Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3558972, 11 pages
https://doi.org/10.1155/2021/3558972

Research Article

Blockchain-Based Privacy Protection Scheme for IoT-Assisted Educational Big Data Management

Xiaoshuang He,[1] **Hechuan Guo**,[2] **and Xueyu Cheng**[3]

[1]*School of Education, Tianjin University, Tianjin, China*
[2]*School of Computer Science and Technology, Shandong University, Qingdao, China*
[3]*Rizhao Lanshan Experimental Middle School, Rizhao, China*

Correspondence should be addressed to Hechuan Guo; ghc@mail.sdu.edu.cn

Received 16 June 2021; Accepted 21 July 2021; Published 15 August 2021

Academic Editor: Zhuojun Duan

Copyright © 2021 Xiaoshuang He et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Adoption of the Internet of Things (IoT) in education brings many benefits. However, the poor implementation of access control of educational data produced by the IoT devices has brought students' and teachers' privacy into danger. Attackers can access educational data that they are not permitted to access and even erase the records during access. To tackle this problem, we employ blockchain technology to guarantee the integrity of access control rules and trace the records of access events. In this paper, we propose a blockchain-based access control scheme for the data produced by IoT devices. The scheme consists of three components: (1) a well-implemented data collection module that is deployed in smart classrooms, which collects and uploads data about the real-time situation inside the smart classroom to the data center; (2) a MongoDB-based data center and its control module that makes access control decisions based on the verification of the permissions of visitors, where the permissions are managed by blockchain; and (3) a customized blockchain system that stores and keeps security policy updates of the role-based access control module and records access events in a trusted way. Our analysis indicates that the proposed access control scheme guarantees the correctness of the access control process and makes the access of collected educational data auditable and responsible. Our system collectively analyzes the context of the smart classroom and is capable of detecting multiple scenarios such as absence, lateness, and gunshot. We show how the scheme preserves students' and teachers' privacy by carrying out extensive experimental studies. The results indicate that the proposed data management system can give correct responses as quickly as a traditional data server does while preserving privacy.

1. Introduction

With the rapid development of Internet of Things (IoT), cities around the world are becoming smarter and smarter. One of the most widespread application scenarios of smart city is smart education, where educational big data is collected through multiple IoT devices deployed in smart campuses and smart classrooms and stored for a variety of data processing and analysis tasks.

Adoption of IoT in education has been widely studied. Marquez et al. [1] proposed a model to integrate objects to Virtual Academic Communities (VAC). Their results indicate that the adoption of IoT yields a more engaging learning environment for learners, and the instructors can obtain more information about the learning process, which in turn enhances the pedagogical process. Moreira et al. [2] conducted a study to provide personalized education to learners by using the data collected through IoT, cloud computing, and learning analytical tools. It is indicated that this approach is able to provide personalized curricula that depend on the abilities of each student. Last but not least, Bagheri and Movahed [3] showed that the use of IoT in education is not limited to teaching and learning. Their study indicated that IoT in education can be used to (1) manage energy and monitor ecosystem; (2) implement secure campus and classroom access control; and (3) monitor student's health. In one word, adoption of IoT in education brings many benefits.

However, the access of the educational data produced during the work flow of these applications is not carefully controlled. Particularly, the privacy of the involved teachers

and students is in danger of being violated. There exist many instances demonstrating the severity. Here are a couple of examples. InBloom was a nonprofit educational technology company, which developed educational technology products to provide students with personalized learning services. But inBloom survived only 15 months. The main reason lies in that the information collected by the company involves too much privacy of students, and the company shared these data with other companies. Eventually, public protests and pressure from public opinion caused the company to apologize and shut down [4]. In September 2016, a high school student in Tianjin broadcasted the scenes of her classmates' learning, breaks, outdoor activities, etc., on a live broadcast platform, without the attention of her classmates. There were hundreds of people viewing the live broadcast, and some of them posted explicit information and messages, which include the personal information of the students [5].

As more and more schools are in progress of having smart campuses and smart classrooms, more and more IoT devices are used by students and teachers to interact. The involved privacy problems brought by IoT urge to be solved, which can be summarized as follows:

(1) The uses of the sensors and the data produced by the sensors are unlimited. Access control schemes of the educational data are not well implemented. Attackers can cross the access restrictions by tampering with the access rules using methods such as SQL injection

(2) The access of the data is not auditable. Attackers can erase the records of their visits using simple methods

To address these issues, we propose a blockchain-based access control scheme to ensure that the probability at which an adversary successfully accesses the data is a negligible probability. Our scheme consists of (1) a well implemented data collection module that is deployed in smart classrooms to collect and upload data to the data center; (2) a MongoDB-based data center and its control module that checks permissions on the blockchain and implements the results of the permissions; and (3) a role-based access control module maintained by a customized blockchain system that manages the access permissions and records access events in a trusted way.

The contributions of this paper are summarized as follows:

(1) We propose an educational data access control scheme to support trustworthy educational data management. We use blockchain as a trusted, distributed database to store and keep the updates of the security policies involved in the role-based access control scheme, thus achieving secure and trusted data management. We illustrate that our scheme is effective to preserve privacy for IoT-assisted educational big data management. By using blockchain to record the visit events of educational data, we make the access of educational data auditable

(2) We fully implement an educational data collection and access control system. The system includes a data collection module deployed in a smart classroom, a MongoDB-based data center and its control module, and a role-based access control module running on top of a customized blockchain system. Our system collectively analyzes the context of the smart classroom and can detect multiple scenarios such as absence, lateness, and gunshot

(3) We test the correctness and performance of our system. The results indicate that our system gives correct responses to users in less than one second, which is an acceptable performance for most application scenarios

The paper is organized as follows. Background and related works are presented in Section 2. Our blockchain-enabled access control scheme for educational data is proposed in Section 3. Experimental studies are reported in Section 4, and the paper is concluded in Section 5 with a discussion.

