Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2741435, 12 pages
https://doi.org/10.1155/2021/2741435

Research Article

BEI-TAB: Enabling Secure and Distributed Airport Baggage Tracking with Hybrid Blockchain-Edge System

Pengbo Si, Fei Wang, Enchang Sun, and Yuzhao Su

Faculty of Information Technology, Beijing University of Technology, 100124, China

Correspondence should be addressed to Pengbo Si; sipengbo@bjut.edu.cn

Received 7 July 2021; Accepted 27 August 2021; Published 23 September 2021

Academic Editor: Michel Kadoch

Global air transport carries about 4.3 billion pieces of baggage each year, and up to 56 percent of travellers prefer obtaining real-time baggage tracking information throughout their trip. However, the traditional baggage tracking scheme is generally based on optical scanning and centralized storage systems, which suffers from low efficiency and information leakage. In this paper, a blockchain and edge computing-based Internet of Things (IoT) system for tracking of airport baggage (BEI-TAB) is proposed. Through the combination of radio frequency identification technology (RFID) and blockchain, real-time baggage processing information is automatically stored in blockchain. In addition, we deploy Interplanetary File System (IPFS) at edge nodes with ciphertext policy attribute-based encryption (CP-ABE) to store basic baggage information. Only hash values returned by the IPFS network are kept in blockchain, enhancing the scalability of the system. Furthermore, a multichannel scheme is designed to realize the physical isolation of data and to rapidly process multiple types of data and business requirements in parallel. To the best of our knowledge, it is the first architecture that integrates RFID, IPFS, and CP-ABE with blockchain technologies to facilitate secure, decentralized, and real-time characteristics for storing and sharing data for baggage tracking. We have deployed a testbed with both software and hardware to evaluate the proposed system, considering the performances of transaction processing time and speed. In addition, based on the characteristics of consortium blockchain, we improved the practical Byzantine fault tolerance (PBFT) consensus protocol, which introduced the node credit score mechanism and cooperated with the simplified consistency protocol. Experimental results show that the credit score-based PBFT consensus (CSPBFT) can shorten transaction delay and improve the long-term running efficiency of the system.

1. Introduction

According to the latest data from the International Air Transport Association (IATA), global air transport carries about 4.3 billion pieces of baggage every year [1]. The annual increase of baggage brings new challenges to airports. The current status and problems in baggage tracking are as follows:

(1) Optical scanning code is widely used for baggage tracking which leads to the inability to collect real-time information of baggage handling causing passengers' anxiety for waiting and management's failure to grasp the handling situation of baggage [2]

(2) The lack of unitive platform that managed and shared processing information for each baggage leads to information islands [3]

(3) The current logistics records are recorded in centralized database, which more likely leads to passenger information leakage and tampering

Compared with optical scanning code, RFID possesses advantages of fast recognition speed, large data capacity, long service life, and reusability. Baggage tracking technology based on RFID is one of the most advanced technologies in international baggage management, but the data sharing pattern and security still need to be promoted [4].

Blockchain technology could be a promising technology that brings essential changes to air baggage tracking. Blockchain maintains and records transactions of events that are immutable and cannot be falsified. It provides transparent, secure, and trustworthy data in both private and public domains, which solves the problem of information leakage and tampering caused by single centralized database in baggage tracking systems. At the same time, in traditional ways, it is difficult to guarantee data privacy and build trust between participants in multidepartment cooperation. The decentralized nature of blockchain can efficiently establish a data-sharing model and ensure multidepartment encryption cooperation. In addition, the traceable chain structure of blockchain ensures that the data cannot be tampered with, which can significantly improve the trustworthiness of baggage tracking and retrieval. Compared with public blockchain, consortium blockchain only supports the access of the nodes participating in maintenance, and participating nodes need authorization before joining and maintaining blockchain. In addition, its authorized access features can reduce the degree of data leakage, thus enhancing the privacy security of data. Consortium blockchain can guarantee the security of data, but its scalability is challenged in the face of hundreds of millions of baggage [5], so we introduced IPFS in our system.

IPFS provides a point-to-point (P2P) distributed storage structure, which can easily store a large amount of passenger data. IPFS is a content-addressed block storage system with features such as secure transaction hash mapping, high throughput, and concurrent access to transactions by peers in the network [6]. Besides, we take advantage of CP-ABE to realize attribute-based access control.

Consensus algorithm affects the performance of blockchain system. The PBFT algorithm is mainly used in consortium blockchain to solve Byzantine general problem. However, the consistency protocol of PBFT requires to complete two times of node communication which complexity is $O(N^2)$, where N is the total number of nodes in the network, resulting in high communication complexity and cost.

