Preserving Identity Leakage, Data Integrity and Data Privacy Using Blockchain in Education System

Ozgur Oksuz

Adiyaman University
Department of Computer Engineering
Adiyaman, TURKEY
ooksuz@adiyaman.edu.tr

Research Paper Received: 30.01.2022 Revised: 26.03.2022 Accepted: 03.04.2022

Abstract—Today, blockchain technology is evolving and has been used in many sectors such as healthcare, supply chain management, internet of things (IoT) and cryptocurrency exchange. Using this technology in these areas provides very good functionalities. A blockchain network is immutable, public, open, distributed, secure and reliable. This paper is about using blockchain technology in education system. Applying blockchain technology to education system brings all those properties above. However, user (student) data privacy and identity management of the entities in the system should be also considered. In this paper, the proposed scheme not only satisfies all these properties but also protects student data privacy and identity management of the entities when they communicate with each other. The proposed construction consists of encryption algorithms to protect students' private data and provide secure communication between the entities. Moreover, the proposed scheme does not leak students' identities to third parties in the blockchain.

Keywords—blockchain, data privacy, identity leakage, data integrity.

1. Introduction

Blockchain network was firstly introduced and used as a building block for Bitcoin [1]. Then, the importance of it has been evolving since it has very useful properties. A blockchain network is public (all transactions are publicly available), open (anyone can participate in the network), decentralized (it does not require a trusted party for authorization), distributed (each transaction's validity is performed many other nodes in the network), secure (if at least 51% honest nodes exists), and reliable (the network is up all the time). It is used as a building block in many systems such as healthcare [2], internet of things [3], and supply chain management [4].

In this paper, the blockchain technology is used in education system. With this technology many disadvantages in education system are eliminated. One of the disadvantages is to generate fake degree certificates [5]. Moreover, providing false information about academic achievements/transcripts is also

another problem when a student applies for an internship/summer job. Furthermore, storing each student's academic information/records for a long time can create another problem. These records can be lost due to human mistake or to have some natural disasters such as flooding. Putting all students' academic records into blockchain eliminates these kinds of disadvantages. However, other problems can arise. One of the problems is to protect student's data privacy. Data privacy says that any unauthorized entity cannot read the student's sensitive information such as student's real identity, course grades, taken courses, enrolled program, and owned certificates. Thus, data privacy should also be addressed using blockchain. Another issue that should also be considered is identity leakage. The students' identities should be hidden in the system. Otherwise, any untrusted third party can map transactions in blockchain to a student. Then, this untrusted party can do some statistical analysis to make a profile for the student. To address the above problems, this paper introduces a system that has the following properties:

- A brief mathematical formalization and construction of the system is given to protect sensitive data of the students, manage identity of the parties, and process of adding student records to the blockchain.
- The sensitive data (student's real identity, grade, course identity, program identity) of the students is stored in the blockchain and is encrypted. The data is only seen in the clear by the third parties when a student is transferred/enrolled to another university/program or applies to an internship. In this case, the leakage is only limited to the corresponding transactions based on the requirements of the position that the student applies to.
- The data of all transactions are immutable in the blockchain. Student's data integrity is preserved.

The organisation of the paper as follows: Section 2 presents related work. Section 3 introduces the definitions that are going to be used throughout the paper. In Section 4, the architecture and the transaction types are introduced in the blockchain. Then, the proposed scheme is presented in Section 5. In Section 6, security analysis of the scheme is given. In Section 7, we give the conclusion.

2. Related Work

There have been some studies that use blockchain technology in education system. These studies are [6], [7], [8], [9], [10], [11]. In [6] students' grades are not in the blockchain. The completed courses of the student are putting into the blockchain. However, the courses are still disclosed. This may rise a problem that student can be rule out if it is overqualified. Our scheme does not leak any sensitive information about student at the beginning. Our scheme leaks only the required courses based on the job title. The study in [8] uses two kinds of blockchains: private and public. Moreover, it focuses on privacy and integrity of the students' data. [9], [10] focus on putting students' certificates into the blockchain not the courses and grades of the students. In [11], the first university (the university of Nicosia) issues the academic certificates for students. The given certificate is verified using bitcoin blockchain. In [7], Sony Global Education develops a new blockchain for storing academic records. The work in [12] presented a blockchain-based (permissioned) repository for educational credentials but it does not have privacy protection of the students' private data. The work in [13] has the same problem that it does not introduce any protection mechanism for students' private data. The studies in [14], [15], [16], [17], [18] use permissionless blockchain in

education system. [15] does not have protection mechanism against student private data. Studies in [15], [16], [17], [18] use off-chain storage. It means that students' credentials are not stored in blockchain. They are stored in a file storage system.

Indy Hyperledger [19] is a public permissioned blockchain system that allows users to share their identities (based on credentials) anonymously. Moreover, this system is able to issue and revoke cryptographic credentials [20], [21]. Anyone with read access to the ledger can verify signatures made by issuers on credentials. In order to use this system (hyperledger Indy), the credentials that students own need to be known in advance. Every user can have different credentials. However, it could be very difficult to manage these credentials. In addition, it uses heavy cryptographic operations. Moreover, in this work, these kind of heavy cryptographic tools are not used. The students do not participate in the blockchain network and the blockchain does not contain the students' identities in the clear. This hides students' identities in the blockchain. However, once the student applies/communicates to a company for an internship or a permanent job, the student's real identity is disclosed to corresponding recruiter. The recruiter needs to know the student to communicate with the institution for this student to verify if the student has satisfied the internship/job requirements.