2. Previous Knowledge and Related Work

2.1. Previous Knowledge. Here, we introduce the key technologies and their related concepts used in our work.

2.1.1. Role-Based Access Control. We use the role-based access control (RBAC) model to represent and manage access privileges of the educational data. Role-based access control is a policy-neutral access-control mechanism defined around roles and privileges. Within an organization, users are grouped into different roles. The permissions to access certain series of data or to perform certain operations are assigned to specific roles rather than specific users. RBAC play a role as the bridge between users and permissions. A role represents a set of users and takes place of the users to be assigned permissions to, for simplification, clearance, and performance. In fact, there exist many other access control schemes such as attribute-based access control (ABAC), access control matrix (ACM), access control list (ACL), and capability-based access control (CapBAC). RBAC is proved to be equivalent to ACM with respect to the policies they can represent. Besides, RBAC is one of the most widespread, clear, and easy-to-develop access control models. The components of RBAC such as role-permission, user-role, and role-role relationships make it simple to perform user assignments, especially for user assignments on blockchain, because role-permissions, user-role, and role-role relationships are highly isomorphic with transactions on blockchain. And by maintaining RBAC with a blockchain system, we can guarantee that all access privileges are correctly stored and cannot be tampered with.

2.1.2. Blockchain. In our scheme, role-based access control is maintained by a blockchain system. Blockchain has served as a trustworthy environment for many different applications, ranging from secure transactions to trusted verifiable computing. Generally, blockchain can be regarded as a distributed ledger, which is kept by a series of computers called *blockchain nodes*. To make sure that every blockchain node keeps the same ledger, blockchain systems use *consensus*

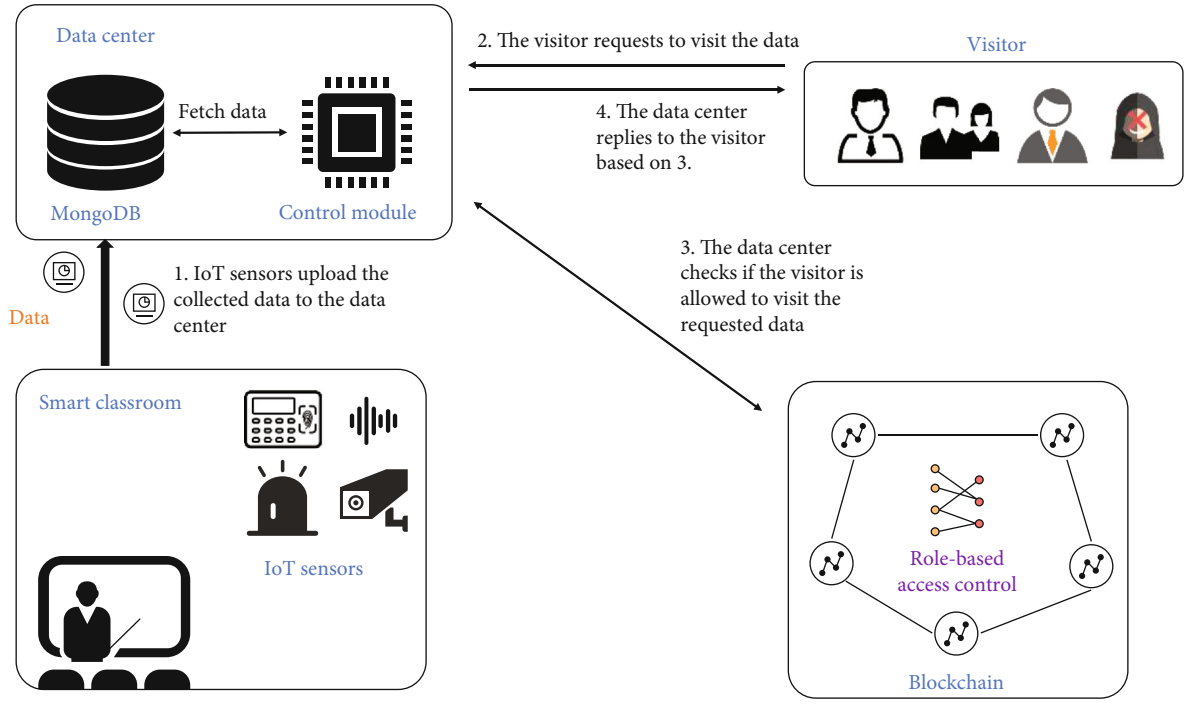

Figure 1: Abstract architecture of the system

algorithms. There are many kinds of consensus algorithms such as Proof-of-Work (PoW), Proof-of-Stake (PoS), and Delegated Proof-of-Stake (DPoS). Practical Byzantine Fault Tolerance (PBFT) represents the consensus algorithms from the Byzantine Fault Tolerance (BFT) consensus family. Although BFT consensus algorithms are well studied, their performance and scalability are still restricting their applications. In this paper, we choose PoW to be the consensus algorithm of our blockchain system. We make this choice for two reasons. On the one hand, PoW is the consensus algorithm for the first blockchain: bitcoin blockchain [6]. On the other hand, PoW is the most widely adopted consensus algorithm in blockchain community.