Aiming at the above problems, we propose a blockchain and edge computing-based IoT system for tracking of airport baggage (BEI-TAB). Our contributions are mainly as follows:

(1) The application of blockchain in baggage tracking not only reduces the degree of data leakage but also enhances the privacy and security of data through utilizing the features of nontampering of blockchain and authorized access of consortium blockchain

(2) The multichannel design enables the airport to process multiple types of data and business requirements in parallel and rapidly, providing coarse-grained privacy protection and promotes information sharing. It realizes the physical isolation of data and further ensures the confidentiality of transmission

(3) By combining RFID with the blockchain, the real-time baggage processing information is automatically stored in the blockchain, which effectively saves the labor cost as well as guarantees the safety of data transmission and improves the degree of informatization

(4) The integration of IPFS and blockchain realizes the storage of encrypted basic baggage information in the IPFS network, while only the IPFS address hash is stored in blockchain, which increases the scalability of the blockchain system. At the same time, the application of CP-ABE takes advantage of attribute-based encryption and provides fine-grained privacy protection

(5) Both software and hardware were deployed in a testbed to evaluate the performance of transaction processing time and speed for the proposed system

(6) In order to simplify the communication process, in the absence of Byzantine nodes, CSPBFT adopted a simplified consistency protocol to reduce the communication traffic between nodes. Besides, the node credit score mechanism effectively identifies and excludes Byzantine nodes in the system, so that the algorithm can execute the simplified consistency protocol most of time, thus improving the long-term operation efficiency of the system

2. Related Works

In this section, we review the related work on baggage tracking. At present, the improvement of baggage handling system mainly combines new technologies such as machine vision and IoT to solve baggage transportation errors, and there are few methods using blockchain technology. Singh et al. [7] propose a design of baggage tracing and handling system using smart RFID tags and IoT which is based on cloud server. However, the baggage's real-time position is tracked and stored in a cloud, which centralized storage potentially leads to information leakage. Jerry et al. [8] propose a system based on RFID, ZigBee, and GSM to update the status of baggage at various points in the journey map. However, it does not focus on secure information transmission and sharing. Johnson et al. [9] design a machine vision-based airport baggage tracking system using an integral image to obtain the bag location, but the massive amount of information poses a challenge to the system. Gao and Liang [10] adopt a convolutional neural network with video input to detect the appearance transportability of baggage. However, problems such as baggage recovery and reliable cooperation between different departments cannot be resolved. Wang et al. [11] introduce a social network that combines the IPFS, Ethereum, and attribute-based encryption to realize the access control to the data by setting out the access policy, but it has two problems. Firstly, the security level of public blockchain is excellent, but it also has slow transaction process speed and low throughput [12]. Conversely, the nodes in the consortium blockchain do not need to keep accounts through the competitive consensus mechanism so that the consensus transaction speed is higher than public blockchain. In addition, its authorized access mechanism can reduce the degree of data leakage and thus enhances the security of privacy data. As noted above, our system chooses Hyperledger Fabric, which is one of the most popular consortium blockchain platforms, as our blockchain platform.