3. Definitions

3.1. Hash Function

A hash function is a mathematical algorithm that has the following properties: It maps an input x of any size to an output of fixed size y. This is shown as $H(x) = y$. It is deterministic algorithm, which means that if the algorithm retakes x, the output is always y. The algorithm is efficiently computed. If an output y is given, it is infeasible to compute x from y. This property is called as "One Way". It is infeasible to find another z that satisfies $H(x) = H(z) = y$. This is called as "Collision Resistant". A secure hash function algorithm should satisfy the all above properties. SHA-256 can be used to have all these properties.

3.2. Symmetric/Private Key Encryption

A symmetric encryption key scheme consists of 3 algorithms: Key generation ($KGen$), Encryption ($SEnc$), and Decryption ($SDec$). $KGen$ is the key generation algorithm takes a security parameter and outputs a key for user u: Secret key (SSk_u). Encryption algorithm takes a message m and the secret key (SSk_u), outputs the ciphertext $C = SEnc_{SSk_u}(m)$. Decryption algorithm takes the secret key SSk_u and C, outputs message $m = SDec_{SSk_u}(SEnc_{SSk_u}(m))$. A secure symmetric key encryption scheme should be resilient at least to Chosen Plaintext Attack (CPA). It means that even a message m is encrypted over and over again, the resulting ciphertext should be different each time. An example of CPA secure algorithm is AES (Advanced Encryption Standard) with CBC mode.

3.3. Asymmetric/Public Key Encryption

An asymmetric encryption protocol consists of 3 algorithms: Key generation ($KGen$), Encryption ($PEnc$), and Decryption ($PDec$). $KGen$ is the key generation algorithm takes a security parameter and outputs a key pair for user u: Public key PPk_u and Secret key PSk_u. Encryption algorithm takes a message m and public key PPk_u, outputs a ciphertext $C = PEnc_{PPk_u}(m)$. Decryption algorithm takes the secret key PSk_u and C, outputs message $m = PDec_{PSk_u}(PEnc_{PPk_u}(m))$. A secure asymmetric key encryption scheme also should be

resilient at least to Chosen Plaintext Attack (CPA). Examples of CPA secure schemes are RSA (not the textbook) [22], and Paillier encryption schemes [23].

3.4. Signature

A digital signature protocol consists of 3 algorithms: Key generation ($KGen$), $SIGN$, and $VERIFY$. $KGen$ is the key generation algorithm takes a security parameter, outputs a key pair for user u: Signature public key/verification key ($SignPk_u$) and Signature secret key ($SignSk_u$). $SIGN$ algorithm takes a message m and signature secret key ($SignSk_u$), outputs a signature $S = SIGN_{SignSk_u}(m)$. $VERIFY$ algorithm takes signature public key $SignPk_u$, signature S, and message m, outputs 1 ($VERIFY(m, S, SignPk_u) == 1$) if the signature is generated by user u under message m. If $VERIFY(m, S, SignPk_u) == 0$, then the signature under message m is not generated by user u.

A secure signature scheme needs to satisfy two properties: authenticity and integrity. Authenticity says that the owner of the signature convinces a verifier that the owner of the signature generates the signature using the message. Integrity says that signed data cannot be altered by any entity. An example of secure signature algorithm can be Schnorr signatures [24].

3.5. Blockchain

Blockchain is a technology that it has the following properties: It is a peer-to-peer decentralized system that nobody controls the system. It has the ledger technology that it keeps the information (transaction) up to date and it is immutable. Each node (computer) keeps a copy of this ledger. Since it is decentralized, other nodes check each transaction.

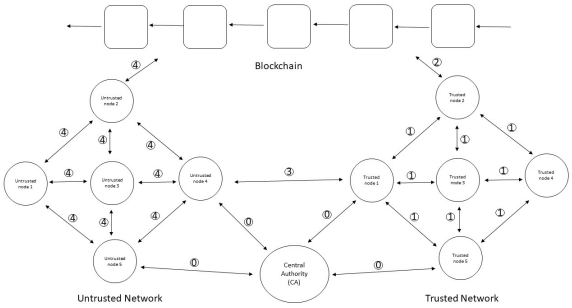

Figure 1. Network architecture of the proposed scheme.

If the transaction is intact, then it is added to the ledger. In the ledger, transactions are kept as blocks. It uses mathematical structures (hash functions and signatures) to verify each transaction. The first block is always named as genesis block. This is a special block.

4. Architecture of the Proposed Scheme

The proposed scheme consists of 4 different actors: a Central Authority (CA), Institutions (Ins), Students (Stu) and Recruiters (Rec). CA is responsible for setting up the system PKI (public key infrastructure). Moreover, CA also authorizes undergraduate courses and degrees. CA can be government that everyone can rely on. The institutions are the places that students take courses and get degrees. Institutions can be schools, universities. Students take courses from schools/institutions. Recruiters are the employers that wants to hire qualified students. Moreover, the recruiters should also be convienced that the students satisfy the requirements of the positions.