In blockchain, events recorded on the ledger are called *transaction*s, and transactions are packed into blocks to be added to the end of the blockchain. In PoW, nodes compete to get the right of packing blocks. To get this right, nodes need to find a nonce, by appending the nonce to the block and calculating its hash; the outcome hash is smaller than a predefined threshold. This process is called *mining*. As the outcome of the hash process can be seen as completely random, the only way to find such a nonce is to guess and try. Then, we can expect that nodes need to try many times to find a valid nonce and add the block to the blockchain. In our experiment, the difficulty of finding a valid nonce was decreased to make the blockchain system run faster.

2.2. Related Work. Before cloud computing becomes prevalent, most information and data are stored locally in users' computers. As cloud computing and mobile network prevail, educational programs, applications, and data are stored in clouds, and users do not know the specific storage location of personal data. In [7], Madeth raised awareness of privacy issues in E-learning that implicate user tracking and personal data usage for instructional purposes. In response to these privacy problems, a widely adopted method is to evaluate privacy-preserving technology of educational technology products. In fact, many schools and districts in the United States conducted privacy technology reviews on commonly used educational technology products with the help of technology review organizations [8]. The results indicate that most educational technology products cannot protect privacy. To solve the privacy problem fundamentally, a secure and trusted data management system is needed.

In fact, people have been trying to protect education privacy. Specifically, the main practices of American society to protect student privacy include three aspects: (1) publishing education privacy laws and regulations [9], (2) setting dedicated student privacy protection position at education departments [10], and (3) carrying out technical privacy reviews for educational products. At the technical level, preserving IoT data privacy in crowdsourcing with blockchain was studied by [11, 12]. Blockchain-based privacy preserving schemes on data uploading, trading, and sharing were explored by [13–15]. Blockchain systems addressing wireless challenges such as channel variation and adversarial jamming under IoT settings were thoroughly studied in [16, 17]. A cloud-enabled blockchain to support IoT applications taking advantage of the advances such as remote direct memory access and shared memory technology was presented in [18]. Trust extension from on-chain to off-chain and ground-truth data collection to blockchain were, respectively, investigated by [19, 20]. To the best of our knowledge, there is a lack of decentralized, trusted, automated access control solution to protect educational privacy, which is what this paper intends to address.

```
{
    "record_hash": "7e9c27b43a09c6e1ab...",
    "time": 1619856317,
    "value": 23,
    "location": "location 2",
    "GPS": {
        "x": 36.363,
        "y": 120.690
    }
}
```


(a) A sound record


```
{
    "record_hash": "b8ed724fae9aa3a099...",
    "action_type": "check-in",
    "RFID": "D1 D7 51 02",
    "username": "teacher2",
    "role": "teacher",
    "time": 1619856326,
    "location": "location 1",
    "GPS": {
        "x": 36.363,
        "y": 120.689
    }
}
```


(b) A check-in record


```
{
    "record_hash": "5b1da100ba8fb7281b...",
    "time": 1619856317,
    "value": "c217cab6a1df62a762b58b1e...",
    "location": "location 2",
    "GPS": {
        "x": 36.363,
        "y": 120.690
    }
}
```


(c) A photo record

Figure 2: Data stored in data center.

3. Blockchain-Based Access Control of Educational Data

In this section, we describe the details of our blockchain-based access control scheme of educational data. As illustrated in Figure 1, our scheme consists of a smart classroom with IoT devices, a data center, and a role-based access control module running on a blockchain system.

3.1. Smart Classroom. IoT devices continuously monitor and collect data in smart classrooms. The collected data is uploaded to the data center for further processing and analysis. In this paper, our IoT devices include sound sensors, RFID sensors, and cameras. Sound sensors can be used to monitor whether most students are studying attentively or just chatting with each other. They can also be used as gunshot detectors, to set up alarm and notify the police when gunshot is detected. RFID sensors can be used to record attendance of teachers and students, by giving each teacher and student an RFID card. Cameras can take photos and videos of the interior of the classroom. They can be very useful, because based on Artificial Intelligence and Computer Vision technologies, photos and videos can be used to recognize human faces, analyze students' focus and emotion, and extract many other useful information.

For privacy concern, we use a sound detection module as the sound sensor. It only senses the sound intensity of the environment without collecting detailed sound information such as the timbre, frequency, phase or, any other information about the waveform. So, the content of conversations in the classroom is not recognized. The sound detection module outputs a value between 0 and 1023 representing the current sound intensity in the environment. A larger output value means a louder environment. Particularly, as our experiment shows in Section 4, the output value ranges between 21 and 24 in a relatively quiet environment, and goes up to between 30 and 50 when a loud sound is detected.

Most schools, companies, and organizations use RFID sensors to take check-in and check-out records for their members. We do not explain how RFID sensors work here in detail, as it does not affect the design of our system. But we introduce how we use RFID sensors to collect important data. When a check-in action is detected (someone has tapped his/her RFID card or RFID tag at the RFID sensor), the user's RFID (usually a 4-byte array), the type of action (check-in or check-out), user's name, role, and the time and location of the action are collected. Besides, we calculate the SHA256 hash [21] of a record as its digest. Formally,

$$\text{Hash} = \text{SHA256}(\text{action} + \text{RFID} + \text{name} + \text{role} + \text{time} + \text{location} + \text{GPS}). \quad (1)$$

Using hash, we can setup a trusted digest to the activity, verify the integrity of data, and increase difficulty for attackers to tamper with the records.