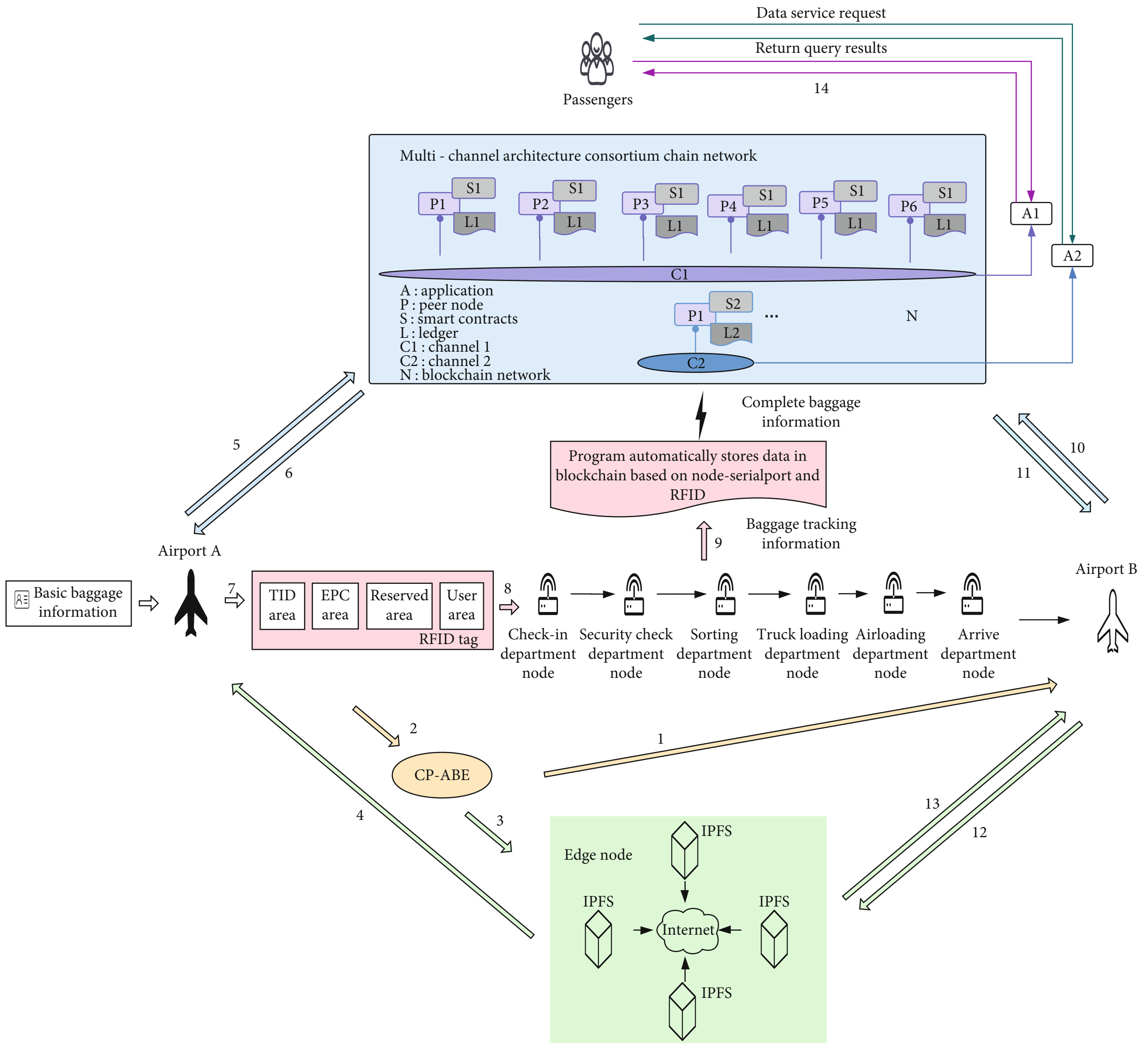


Figure 1: BEI-TAB architecture.

3. BEI-TAB Architecture

In this section, we illustrate the overall architecture of BEI-TAB and account for the following entities that take part in our architecture by referring to Figure 1.

Nodes. On the one hand, the nodes in consortium blockchain network refer to departments of airlines. There are six basic departments involved in baggage transport at one time: check-in, security check, sorting, trucking loading, air loading, and arrival. On the other hand, these nodes are also equipped with RFID readers. RFID readers obtain data from RFID tags by radio waves, which is a kind of automatic identification and data collection (AIDC) technology [13]. RFID system is composed of three components: RFID tags, RFID readers, and antenna. The internal storage area of RFID tags can be divided into four areas:

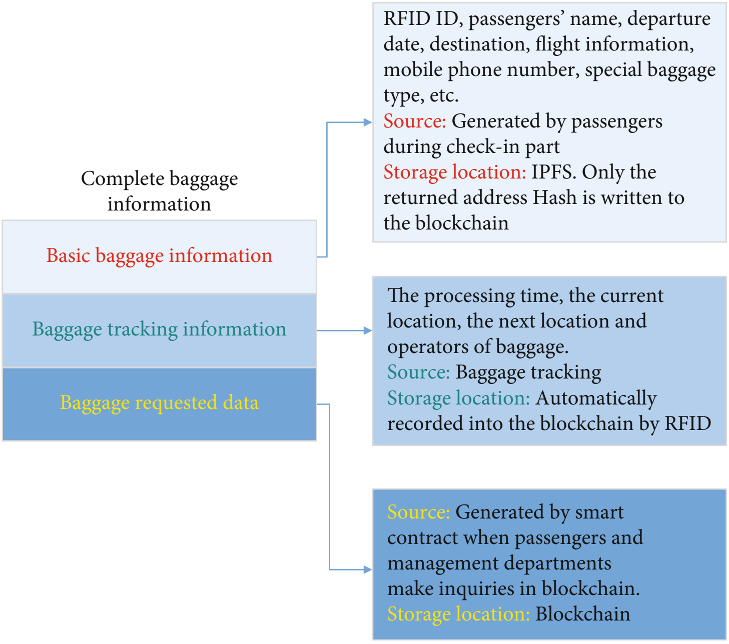
(1) *Tag Identification (TID) Area*. Storing TID number, it is readable but not writable, and each TID number is unique

(2) *User Area*. Storing user-defined data

(3) *Electronic Product Code (EPC) Area*. Storing EPC numbers, which are unique electronic codes of objects

(4) *Reserved Area*. Storing kill password and access password

Edge nodes. Edge nodes play the role of off-chain storage devices in our architecture. We build the IPFS network on edge nodes, which stores basic baggage information data and imposes attribute-based access control policies. IPFS is a decentralized data management system based on a P2P network model. It can connect computer devices in the same