The network architecture of the proposed scheme is shown in Fig.1. In the system, there are two kinds of nodes: trusted and untrusted. The trusted nodes can only issue transactions into a ledger. After a block is formed, the block is published and being

public.

The work flow of the system is the following:

0. Each node (trusted or untrusted) registers its public key to CA. Then, CA authorizes each node's public key. Moreover, CA makes these public keys public. Thus, everyone in the system knows the identities of others.
1. After grading a student's exam, an institution sends the student's grade as a transaction to other trusted nodes.
2. If most of the nodes (more than 50%) verify that the transaction is correctly formed, it is put into the blockchain as a block by every trusted node.
3. If a Stu is transferred to another institution or applying on a job, an Ins or a Rec asks the current institution to hand over some tokens to retrieve the student information from the blockchain.
4. Once an Ins or a Rec gets corresponding tokens from the student's current institution, the Ins or the Rec retrieves the corresponding student's information (as a transaction) from the blockchain.

4.1. Threat Model

In this paper, CA and Institutions are trusted entities. Recruiters and students are untrusted entities. Recruiters and students are malicious adversaries that they can provide fake or altered information to institutions. Recruiters try to learn private data of students in the blockchain. Recruiters can ask institutions for decryption keys to learn specific students' private data in the blockchain. Moreover, recruiters can change data that comes from student A. Then, it asks institution for decryption keys to learn private data of student B. Student A can also pretend to be student B when applying a job or an internship. Students can present fake information about their academic history when they apply a job or an internship.

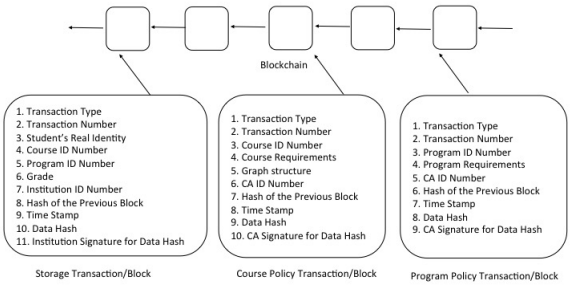

Figure 2. An illustration of storage, course policy and program policy blocks/transactions.

We assume that malicious adversaries can not collude with each other. It means that they can not share specific information with each other. This information can be their real identities, public keys (pseudonym identities) and their secret keys. Moreover, any student can not collude with any recruiter.

4.2. Transactions

In this section, transactions (storage, course policy and program policy) of the proposed scheme as blocks are going to be presented. In other words, each transaction is going to be a block. In Fig.2, there are three types of transactions. Multiple transaction types in a blockchain are not new. These are used in other studies such as in [2]. While institutions issue storage transactions, course policy transactions and program policy transactions are only issued by CA.

A storage block consists of the following information:

S1. Transaction Type (TT): This is type 00 which is storage transaction.
S2. Transaction Number (TN)

S3. Student's Real Identity (SRI): This identity consists of student's national identity number, full name, date of birth, and gender.
S4. Course ID Number (CID): This number is going to be course number that a student takes.
S5. Program ID Number (PID): The number of the program that the student enrols.
S6. Grade (G): This is a number that is going to be between 0 and 100.
S7. Institution ID Number (IID): This is the signature public key of the institution.
S8. Hash of the Previous Block (HPB): This is for immutability for the blockchain. The hash of the previous block is added. This is done by an Ins.
S9. Time Stamp (TS): Time that the transaction occurs. This is done by an Ins.
S10. Data Hash (DH): Hash of all the data (step S1-S9). This is done by an Ins.
S11. Institution Signature for Data Hash ($InsSIGN$): This is for data integrity. The institution signs the data in step S10.

A course policy transaction/block consists of the following information:

CP1. Transaction Type (TT): This is type 01 which is a course policy transaction.
CP2. Transaction Number (TN)
CP3. Course ID Number (CID): This number is going to be course number that a student takes.
CP4. Course Requirements (CR): It gives course information such as number of credits, course syllabus and minimum score to pass the course.
CP5. Graph Structure (GS): It shows which other programs offer this course.
CP6. CA ID Number ($CAID$): This is the signature public key of CA.
CP7. Hash of the Previous Block (HPB): This is for immutability for the blockchain. The hash of the previous block is added. This is done by CA.
CP8. Time Stamp (TS): Time that the transaction occurs. This is done by CA.
CP9. Data Hash (DH): Hash of all the data (step CP1-CP8). This is done by the CA.
CP10. CA Signature for Data Hash ($CASIGN$): This is for data integrity. CA signs the data in step CP9.