All the collected data including sound, RFID records, and photos are uploaded and stored in the data center, for further processing and analysis.

3.2. Data Center. Collected educational data are stored in the data center. Teachers, parents of students, education managers, or someone else may need to access these data for different reasons, such as making educational decisions,

```
1: Initialization: Synchronize the blockchain object bc with blockchain nodes to get the newest state of the access model. Connect to
the MongoDB database and get an object db.
2: // verify the permission of an access request
3: function VERIFY(uid, hash, sig)
4:     // check if the user is offering correct signature to be identified as user uid
5:     if bc.verifySignature(uid, sig) == false then
6:         return false
7:     end if
8:     role = bc.getRole(uid)
9:     tags = bc.getTags(hash)
10:    for tag in tags do
11:        if bc.findState(role, tag) == false then
12:            return false
13:        end if
14:    end for
15:    for tag in tags do
16:        if bc.findState(role, tag) == true then
17:            return true
18:        end if
19:    end for
20:    return false
21: end function
22: function RUNSERVER
23:    while true do
24:        uid, hash, sig, addrFrom = getRequestParams()
25:        if verify(uid, hash, sig) == false then
26:            sendMessage("Authorization failed.")
27:        else
28:            sendMessage("Authorization suceed.")
29:            data = db.getData(hash)
30:            sendData(addrFrom, data)
31:        end if
32:    end while
33: end function
```

Algorithm 1: Data center control utilities.

Table 1: Access control rules in our experiment.

	Sound	Check-in & check-out	Camera
Students	✓	✗	✗
Parents	✓	✓	✗
Teachers	✓	✓	✓
Education managers	✓	✓	✓
Unauthorized people	✗	✗	✗

guarding safety of the school, and teaching enrichment. In our scheme, these educational data are stored in a MongoDB database.

MongoDB is a popular NoSQL, nonrelational database for modern app development [22]. When compared to relational databases, NoSQL databases are often more scalable and can provide superior performance. SQL databases are most often implemented in a scale-up architecture, which is based on larger computers with more CPUs and more memory to improve performance, while NoSQL databases are created in Internet and cloud computing eras that make it possible to more easily implement a scale-out architecture. In addition, the flexibility and ease of use of their data models can speed up development in comparison to the relational model, especially in IoT and cloud computing environments.

In our design, the database stores three kinds of data: sound records, check-in and check-out records, and photo records. For a sound record, we store 5 fields: record hash, time, value, location, GPS: a check-in and check-out record contains 8 fields: record hash, action type, RFID, user's name, user's role, time, location, and GPS. And a photo record contains 5 fields: record hash, time, value, location, and GPS. Figure 2 shows one example of each kind of data.

We attach each record of data its hash as its index in both the data center and the blockchain. To protect privacy of students and teachers, access of these data should be under control. Data center should only allow authenticated access of designated data. In our access control scheme, we adopt the role-based access control model and deploy it on a customized blockchain system. When a visitor requests to access some data, the data center checks on the blockchain whether the visitor's role is allowed to access the requested data. If so, the data center grants to the visitor the access right to the data. Otherwise, the data center refuses the visitor's request. Based on this principle, we propose a control module of the data center to process visitors' access requests and verify

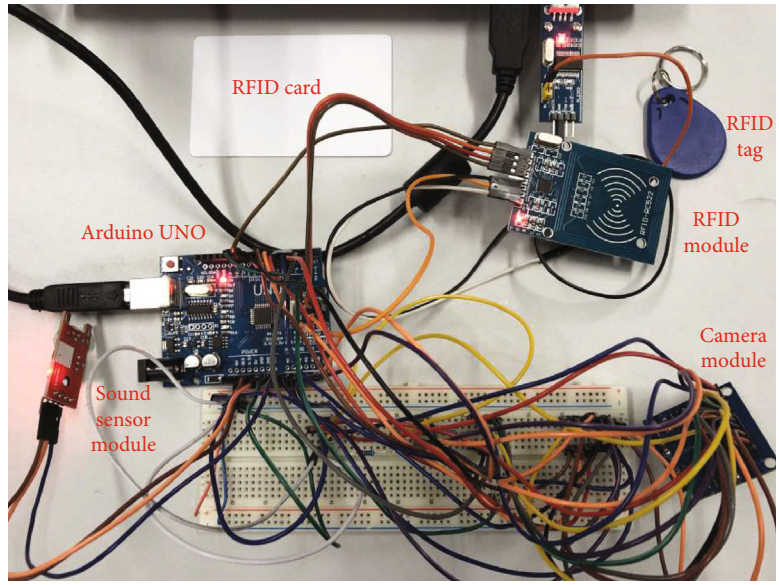

Figure 3: Data collection module deployment.

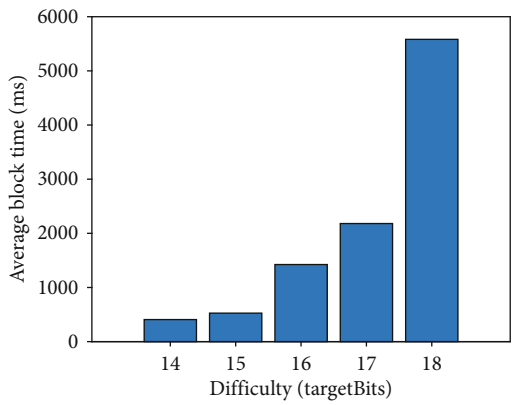

Figure 4: Average block time under different difficulty settings.

the permissions of visitors' access on the blockchain. Specifically, the control module is programmed to synchronize the state of RBAC module as a blockchain node and makes access control decisions based on the state of the RBAC module. Algorithm 1 shows the main frame of the control module's workflow.

