination.

Scenario: University C wants to verify the credit of Student A for Course B, then Student A sends C_{Work} which is generated by his homework and final examination M_{Work} and the public key Q_{UC} to University C. Supposing that an attacker wants to obtain the homework and final examination without the permission of Student A, attacker intercepts the data when Student A transmits the message.

Analysis: by intercepting the data, the attacker obtains C_{Work} which is displayed as a meaningless string and the attacker cannot get any valid information from C_{Work}. If the attacker wants to decrypt C_{Work}, he needs to use the private key of University C to calculate $M_{\text{Work}} = D_{d_{\text{UC}}}(C_{\text{Work}})$. However, d_{UC} is known only to University C and is not accessible to the attacker, and thus the attacker cannot get the content of the homework and final examination.

4.4. Decentralization and Distribution. The blockchain system has the characteristics of decentralization and distribution, and the data on the blockchain are not managed by a central organization. In the proposed system, universities jointly maintain the data on the blockchain, the entire ledger is stored on the peer nodes of each organization, and any user's operation on the blockchain in the system is executed synchronously on each peer node. Therefore, the data will remain consistent across the organization, and any two members of the system can reach a consensus on the data on the blockchain such as students' credits, course grades, and the hash values that students upload.

Scenario: a Peer node in the system fails and the data on the node is lost.

Analysis: the remaining peer nodes in the system can still operate normally for transaction endorsement and ledger updates. Since the entire ledger is stored on each peer node, the loss of one copy of the ledger will not affect the operation of the system.

4.5. Traceability. All transactions in a blockchain system are packaged into blocks and arranged in chronological order, so all operations of the system users on the data on the blockchain can be traced, and cheating can be eliminated by using the traceability of blockchain.

Scenario: supposing that Student A cannot finish Course B and fails the final examination, Teacher B wants to help Student A cheat by calling the chaincode AddGrade to give him a high score g'.

Analysis: University A wants to verify the credit of Student A for Course B, and gets the grade g' of Student A by invoking the chaincode CheckStudent. The administrator of University A will first review the content of the homework and final examination, finding that M_{Work} is unqualified and cannot meet g', then complain to the ministry of education. The education department queries all transactions related to Teacher B and Student A on the blockchain to find the block where Teacher B invoked the chaincode AddGrade, which contains the irregular operation of Teacher B and the timestamp, and can be used as evidence for prosecution.

Table 2: The computation cost analysis of the proposed scheme.

Phase role	User registration phase	Course registration phase	Course learning phase	Credit application phase	Credit verification phase
Administrator	$1T_{\text{Cmp}}$	N/A	N/A	N/A	N/A
CA	$1T_{\text{Mul}}$	N/A	N/A	N/A	N/A
University A	N/A	N/A	N/A	$1T_{\text{Cmp}} + 2T_{\text{Sig}}$	N/A
University B	N/A	$7T_{\text{Mul}} + 2T_H + 3T_{\text{Cmp}} + 2T_{\text{Sig}}$	N/A	N/A	N/A
University C	N/A	N/A	N/A	N/A	$7T_{\text{Mul}} + 3T_H + 4T_{\text{Cmp}} + 1T_{\text{Enc}} + 2T_{\text{Sig}}$
Teacher B	N/A	$7T_{\text{Mul}} + 2T_H + 3T_{\text{Cmp}} + 1T_{\text{Sig}}$	$8T_{\text{Mul}} + 2T_H + 6T_{\text{Cmp}} + 1T_{\text{Enc}} + 2T_{\text{Sig}}$	N/A	N/A
Student A	N/A	N/A	$6T_{\text{Mul}} + 3T_H + 1T_{\text{Enc}} + 3T_{\text{Sig}}$	N/A	$7T_{\text{Mul}} + 2T_H + 3T_{\text{Cmp}} + 1T_{\text{Enc}} + 1T_{\text{Sig}}$

T_{Mul}: multiplication operation T_{Enc}: asymmetric encryption. T_H: Hash function operation T_{Sig}: signature operation. T_{Cmp}: comparison operation.

Table 3: Communication cost analysis of the proposed scheme.

Item Phase	Message length (bits)	Round	3.5 G (14 Mbps)	4 (ms) G (100 Mbps)	5 (ms) G (20 Gbps)
User registration phase	640	3	0.046	0.006	0.032 us
Course registration phase	4272	2	0.305	0.043	0.214 us
Course learning phase	6320	2	0.451	0.063	0.316 us
Credit application phase	240	1	0.017	0.002	0.012 us
Credit verification phase	6400	2	0.457	0.064	0.320 us

Table 4: The comparison of the previous schemes and the proposed scheme.