In step CP5, this structure is important when a student decides to enrol another program of the same university or a different university. The graph shows if the course is offered by different programs. Thus, the student does not need to retake the course. This provides minimum of difficulty when student change its major. As an example, introduction to computer programming course is offered to many other programs such as Mathematics, Computer Science, Civil Engineering and so on. If a student decides to change its major from Mathematics to Computer Science, the students only need to transfer its grade. It does not need to retake the same course (introduction to computer programming). An illustration of a graph structure is shown in Fig.3. In the figure, Mathematics program is represented by 146729, Computer Science program is represented by 190834, Environmental Engineering program is represented by 146892, Electric Engineering program is represented by 154266, and Introduction to the Programming course is represented by 034783.

A program policy transaction/block consists of the following information:

PP1. Transaction Type (TT): This is type 10 which is a program policy transaction.
PP2. Transaction Number (TN)
PP3. Program ID Number (PID): This number is going to be program number that a student enrols.
PP4. Program Requirements (PR): It includes total credits that student needs to take to complete

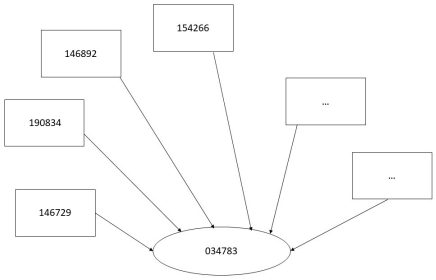

Figure 3. An illustration of a course-program relation graph.

the program, program information, program courses and their numbers.

PP5. CA ID Number ($CAID$): This is the signature public key of CA.

PP6. Hash of the Previous Block (HPB): This is for immutability for the blockchain. The hash of the previous block is added. This is done by CA.

PP7. Time Stamp (TS): Time that the transaction occurs. This is done by CA.

PP8. Data Hash (DH): Hash of all the data (step PP1-PP7). This is done by the CA.

PP9. CA Signature for Data Hash ($CASIGN$): This is for data integrity. CA signs the data in step PP8.

5. Construction

The proposed protocol uses blockchain network. There are two kinds of blockchain network that are open to public: Permissionless and Permissioned. In a permissionless blockchain, anyone can join and leave the network. When a participant joins the network, it can have the copy of entire blockchain and operate as a full node. Anyone can validate the transactions publicly. All data in the blockchain is available to everyone. There is no trust between the participants. Moreover, a participant does not need to be authorized to join the blockchain network. Permissionless blockchain network is used in Bitcoin [1] and Etherium [25]. In a permissioned blockchain, not anyone can join the network. Permission is provided to certain identifiable participants to join the network. This requirement adds an additional level of security. Participants are known to each other. The trusted parties only write to the ledger. Permissioned blockchain is also used in many protocols since permissioned blockchain has an additional level of security. Moreover, permissioned blockchain is more efficient and more scalable. Since in our protocol, the entities are known in advance we use permissioned blockchain network. CA and Ins are allowed to write data to the ledger. The scheme enjoys with the following properties:

- The sensitive data (student's real identity, grade, course identity, program identity) of the students are encrypted in the blockchain. Moreover, there is no information about identitites of the students in the clear. This information is hidden. The data is only seen by the third parties in the clear when a student is transferred/enrolled to another university/program or or applying to an internship. The only leakage is based on the required job position when the student applies for or courses that need to be transferred when the student is transferred to another institution.
- The data integrity is preserved and cannot be changed. The data of all transactions are immutable in the blockchain ledger.

The proposed protocol consists of 4 algorithms: ***Setup***, $Register, Transaction$, and $Key\ Transfer\ for\ Decryption$.

Setup: In this phase, CA decides which signature ($SIGN$), public/symmetric key encryption algorithms ($PEnc/SEnc$) and their appropriate security parameters will be used in the system. Once the

algorithms are announced to the public, each node (trusted and untrusted) and CA chooses their secret key/public key pair for signature algorithm and chose their secret/public key pair for public key encryption algorithm. CA's secret key and public key pair for signature algorithm are represented as $SignSk_{CA}$ and $SignPk_{CA}$. A trusted node i's secret key and public key pair for signature algorithm are represented as $SignSk_{i,t}$ and $SignPk_{i,t}$. An untrusted node j's secret key and public key pair for signature algorithm are represented as $SignSk_{j,ut}$ and $SignPk_{j,ut}$.

Moreover, CA's secret key and public key pair for public key encryption algorithm are represented as PSk_{CA} and PPk_{CA}. A trusted node i's secret key and public key pair for public key encryption algorithm are represented as $PSk_{i,t}$ and $PPk_{i,t}$. An untrusted node j's secret key and public key pair for public key algorithm are represented as $PSk_{j,ut}$ and $PPk_{j,ut}$.

Furthermore, each student Stu_q also chooses secret key and public key pair for signature algorithm as $SignSk_{Stu_q}, SignPk_{Stu_q}$ and chooses public/secret key as PSk_{Stu_q}, PPk_{Stu_q} for asymmetric public key encryption algorithm. A note that all secret keys are chosen randomly for security.

Register: In this phase, each node's (trusted or untrusted) public encryption key PPk_i and signature public key $SignPk_i$ for node i are going to be registered to the system by CA. In the system, each public key (for signature and encryption) will be identity of a party (trusted node, untrusted node, CA). A standard Public Key Infrastructure (PKI) scheme can be used to register entities public keys. CA also acts as certificate authority that is responsible for identities of the parties in the blockchain network. After the parties' identities have been verified, CA generates digital certificate under the parties' public keys. When the entities communicate with each other, they show their valid certificates – that are digitally signed by CA's private key- to each other.