In our implementation, the main thread of the control module runs the *runServer* function, which continuously waits for access requests. When receiving a request, *runServer* parses parameters of the request and verifies whether it is permitted or not, using the core part of the control module, *verify* function.

The *verify* function first checks the signature to make sure that the access request is sent by the corresponding user *uid*. Then, it extracts the tags of the data and examines the role of *uid*. Following that are two *for* loops, with the first one checking if the role of *uid* has been banned from some tag of the data and returns false if it is true and the second one checking if the role of *uid* has been authorized to access the data and returns true if it is true.

3.3. RBAC Blockchain System. As mentioned earlier, role-based access control (RBAC) is a policy-neutral access-control mechanism defined around roles and privileges. The components of RBAC such as role-permission, user-role, and role-role relationships make it simple to perform user assignments. A study by NIST has demonstrated that RBAC addresses many needs of commercial and government organizations. RBAC can be used to facilitate administration of security in large organizations with hundreds of users and thousands of permissions. Although RBAC is different from mandatory access control (MAC) and discretionary access control (DAC) frameworks, it can enforce these policies without any complication. Under the role-based access control model, users are grouped into several roles. Access actions of users are permitted or refused based on their roles.

For example, in our experiment described in Section 4, the roles and access control rules are designated as Table 1 shows. There are 5 roles in total: students, students' parents, teachers, education managers, and unauthorized people.

In our access control scheme, we use a blockchain system to implement the RBAC model. We authenticate user identities using SHA256 signatures and public cryptography schemes. The identities are registered on the blockchain via an authenticated trusted blockchain node. After registration, a public-private key pair is generated for each user, and the trusted node broadcasts a *user-registration* transaction on the blockchain. The transaction includes designated role for the user, and the public-private key pair can be used to verify whether the role in the transaction is designated to the user by verifying the SHA256 signature. To achieve role-based access control features, we implement 4 transaction types:

(i) *User-registration*: as described above, we use *user-registration* transactions to register an identity for a user. The format of a user-registration transaction is $\{user - reg, pk_{uid}, role\}$, in which *uid* is the user's

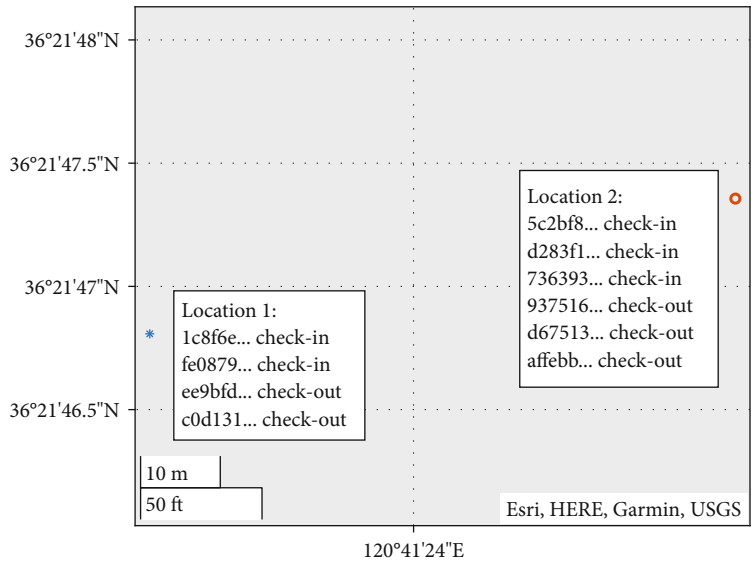

Figure 5: Check-in record.

Table 2: RFID record.

Record hash	Action	RFID	Name	Role	Time	Location	GPS
1c8f6e...	Check-in	E1 0D 2D 21	Teacher1	Teacher	08:05:17	Location 1	36.363, 120.689
fe0879...	Check-in	D1 D7 51 02	Teacher2	Teacher	08:05:26	Location 1	36.363, 120.689
5c2bf8...	Check-in	06 7F FB AD	Student1	Student	08:25:38	Location 2	36.363, 120.690
d283f1...	Check-in	82 3F B1 58	Student3	Student	08:25:43	Location 2	36.363, 120.690
736393...	Check-in	6D B6 6B 4A	Student2	Student	08:25:49	Location 2	36.363, 120.690
937516...	Check-out	6D B6 6B 4A	Student2	Student	17:01:23	Location 2	36.363, 120.690
d67513...	Check-out	06 7F FB AD	Student1	Student	17:01:48	Location 2	36.363, 120.690
affebb...	Check-out	82 3F B1 58	Student3	Student	17:01:59	Location 2	36.363, 120.690
ee9bfd...	Check-out	E1 0D 2D 21	Teacher1	Teacher	18:00:16	Location 1	36.363, 120.689
c0d131...	Check-out	D1 D7 51 02	Teacher2	Teacher	18:00:29	Location 1	36.363, 120.689

id, pk_{uid} is the public key generated for the user, and *role* is the designated role for the user