Authors	Year	Objective	1	2	3	4	5
Turkanović et al. [9]	2018	Higher education credit platform	Y	N	N	N	N
Zhao et al. [10]	2020	System for student e-portfolio assessment	Y	N	Y	N	N
Mishra et al. [11]	2021	System for sharing students' credentials	Y	N	N	Y	Y
Jeong et al. [12]	2021	Multilateral personal portfolio authentication system	Y	N	Y	N	N
The proposed scheme	2021	A university course learning system with credit verifiable	Y	Y	Y	Y	Y

Notes: 1—propose an architecture or framework, 2—verifiable learning process, 3—no token consumption, 4—encrypt private information, 5—security analysis, Y—yes, N—no.

4.6. Credit Is Verifiable. In the proposed system, a university can verify the credits of students from other universities to determine whether the students' abilities meet the university's requirements. If student's homework and final examination are sent to the university, then it means that all teachers and administrators in the university can review the content of the homework and final examination.

Scenario: Student A obtains the credit for Course B and wants to transfer from University A to University C. To verify the learning situation of Student A on Course B, University C needs to review content of the homework and final examination.

Analysis: Student A sends C_{Work} which was generated by his homework and final examination M_{Work} and the public key Q_{UC} to University C, and University C decrypt C_{Work} with the private key d_{UC} to generate M_{Work}. For ensuring that M_{Work} generated by University C are the same as the homework and final examination submitted by Student A in the course learning phase, the administrator of University C invokes the chaincode CheckStudent to get the hash value H_{Work} of the homework and final examination which was uploaded by Student A and calculates $H'_{\text{Work}} = \text{hash}(M_{\text{Work}})$. If $H'_{\text{Work}} = H_{\text{Work}}$, then it means that M_{Work} sent by Student A were not tampered with, and the administrator reviews the content of the homework and final examination.

5. Discussion

5.1. Computation Cost Analysis. The computation cost analysis of the proposed scheme is shown in Table 2, and the highest computation cost is found in the course learning phase. Teacher B requires 8 multiplication operations, 2 hash function operations, 6 comparison operations, 1 asymmetric encryption operation, and 2 signature operations. Student A requires 6 multiplication operations, 3 hash function operations, 1 asymmetric encryption operation, and 3 signature operations. Thus, the proposed scheme has a good computational cost.

5.2. Communication Cost Analysis. Table 3 shows the communication efficiency of the proposed system. It is assumed that the ECDSA key and signature require 160 bits, course, homework, and final examination message and certificate require 1024 bits, and encrypted homework and exam message require 3072 bits, while other messages, like timestamp, identity information, request message, and result from the message, require 80 bits. Taking the credit verification phase, for example, it requires four ECDSA signatures, two certificates, an encrypted homework and exam message, and eight other messages. Thus, it requires $160 \times 4 + 1024 \times 2 + 3072 \times 1 + 80 \times 8 = 6400$ bits in total, which takes 0.457 ms under 3.5 G (14 Mbps) communication environment, 0.064 ms under 4G (100 Mbps) communication environment, and 0.320 us under 5G (20 Mbps) communication environment.

5.3. Comparison. Table 4 shows the comparison of the previous schemes and the proposed scheme. Compared to the related works, the proposed scheme focuses on proposing a university course learning system which has the advantages of verifiable learning process, no token consumption, protection of students' privacy and complete security analysis.

6. Conclusions

With the increase in the number of university online courses and students, the problem of credit verification becomes inevitable. Previously, credits were managed by each university and each online education platform alone, which led to the fact that credits earned by students in one university could not be recognized by other universities. This paper proposes a cross-university course learning system based on Hyperledger Fabric, which stores students' credits and hash values of the homework and final examinations on the blockchain, and the data on the blockchain are jointly maintained by all universities. For universities, they can

verify students' credits and review the content of students' homework and final examinations by invoking the chaincode, to determine whether students' abilities meet their requirements and then recognize the credits.