Moreover, each student Stu_q registers its signature public key $SignPk_{stu_q}$ and public encryption key PPk_{Stu_q} to its corresponding institution Ins_j. This can be done when student enrols a program in that institution.

Transaction: Three cryptographic primitives are used for storage transaction phase: Symmetric Encryption ($SEnc$), Hash (H) and Signature ($SIGN$). Once student's exam is graded, institution j (Ins_j) encrypts its grade G, corresponding real identity (SRI) of the student, corresponding course identity number CID and corresponding program identity number PID. We use the following notations: G_{Stu_q} as the grade of student q, $SEnc_k(G_{Stu_q})$ as the encryption of the grade G of student q, $SEnc_{k'}(CID)$ as the encryption of the course identity number CID, $SEnc_{k''}(PID)$ as the encryption of the program identity number PID and $SEnc_{k'''}(SRI)$ as the encryption of the student's real identity. Here k, k', k'', k''' are four different secret private encryption keys chosen randomly by Ins_j for $SEnc$ algorithm. Moreover, Ins_j uses hash algorithm H to hash the following data

$$DH = H(TT||TN||SEnc_{k'''}(SRI)||SEnc_{k'}(CID)||$$
$$SEnc_{k''}(PID)||SEnc_k(G_{stu_q})||SignPk_j||HPB_l||TS),$$

where $\|$ is the concatenation symbol.

In the hash data, institution signature public key $SignPk_j$ is chosen as the institution identity number. Moreover, HPB_l is chosen as hash of the previous block. We assume that current block is $l+1$. We also assume that SRI is only known to student itself and the institution. This information is never seen in public. Only authorized people (old and new institution, recruiter when student applies a job or an internship, and CA) can know

this information. Finally, Ins_j signs DH value as $SIGN_{SignSk_j}(DH)$ and generate the transaction that consists of the followings:

$S1 = TT$, $S2 = TN$, $S3 = SEnc_{k'''}(SRI)$, $S4 = SEnc_{k'}(CID)$, $S5 = SEnc_{k''}(PID)$, $S6 = SEnc_k(G_{stu_q})$, $S7 = SignPk_j$, $S8 = HPB_l$, $S9 = TS$, $S10 = DH$, $S11 = SIGN_{SignSk_j}(DH)$.

TT is set to 00 since it is a storage transaction. Once the transaction is generated by Ins_j, Ins_j sends this transaction to the other trusted nodes. Other trusted nodes check the signature and data if they pass from verification of the signature. If more than 50% of the trusted nodes respond that the signature is valid, this transaction is added to the ledger.

As a note that noone can see the student's real identity, grade of the student, the student's course information and program information in the clear in the transactions since these are all encrypted. Thus any unauthorized party cannot see the plaintexts of them. Since SRI is encrypted so any unauthorized entity cannot map this encrypted identity to a real person. Furthermore, there is no information about student in the clear in the blockchain. So the identities of the students are hidden.

A course policy transaction phase consists of $CP1 = TT$, $CP2 = TN$, $CP3 = CID$, $CP4 = CR$, $CP5 = GS$, $CP6 = CAID$, $CP7 = HPB_t$, $CP8 = TS$, $CP9 = DH$, $CP10 = SIGN_{SignSk_{CA}}(DH)$.

In the transaction, TT is set to 01 since it is a course policy transaction. We assume that the current (this case) block is $t + 1$. This case DH is

$$DH = H(TT||TN||CID||CR||GS||CAID||HPB_t||TS).$$

A program policy transaction phase consists of $PP1 = TT$, $PP2 = TN$, $PP3 = PID$, $PP4 = PR$, $PP5 = CAID$, $PP6 = HPB_f$, $PP7 = TS$, $PP8 = DH$, $PP9 = SIGN_{SignSk_{CA}}(DH)$.

In the transaction, TT is set to 10 since it is a program policy transaction. We assume that the current (this case) block is $f + 1$. This case DH is

$$DH = H(TT||TN||PID||PR||CAID||HPB_f||TS).$$

As a note that course policy transactions and program policy transactions are done by CA so these transactions can be done before any storage transactions.

Key Transfer for Decryption: In this phase happens when a student is transferred to another institution or student applies a job or an internship. An illustration of the message flow diagram is given in Fig. 4. The student sends a message tuple to its current institution when it applies to an internship or a job (S3). Assuming that Stu_q's current institution is Ins_j and applying for an internship in Rec_m. The student sends the same message tuple to the recruiter by mentioning its current institution and its identities (S4).