(ii) *Role-registration*: like *user-registration* transactions, *role-registration* transactions are used to register a new role for the system. For a *role-registration* transaction $\{role - reg, role\}$, *role* is the name of the role being registered

(iii) *Rule-edit*: we use *rule-edit* transactions to create or edit role-based access control rules. For example, transaction $\{rule - edit, role, tag, true/false\}$ creates or edits a rule to allow/forbid users of role *role* to access data with tag *tag*

(iv) *Access-result*: the blockchain system employs *access-result* transactions to respond to the data center's query about whether a user can access some data. Transaction $\{result, uid, tag, t, true/false\}$ means the user of id *uid* can or cannot access the data of tag *tag*, where *t* is the timestamp of the request action. *Access-result* transactions play the role of an immutable access log and make the request action auditable and responsible

The main benefit of running an RBAC model on a blockchain lies in that as all transactions are confirmed by all blockchain nodes, and no adversary can change any user's role at its own will.

4. Experiment

In this section, we report the evaluation results of our system in a practical scenario. We implemented the blockchain-based educational data access control system and used the system to perform the whole process of the educational data from collection, storage, to controlled access.

4.1. Setup. As shown in Figure 3, we used a data collection module to simulate the IoT devices in a smart classroom. The data collection module was implemented on an Arduino

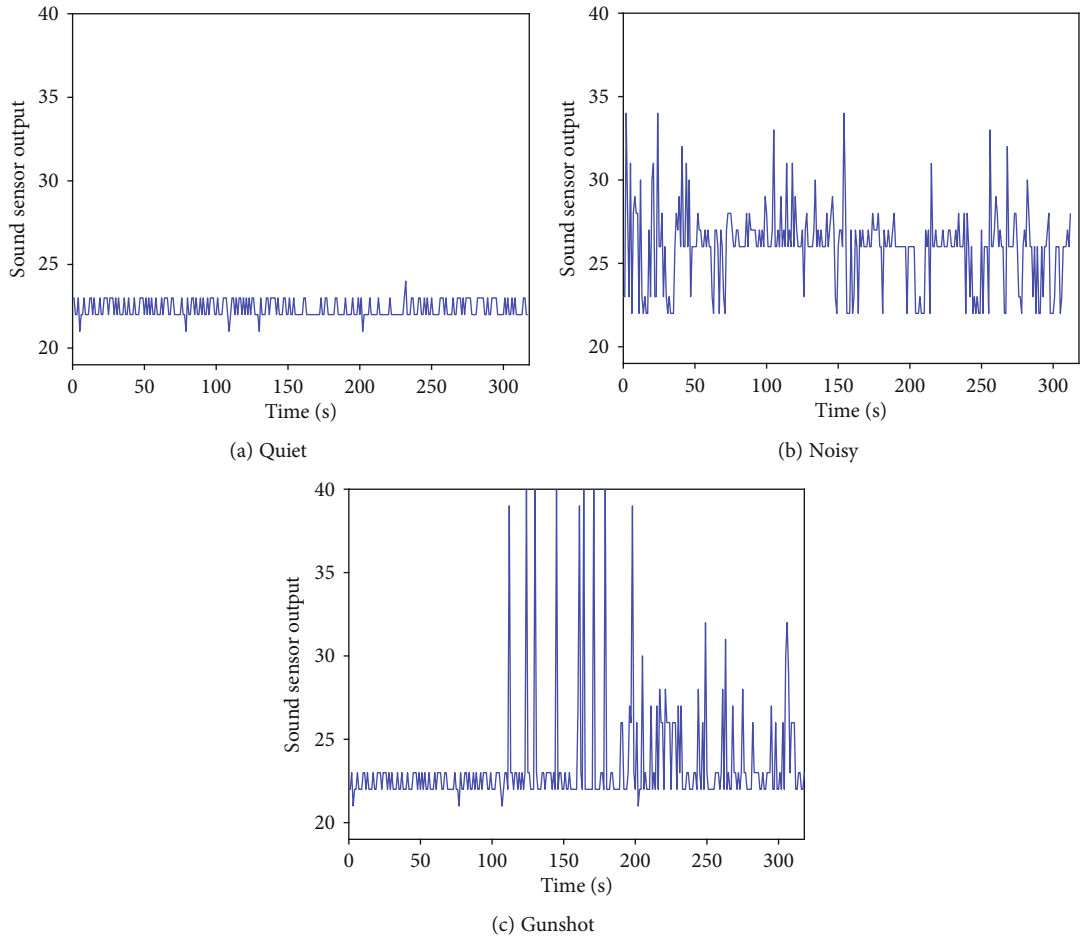

Figure 6: Sound monitoring.

UNO, which was connected to several sensor modules, including the following:

(1) OV7670, a camera module

(2) SY-M213, a sound sensor module

(3) RFID-RC522, an RFID module

At the Arduino UNO, we develop and assemble the drivers of the camera module, sound sensor module, and RFID module in C language. The Arduino UNO board was a microcontroller board based on the ATmega328P, which supports USB connection with a computer [23]. At every second, the camera module uploaded a 320 ∗ 240-sized grayscale image and the sound sensor module uploaded its output (it measures the sound intensity of the environment). The RFID module uploaded a record each time an action was detected.

All the collected data were uploaded to a Lenovo G580 PC running Windows 10 Professional 20H2 through serial port. We developed a Python program to display the data read from serial port. It ran on the Lenovo G580 PC and uploaded data to the data center while displaying the collected data.