This paper shows a complete system architecture, details the application scenario, and provides the chaincode. To improve the security of the system, students' homework and final examinations are encrypted before transmission, thus effectively protecting students' privacy. At the same time, the Elliptic Curve Digital Signature Algorithm in Hyperledger Fabric can ensure data integrity during communication. The security analysis using BAN logic shows that our proposed system enables mutual authentication of the system members. Compared with previous systems based on blockchain technology, the universities in the proposed system recognize credits by reviewing students' homework and final examinations, rather than relying on the authority of central educational institutions. The proposed scheme uses consortium blockchain architecture, which improves system operating efficiency and saves the money needed for mining compared to public blockchain architecture. The final analysis shows that the proposed system also performs well in terms of computational cost and communication cost.

To sum up, this research achieved the following contributions:

(1) Proposes a cross-university course learning system based on Hyperledger Fabric where universities can review students' homework and final examinations to recognize students' credits.

(2) Proposes a complete system architecture, details the application scenario, and provides the chaincode.

(3) Uses Elliptic Curve Digital Signature Algorithm to ensure data integrity during communication and asymmetric encryption algorithm to protect students' privacy.

(4) Uses consortium blockchain architecture to improve system operating efficiency and save the money needed for mining compared to public blockchain architecture

(5) Presents security analysis through BAN logic to ensure mutual authentication of the system members

In the future, the research will consider adding the role of enterprises in the system to realize the verification of students' credits and learning records by enterprises, thus facilitating enterprises to understand the ability of students and select the students they need. At the same time, as the number of system members increases, how to ensure the privacy of students and system operating efficiency also need to be considered.

Notations

User X: Any university or teacher or student in the system
E: The elliptic curve
G: The origin generated on the elliptic curve E
Cert_X: A digital certificate of User X conforms to the X.509 standard
d_X: The private key of User X based on Elliptic Curve Cryptography
Q_X: The public key of User X based on Elliptic Curve Cryptography
ID_C: The identity of a course
ID_X: The identity of User X
TS_X: Timestamp of User X
M_{Request}: Request message sent by a user
M_{Result}: The resulting message of the request
M_{Identity}: Identity information message of a user
M_{Course}: Course message of a teacher
M_{Work}: Homework and final examination message of a student
g: The grade of a student for a course
k_i: The ith random number generated by the user
(r_i, s_i): The ith digital signature generated by the user
$\text{hash}(\cdot)$: One way hash function
H_i: The ith hash value generated by the user
$E_{Q_X}(M)$: Asymmetrically encrypt the message M with the public key Q_X
$D_{d_X}(M)$: Asymmetrically decrypt the message M with the private key d_X
C_{Work}: The cyphertext of homework and final examination message generated by asymmetric encryption
$A \stackrel{?}{=} B$: Verify whether A is equal to B.

Data Availability

The data supporting this study are available within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under contract MOST 110-2218-E-305-001–MBK and MOST 110-2410-H-324 -004 -MY2, and the Education and Teaching Reform Project of the Xiamen University of Technology (no. JG2021007).

References

[1] Q. Yang and Y. C. Lee, "The critical factors of student performance in MOOCs for sustainable education: a case of Chinese universities," *Sustainability*, vol. 13, no. 8089, 2021.

[2] "Udemy vs Coursera - which learning app is better," https://www.mobileappdaily.com/udemy-vs-coursera/amp.

[3] "Filipino Coursera learners complete over 140K courses in 10 months," 2021, https://mb.com.ph/2021/07/09/filipino-coursera-learners-complete-over-140k-courses-in-10-months-dost-chief/.