Since the blockchain is public, recruiter Rec_m can see all the transactions but cannot see the private student information. This is because all sensitive information about the student is encrypted. In order to see sensitive student information, Rec_m needs to know the encryption keys to decrypt the corresponding ciphertexts. Moreover, the corresponding transaction numbers needs to be known by Rec_m since the blockchain data does not contain any student identity in the clear. Thus Rec_m asks the student's institution Ins_j to send the corresponding transaction keys and the corresponding transaction numbers (R2). These keys are needed for decrypting the student's sensitive information (student's real identity, grades, course identities and program identity) that they are required for the position that student applies. For example, if the student

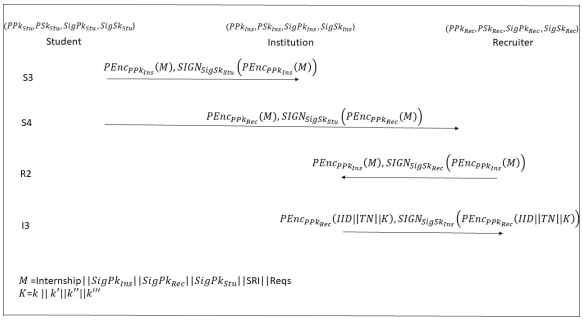

Figure 4. Message Flow Diagram between Parties.

wants to be an intern in a software company. The requirements of the internship for the student are to take some programming language courses. Thus, the student needs to prove that it has taken some programming language courses. Some projects can also be given related to software used by the student during the projects. This information is all encrypted so Ins_j needs to send all decryption keys with the transaction numbers to Rec_m (I3).

Student

S1. Stu_q generates a tuple, $(App||InstitutionInfo||RecruiterInfo||StudentInfo1||StudentInfo2||Reqs)$, where $App \in \{Job, Internship, Transfer\}$ shows the purpose of the application. $InstitutionInfo$ is the current institution information which is $SignPk_{Ins_j}$ (signature verification key of the institution), $RecruiterInfo$ is the recruiter information which is $SignPk_{Rec_m}$ (signature verification key of the recruiter), $StudentInfo1$ is the student's pseudonym identity which is $SID = SignPk_{Stu_q}$ (signature verification key of the student), $StudentInfo2$ is the student's real identity which is SRI, and $Reqs$ is the requirements of the position of a job/an internship. $Reqs$ consists of the requirements of the position, deadline of the application and the signature of $Reqs$ that it is signed by Rec_m. $RecruiterInfo$ can be the new institution information that student is transferred to. We assume that student applies to an internship position. The student prepares the tuple $M = (Internship||SignPk_{Ins_j}||SignPk_{Rec_m}||SignPk_{Stu_q}||SRI||Reqs)$.

S2. Stu_q computes $PEnc_{PPk_{Ins_j}}(M)$ and signs it with its signature secret key $SIGN_{SignSk_{Stu_q}}(PEnc_{PPk_{Ins_j}}(M))$,

S3. Stu_q sends $PEnc_{PPk_{Ins_j}}(M)$ and $SIGN_{SignSk_{Stu_q}}(PEnc_{PPk_{Ins_j}}(M))$ to Ins_j. This is for letting Ins_j know that Stu_q wants to apply an internship. Moreover, the recruiter is Rec_m.

S4. Stu_q also computes $PEnc_{PPk_{Rec_m}}(M)$ and $SIGN_{SignSk_{Stu_q}}(PEnc_{PPk_{Rec_m}}(M))$. Then Stu_q sends them to Rec_m. With this information, Stu_q applies to the internship position.

Recruiter

R1. Rec_m first decrypts $PEnc_{PPk_{Rec_m}}(M)$ in step-S4 as $Dec_{PSk_{Rec_m}}(PEnc_{PPk_{Rec_m}}(M))$ and gets the tuple (M). Thus, Rec_m knows Stu_q's real identity SRI, pseudonym identity $SignPk_{Stu_q}$ and student's current institution information. Rec_m also checks if the signature is a valid signature.

R2. Rec_m knows who is applying to the position. If the signature is valid from step-R1, Rec_m computes $PEnc_{PPk_{Ins_j}}(M)$ and $SIGN_{SignSk_{Rec_m}}(PEnc_{PPk_{Ins_j}}(M))$. Then it sends them to Ins_j.

Institution

I1. Ins_j firstly decrypts the ciphertext in step-S3 and gets M. Then, it checks/verifies the signature using $SignPk_{Stu_q}$ that if the encrypted message (in step-S3) is generated by Stu_q.

If the signature is valid, Ins_j learns what Stu_q wants to do (internship/job application or transfer to another university) and where Stu_q wants to apply (who is the recruiter) and what the requirements are for the position that Stu_q applies ($Reqs$). These are all learned from M. Ins_j also checks if SRI and SID belong to same student.

I2. Next, Ins_j decrypts the ciphertext in step-R2 and gets M. Then, it checks/verifies the signature that if the encrypted message (in step-R2) is generated by Rec_m.

I3. If the tuples from step-I1 and step-I2 are the same and all the checking/verification steps in I1 and I2 are successfully completed, Ins_j sends transaction numbers (TN) and the secret keys to decrypt ciphertexts (encrypted student's private information) in the transactions to Rec_m.