For the data center, we developed a control module using Python to process visitors' access requests and verify the authentication of visitors' access permissions on the blockchain. This module operated as both a server and a blockchain node. It read role-based access control information in on-chain transactions. If a visitor's access permission was authenticated, the control module would fetch data from the MongoDB database and send the data to the visitor by writing the data into the response body of the HTTPS connection. The control module and MongoDB ran on a 16-inch 2019 MacBook Pro with 8-Core Intel i9 @ 2.4 GHz and 16 GB memory that operated on macOS 11.3.

We developed our own blockchain system using Golang for the best flexibility of customization. Golang is a popular programming language in blockchain community and has become a go-to language for developing decentralized systems [24]. We used PoW consensus algorithm, which has practically the best performance and scalability. The PoW difficulty was reduced to 16 leading zero bits as we did not have as much computing power as the bitcoin network has to produce blocks in an acceptable time. That is, mining nodes needed to find a nonce that by appending the nonce to the block data, the produced SHA256 hash had 16 zero bits in the front. So, the expected try times of different nonces for mining a block were $2^{16} = 65536$. We ran the blockchain system on three computers, with each having 8-Core Intel i7-9700 CPU @ 3 GHz and 16 GB memory and running

(a) Students are studying

(b) Class is over

(c) Empty classroom

Figure 7: Camera recording.

Windows 10 Professional 20H2. One blockchain node ran on each of the three computers. Figure 4 shows the average time the blockchain network takes to mine a block, under different difficulty settings. In our implementation (targetBits = 16), the difficulty-reduced PoW blockchain network takes 526 ms to produce a block in average.

4.2. Evaluation

4.2.1. Data Collection. Using the RFID module, we recorded check-in and check-out actions at two different locations (shown in Figure 5). Each time a teacher or a student checks in (swipes his/her RFID card at the RFID sensor), information of his/her identity and the check-in and check-out action including time, location, and hash are collected and uploaded by the RFID module. In our experiment, location 1 is the office of the teacher, and location 2 is the classroom where the students study. Details of these actions are shown in Table 2.

In this experiment, we monitored the environment sound intensity in a classroom with the sound sensor module. As shown in Figure 6, three patterns of environment sound intensity were recorded. In the first pattern (Figure 6(a)), the classroom was relatively quiet, and the uploaded value from the sound sensor module was ranged from 21 to 24. In the second pattern (Figure 6(b)), the classroom was noisy, so the sound sensor module uploaded value higher than 25 with a high frequency. In the third pattern (Figure 6(c)), we simulated a gunshot scene with a loud speaker. From the 110 seconds to the 200 seconds, we used the speaker to a play gunshot sound at the entrance of the classroom. After the sound was played, from the 200 seconds, students in the classroom began to scream; then, the classroom became as noisy as it was in the second pattern.

We also used the camera module to take photos of the interior of the classroom. Limited by the performance of the OV7670 camera module, only one photo per second was taken. Figure 7 shows three representative scenes in the classroom: students studying in the classroom (Figure 7(a)), students leaving the classroom when the class was over while several students chose to stay for discussions (Figure 7(b)), and an empty classroom (Figure 7(c)).

These educational data were all uploaded and stored in the MongoDB database. Further analysis and data process can be done after access control.

4.2.2. Access Control of Collected Data. For simplicity and convenience, we ran the three nodes of the blockchain system, the data center, and the data collection module in the same local area network. This resulted in low network latency. By sending *role-registration* transactions and *user-registration* transactions, we registered 8 users of 5 roles: 3 students, 2 parents, 1 teacher, 1 education manager, and 1 unauthorized person. By sending *rule-edit* transactions, we created the following role-based access control rules:

(i) Students can access the sound monitor data, to get noticed when gunshot is detected

(ii) Parents can access the check-in records of their children and the sound monitor data, to see if their

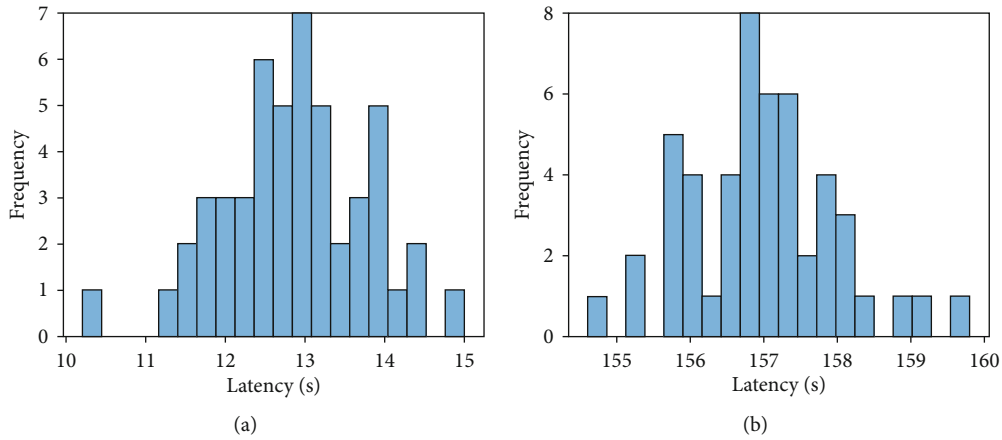

Figure 8: Response latency.

children have gone to school after leaving home and to be aware of students' study environment

(iii) The teacher and the education manager can access all the educational data, for teaching evaluation and enrichment, educational decision making, school safety guarding, etc.