[4] "IIM-Kozhikode partners with Coursera to launch certificate programs," 2021, https://www.thehindu.com/education/colleges/iim-kozhikode-partners-with-coursera-to-launch-certificate-programmes/article35039579.ece.
[5] "European credit transfer and accumulation system (ECTS)," 2021, https://ec.europa.eu/education/resources-and-tools/european-credit-transfer-and-accumulation-system-ects_en.
[6] Credit Bank System(CBS), http://www.cb.or.kr/, 2021.
[7] S. Nakamoto, "Bitcoin: a peer-to-peer electronic cash system," 2021, https://bitcoin.org/bitcoin.pdf.
[8] A. Alammary, S. Alhazmi, M. Almasri, and S. Gillani, "Blockchain-based applications in education: a systematic review," *Applied Sciences*, vol. 92400 pages, 2019.
[9] C. L. Chen, M. L. Chiang, Y. Y. Deng, W. Weng, K. Wang, and C.-C. Liu, "A traceable firearm management system based on blockchain and IoT technology," *Symmetry*, vol. 13, no. 439, 2021.
[10] C. L. Chen, Y. Y. Deng, W. J. Tsaur, C.-Ta Li, C.-C. Lee, and C.-M. Wu, "A traceable online insurance claims system based on blockchain and smart contract technology," *Sustainability*, vol. 13, no. 9386, 2021.
[11] Y. C. Wang, C. L. Chen, and Y. Y. Deng, "Museum-authorization of digital rights: a sustainable and traceable cultural relics exhibition mechanism," *Sustainability*, vol. 13, no. 2046, 2021.
[12] C. L. Chen, C. Y. Lin, M. L. Chiang, Y.-Y. Deng, P. Chen, and Yi-J. Chiu, "A traceable online will system based on blockchain and smart contracttechnology," *Symmetry*, vol. 13, no. 6, 2021.
[13] M. Sharples and J. Domingue, "The blockchain and kudos: a distributed system for educational record, reputation and reward," in *Proceedings of the European Conference on Technology Enhanced Learning*, pp. 490–496, Springer, Cham, Midtown Manhattan, New York City, September 2016.
[14] M. Turkanovic, M. Holbl, K. Kosic, M. Hericko, and A. Kamisalic, "EduCTX: a blockchain-based higher education credit platform," *IEEE Access*, vol. 6, pp. 5112–5127, 2018.
[15] G. Zhao, H. He, B. Bi, Q. Xia, and Z. Fu, "A blockchain-based system for student e-portfolio assessment using smart contract," in *Proceedings of the 2020 4th International Conference on Computer Science and Artificial Intelligence*, pp. 34–40, ACM, Zhuhai China, 11 December 2020.
[16] R. A. Mishra, A. Kalla, A. Braeken, and M. Liyanage, "Privacy protected blockchain based architecture and implementation for sharing of students' credentials," *Information Processing & Management*, vol. 58102512 pages, 2021.
[17] J. Jeong, D. Kim, S.-Y. Ihm, Y. Lee, and Y. Son, "Multilateral personal portfolio authentication system based on hyperledger fabric," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–17, 2021.
[18] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.
[19] N. Szabo, "Smart contracts: building blocks for digital markets," *EXTROPY J. Transhumanist Thought*, vol. 18, no. 16, 1996.
[20] Y. C. Wang, C. L. Chen, and Y. Y. Deng, "Authorization mechanism based on blockchain technology for protecting museum-digital property rights," *Applied Sciences*, vol. 111085 pages, 2021.
[21] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the EuroSys '18: Proceedings of the Thirteenth EuroSys Conference*, pp. 1–15, ACM, Porto Portugal, 23 April 2018.
[22] N. Lu, Y. Zhang, W. Shi, S. Kumari, and K. K. R. Choo, "A secure and scalable data integrity auditing scheme based on hyperledger fabric," *Computers & Security*, vol. 92, Article ID 101741, 2020.
[23] M. Macdonald, L. Liu-Thorrold, and R. Julien, "The blockchain: a comparison of platforms and their uses beyond Bitcoin," *COMS4507 - Advanced Computer and Network Security*, vol. 54, 2017.
[24] P. Yabo, "Key metrics of blockchain platforms," https://docs.google.com/spreadsheets/d/1DQ770nGnHfJOoRSqTLmIkhuVK5CAbsFgqb6UoGMfVM/edit%23gid=0.
[25] T.-T. Kuo, H. Zavaleta Rojas, and L. Ohno-Machado, "Comparison of blockchain platforms: a systematic review and healthcare examples," *Journal of the American Medical Informatics Association*, vol. 26, no. 5, pp. 462–478, 2019.
[26] K. Toyoda, K. Machi, Y. Ohtake, and A. N. Zhang, "Function-level bottleneck analysis of private proof-of-authority ethereum blockchain," *IEEE Access*, vol. 8, pp. 141611–141621, 2020.
[27] "Open source, but private," https://www.corda.net/blog/open-source-but-private-a-case-for-private-decentralized-ledger-tech.
[28] "Build on Quorum, the complete open source blockchain platform for business," https://consensys.net/quorum.
[29] MultiChain for Developers: https://www.multichain.com/developers/.
[30] J. Polge, J. Robert, and Y. Le Traon, "Permissioned blockchain frameworks in the industry: a comparison," *ICT Express*, vol. 7, no. 2, pp. 229–233, 2021.
[31] "Blockchain still not enterprise ready, but the Hyperledger Fabric 1.0 release can show the way," https://www.hfsresearch.com/blockchain/blockchain-not-ready-but-hyperledger-fabric-release-show-the-way_071217.
[32] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.