Once Rec_m has the secret keys, it can decrypt the sensitive data required for the position. For example, if Ins_j sends $PEnc_{PPk_{Rec_m}}(IID||TN||k'''||k'||k''||k)$ and $SIGN_{SignSk_{Ins_j}}(PEnc_{PPk_{Rec_m}}(IID||TN||k'''||k'||k''||k))$ to Rec_m. Rec_m decrypts the ciphertext as $PDec_{PSk_{Rec_m}}(PEnc_{PPk_{Rec_m}}(IID||TN||k'''||k'||k''||k)) = IID||TN||k'''||k'||k''||k$. It gets tuple $(IID||TN||k'''||k'||k''||k)$. Then, it checks if the signature is valid using verification key $IID = SignPk_{Ins_j}$. If the signature is valid, it decrypts the ciphertexts in the transaction using the keys k, k', k'', k''' to retrieve the student's information.

After decrypting all the sensitive information, the recruiter sees the student's grades, course identities, program identity, real identity. Moreover, once the recruiter gets course identity, it can look for the course identity in the course policy transaction block in the blockchain for getting information about the course. In the block, the recruiter simply retrieves CP4 (course requirements). This allows the recruiter to see what topics have been covered in this course.

5.1. Discussions

When a student wants to be transferred another institution and wants to study the same program, the new institution requires all the course history of the student. This case old university needs to send all the secret keys of the corresponding ciphertexts with all the corresponding transaction numbers. When a student applies for a job that it requires completing the program of study, the recruiter (employer) also needs to have all course history of the student with the transaction numbers. This is because whether the student has completed the all the requirements of the program for a degree. For this case, the recruiter needs to retrieve program policy transaction in the blockchain for verification.

A student can provide fake academic history via its CV/Resume to the recruiter when the student applies for the position. This can be done by the student updating the message tuple as follows: $(Internship||SignPk_{Ins_j}||SignPk_{Rec_m}||SignPk_{Stu_q}||SRI||CV/Resume||Reqs)$. Since all the academic history of the student is stored in the blockchain and it is immutable, student does not get any advantage by presenting fake results (CV) to the recruiter. The recruiter needs to communicate with the institution for the student's real academic history for the requirements ($Reqs$) of the position. Thus, the recruiter can easily verify student's CV this case. Since the student can not get any advantage by faking its academic history, we don't add the student's CV to the tuple in the real protocol.

As a note that students can study multiple programs, masters and PhDs. Moreover, students can enroll multiple programs and have multiple degrees,

and have their certificates. Putting these certificates as pictures and all other information to the blockchain can be problematic. This is because each node in the blockchain network has limited source capabilities such as storage. This case the values should be stored in other sources (off-chain) such as in a cloud [2] or in a central database [26]. For these cases, the encrypted private data of the students are stored in a database, the addresses of these data are stored in the blockchain. After the student applies for an internship, the institution not only sends the decryption keys but also sends the transaction numbers to the recruiter. Once the recruiter gets the transaction numbers, it looks for the transaction numbers in the blockchain to get the addresses of the encrypted student's data. Then, the recruiter retrieves the encrypted student's data from the database using the addresses. Using the decryption keys, the recruiter gets the plaintexts of the data.

6. Security Analysis

User/student Data Privacy: User/student's sensitive data consists of Student's grade, Student's real identity, Course identity number and Program identity number. These data are very sensitive. Leaking this information in the clear results breaching of student's data privacy. However, any unauthorized entity cannot see these data in the clear. This information is encrypted using secure symmetric key encryption scheme such as AES with CBC mode. $AES-CBC$ mode consists of an initialization vector (IV) as a random number in the algorithm that even the same message is encrypted more than once, the resulting ciphertext is going to be different each time. Thus the proposed scheme provides data privacy of the student. In the case when the student is transferred to another institution to study different program or when the student wants to do internship in a company (recruiter), the leakage is limited that only the required decryption keys of the transactions and corresponding secret keys are sent to the recruiter. Moreover, the decryption keys are not sent in the clear, they are sent in the encrypted form. Thus, only the authorized recruiter can recover the keys by decryption. Then, the recruiter can recover the student's real identity, grade, program identity and course identity by using those keys.

Data Integrity and Data Immutability: Data integrity is observed by using signature algorithm in the proposed scheme. With this algorithm the data cannot be altered by any entity. The data immutability is preserved in the construction by using blockchain technology. With this technology each block (transaction) is tied each other with the help of signature and hash algorithms. Thus, no one can break this chain to change data. Moreover, a malicious recruiter can not gain any useful information by changing student A's message to student B's message. The malicious recruiter can do this to get decryption keys to learn student B's private data. This is because institution needs to have the appropriate message from student B too. In other words, steps S3 and R2 should be consistent. A student can not pretend being another student when it generates a message since the student needs to know another student's real identity and pseudonym identity and signature secret key. The student can not provide false information about its academic history since all the academic history is stored in the blockchain by the trusted parties (institutions).

7. Conclusion

In this paper, a protocol is introduced for an education system using blockchain technology that provides student data privacy, data integrity, and immutability of the data. The proposed construction

uses encryption algorithms (symmetric and asymmetric) to protect data privacy, uses signatures for data integrity and uses blockchain for immutability. Moreover, the proposed scheme does not leak students' real identities in the blockchain since the students do not need to participate in the blockchain network. In addition, all the sensitive data is encrypted. The proposed protocol models communication between the parties and proposes identity management. This paper also proposes course-program graph for the scheme to provide minimum of difficulty when the student change its major.