(iv) The unauthorized person cannot access any data, as he or she is not authorized

Then, we tested our educational data access control system. We used different *uid*-sk_{uid} pairs to simulate different users and sent requests to the data center to access the data collected from the smart classroom. We sent 100 requests, 50 of them were good ones that should be accepted, while the other 50 were bad that should be refused. As a result, our access control system worked correctly. That is, for all the 50 good requests, the data center sent data to the user, and for all the 50 bad ones, the data center refused to offer data to the requester. Besides, the result of each request was logged on the blockchain in the form of an *access-result* transaction. We also analyzed the performance of our access control system. In our observation, it costs the user 13 ms in average to get a refuse message (Figure 8(a)) or 157 ms in average to get the requested data (Figure 8(b)), counting from sending a request to the data center. It can be concluded that the response time of our educational data access control system under local area network is acceptable.

5. Conclusions

In this paper, we proposed a scheme to preserve privacy in educational application of IoT. We achieved our privacy preservation goal by implementing a blockchain-based access control system. We implemented the full system including the components of collecting educational data, storing the data in a data center, and maintaining a role-based access control on educational data. Our scheme consists of a data collection module, a MongoDB-based data center, and the role-based access control module running on top of a blockchain system. Our educational data access control system guarantees correct execution of the access control rules and makes the access events of educational data auditable and responsible. Our experiment results indicate that our access control system gives a correct response as quickly as a traditional data server. What is more, our access control system was designed to be relatively general-purpose. So, it can be easily extended to other application fields, by including necessary IoT devices and implementing drivers for them.

Data Availability

The recorded data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

[1] J. Marquez, J. Villanueva, Z. Solarte, and A. Garcia, "Iot in education: integration of objects with virtual academic communities," in *New Advances in Information Systems and Technologies*, Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, and M. M. Teixeira, Eds., vol. 444, pp. 201–212, Springer International Publishing, 2016.

[2] F. Moreira, M. J. Ferreira, and A. Cardoso, "Higher education disruption through IoT and big data: a conceptual approach," in *Learning and Collaboration Technologies. Novel Learning Ecosystems*, P. Zaphiris and A. Ioannou, Eds., pp. 389–405, Springer International Publishing, 2017.

[3] M. Bagheri and S. H. Movahed, "The effect of the Internet of Things (IoT) on education business model," in *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 435–441, Naples, Italy, 2016.

[4] Parent Coalition for Student Privacy, *inBloom Timeline*, vol. 1, 2021, https://studentprivacymatters.org/inbloom-timeline/.

[5] The Beijing News, *Senior high school girls were interviewed for live broadcast, claiming to be early adopters*, vol. 1, 2021,

http://epaper.bjnews.com.cn/html/2016-09/04/content_650845.htm?div=1.

[6] N. Satoshi, "Bitcoin: A Peer-to-Peer Electronic Cash System," *Decentralized Business Review*, vol. 1, p. 21260, 2008.

[7] M. May and S. George, "Privacy concerns in e-learning: is using tracking system a thread?," *International Journal of Information and Education Technology*, vol. 1, no. 1, 2011.

[8] Common Sense, "2019 State of Edtech Privacy Report," vol. 1, 2019.

[9] U.S. Department of Education, *Department of Education's Computer Matching Agreements*, vol. 1, 2021, https://www2.ed.gov/about/offices/list/om/pirms/cma.html.

[10] Chief Privacy OfficerU.S. Department of Education.

[11] S. Zhu, Z. Cai, H. Hu, Y. Li, and W. Li, "zkCrowd: a hybrid blockchain-based crowdsourcing platform," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4196–4205, 2020.

[12] S. Zhu, W. Li, H. Li, L. Tian, G. Luo, and Z. Cai, "Coin hopping attack in blockchain-based IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4614–4626, 2019.

[13] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[14] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.

[15] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.

[16] M. Xu, C. Liu, Y. Zou, F. Zhao, J. Yu, and X. Cheng, "wChain: a fast fault-tolerant blockchain protocol for multihop wireless networks," *IEEE Transactions on Wireless Communications*, p. 1, 2021.

[17] M. Xu, F. Zhao, Y. Zou, C. Liu, X. Cheng, and F. Dressler, "BLOWN: a blockchain protocol for single-hop wireless networks under adversarial SINR," *arXiv:2103.08361*, vol. 1, 2021.

[18] M. Xu, S. Liu, D. Yu, X. Cheng, S. Guo, and J. Yu, "CloudChain: a cloud blockchain using shared memory consensus and RDMA," *arXiv:2106.04122*, vol. 1, 2021.

[19] C. Liu, H. Guo, M. Xu et al., "Extending on-chain trust to off-chain-a trustworthy vaccine shipping example," *arXiv:2106.15934*, vol. 1, 2021.

[20] C. Liu, M. Xu, H. Guo et al., "Tokoin: a coin-based accountable access control scheme for Internet of Things," *arXiv:2011.04919*, vol. 1, 2020.

[21] National Institute of Standards and Technology (NIST), *SHA-2 Standard*, vol. 1, 2021, https://www.itl.nist.gov/fipspubs/fip180-2.htm.

[22] Mongo DB, *What is NoSQL? NoSQL Databases Explained*, vol. 1, 2021, https://www.mongodb.com/nosql-explained.

[23] Arduino, "Arduino uno description," 2021, https://store.arduino.cc/arduino-uno-rev3.

[24] S. J. Naqvi, "Why I am building a blockchain in Go," *Karachain*, vol. 1, 2018.