Acknowledgments

The author thanks the anonymous reviewers for their useful comments and suggestions.

References

[1] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Accessed April 4, 2022. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[2] L. Hirtan, P. Krawiec, C. Dobre, and J. M. Batalla, "Blockchain-based approach for e-health data access management with privacy protection," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. Limassol, Cyprus: IEEE, 11-13 September 2019, pp. 1–7.

[3] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[4] D. Salah, M. H. Ahmed, and K. ElDahshan, "Blockchain applications in human resources management: Opportunities and challenges," in *Proceedings of the Evaluation and Assessment in Software Engineering*. Trondheim, Norway: Association for Computing Machinery, New York NY, United States, 15-17 April 2020, pp. 383–389.

[5] M. C. H. Clifton and S. Cox. (2018, January) 'staggering' trade in fake degrees revealed. Accessed April 4, 2022. BBC News. [Online]. Available: https://www.bbc.com/news/uk-42579634

[6] L. M. Palma, M. A. Vigil, F. L. Pereira, and J. E. Martina, "Blockchain and smart contracts for higher education registry in brazil," *International Journal of Network Management*, vol. 29, no. 3, p. e2061, May 2019.

[7] S. G. Education, "Sony global education develops technology using blockchain for open sharing of academic proficiency and progress records," Tech. Rep., 2016.

[8] K. Kuvshinov, I. Nikiforov, J. Mostovoy, D. Mukhutdinov, K. Andreev, and V. Podtelkin, "Disciplina: Blockchain for education," Yellow Paper, Tech. Rep., 2018.

[9] R. Arenas and P. Fernandez, "Credenceledger: A permissioned blockchain for verifiable academic credentials," in *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Stuttgart, Germany, 17-20 June 2018, pp. 1–6.

[10] Blockcerts, the open standard for blockchain certificates. http://www.blockcerts.org/. Accessed April 5, 2022. [Online]. Available: http://www.blockcerts.org/

[11] University of nicosia. blockchain certificates (academic and others). https://www.unic.ac.cy/iff/blockchain-certificates. Accessed April 5, 2022. [Online]. Available: https://www.unic.ac.cy/iff/blockchain-certificates

[12] E. E. Bessa and J. S. Martins, "A blockchain-based educational record repository," 2019, arXiv preprint arXiv:1904.00315.

[13] M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record, reputation and reward," in *Adaptive and Adaptable Learning*, K. Verbert, M. Sharples, and T. Klobučar, Eds. Cham: Springer International Publishing, 2016, pp. 490–496.

[14] M. Han, Z. Li, J. S. He, D. Wu, Y. Xie, and A. Baba, "A novel blockchain-based education records verification solution," in *Proceedings of the 19th Annual SIG Conference on Information Technology Education*, ser. SIGITE '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 178–183. [Online]. Available: https://doi.org/10.1145/3241815.3241870

[15] M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, "Eductx: A blockchain-based higher education credit platform," *IEEE Access*, vol. 6, pp. 5112–5127, 2018.

[16] W. Gräther, S. Kolvenbach, R. Ruland, J. Schütte, C. Torres, and F. Wendland, "Blockchain for education: lifelong learning passport," in *Proceedings of 1st ERCIM Blockchain workshop 2018, 2018: European Society for Socially Embedded Technologies (EUSSET)*. Amsterdam, Netherlands: European Society for Socially Embedded Technologies (EUSSET), 2-8 May 2018, pp. 1–8.

[17] P. Ocheja, B. Flanagan, H. Ueda, and H. Ogata, "Managing lifelong learning records through blockchain," *Research and Practice in Technology Enhanced Learning*, vol. 14, no. 1, p. 4, 2019. [Online]. Available: https://doi.org/10.1186/s41039-019-0097-0

[18] R. A. Mishra, A. Kalla, A. Braeken, and M. Liyanage, "Privacy protected blockchain based architecture and implementation for sharing of students' credentials," *Information Processing and Management*, vol. 58, no. 3, p. 102512, 2021.

[19] M. Lodder and B. Zundel. Hyperledger indy hipe. Accessed April 4, 2022. [Online]. Avail-

able: https://hyperledger-indy.readthedocs.io/projects/hipe/en/latest/text/0109-anoncreds-protocol/README.html

[20] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Advances in Cryptology — EUROCRYPT 2001*, B. Pfitzmann, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 93–118.

[21] J. Camenisch and A. Lysyanskaya, "A signature scheme with efficient protocols," in *Security in Communication Networks*, S. Cimato, G. Persiano, and C. Galdi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 268–289.

[22] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[23] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*, J. Stern, Ed. Prague, Czech Republic: Springer Berlin Heidelberg, 2-6 May 1999, pp. 223–238.

[24] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[25] Ethereum. https://www.ethereum.org. Accessed April 5, 2022. [Online]. Available: https://www.ethereum.org

[26] M. Hanley and H. Tewari, "Managing lifetime healthcare data on the blockchain," in *IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Guangzhou, China, 8-12 October 2018, pp. 246–251.