Complete baggage information
Basic baggage information
Baggage tracking information
Baggage requested data

RFID ID, passengers' name, departure date, destination, flight information, mobile phone number, special baggage type, etc.
Source: Generated by passengers during check-in part
Storage location: IPFS. Only the returned address Hash is written to the blockchain

The processing time, the current location, the next location and operators of baggage.
Source: Baggage tracking
Storage location: Automatically recorded into the blockchain by RFID

Source: Generated by smart contract when passengers and management departments make inquiries in blockchain.
Storage location: Blockchain

Figure 2: A complete piece of baggage information.

network while ensuring that files will not be tampered with [14]. IPFS stores files through content-addressed hash in a distributed hash table (DHT) which adopts version-control history to remove duplicate files. After uploading a file to IPFS, it returns the unique content-addressed hash while users only need this hash to access the resource [15].

Airport. Different airports play the role of baggage information provider or visitor. Encryption policies are implemented in CP-ABE to control business access of different airport departments to ensure decentralized and secure characteristics for basic baggage information.

Attribute-based encryption (ABE) is an access control technology. Private keys and ciphertexts in ABE are associated with the attributes of users or organizations. The resource providers only need to encrypt the message according to the attributes, no longer need to pay attention to the number or identity of the members in the group, which reduces the data encryption cost and protects the privacy of users. ABE can be divided into two categories: CP-ABE and key policy attribute-based encryption (KP-ABE). In CP-ABE, the access policy is generated by senders and bounded to the ciphertext, and private key is combined with the user's own attributes. In KP-ABE, the access policy is generated by receivers. As described above, CP-ABE is more favorable for our framework compared with KP-ABE. CP-ABE has four algorithms [16]:

$$\text{Setup}(\lambda) \longrightarrow \text{PK}, \text{MSK}. \tag{1}$$

The setup algorithm takes the security parameter λ as input and outputs the public key PK and master key MSK.

$$\text{Encrypt}(\text{PK}, p, m) \longrightarrow \text{CT}. \tag{2}$$

The encryption algorithm takes PK, message m, and an access policy p as input. It will produce ciphertext CT.

$$\text{keyGen}(\text{MSK}, \text{S}) \longrightarrow \text{SK}. \tag{3}$$

The key generation algorithm accepts the input including MSK and a set of attributes S. It creates a private key SK which is linked with attributes.

$$\text{Decrypt}(\text{PK}, \text{CT}, \text{SK}) \longrightarrow m. \tag{4}$$

The decryption algorithm receives the input including CT , p, and SK. Attributes that satisfy the policy p are able to decrypt the message. This step will decrypt the ciphertext and return message$m$.

Complete baggage information. A complete piece of baggage information is divided into three parts as Figure 2 showed.

Smart contract. The business logic of smart contract can be summarized as following three parts:

(1) Combining with RFID to realize real-time automatic storage of baggage tracking information in blockchain

(2) Providing data access and interaction interfaces for passengers and administrators, respectively. Passengers can query baggage information through RFID ID to obtain the whole process of baggage tracking, thus reducing anxiety. The administrator can carry out accurate or batch retrieval through RFID ID and flight number so that the handling status of baggage can be grasped

(3) Generating statistics of baggage request data when passengers and management departments make real-time query requests

The smart contract interface is shown in Table 1.

Table 1: Smart contract interface.

Interface definition	Interface description	Function description
Query	Information query	Data sharing and retrieval
Delete	Information delete	
CreateZJxl	Check-in department information added to the blockchain	Data access and storage
CreateAJxl	Security check department information added to the blockchain	
CreateFJxl	Sorting department information added to the blockchain	
CreateZhuangJxl	Truck loading department information added to the blockchain	
CreateZCxl	Air loading department information added to the blockchain	
CreateDDxl	Arrival department information added to the blockchain	
QueryID	Query the whole process information and the current search times through RFID ID	Data sharing and retrieval
GethbID	Query the whole process information and the current search times through flight number	

Our system developed a consortium blockchain network on the Hyperledger Fabric with nodes which are check-in department node, security check department node, sorting department node, trucking loading department node, air loading department node, and arrival department node. At the same time, we deployed IPFS cluster on the edge nodes. Furthermore, the multichannel architecture was designed to achieve physical isolation for different businesses and coarse-grained access control. The multichannel structure refers to a channel corresponding to a business of airline company or different businesses corresponding to different channels. A channel is parallel to a consortium blockchain. Channels are physically isolated from each other so that ledger information is only visible to the members of the channel thus providing coarse-grained privacy protection. The multichannel design also enables the airport to process multiple types of data and business requirements in parallel and rapidly. Each channel was equipped with smart contract to realize data management and sharing control. Moreover, RFID readers were also deployed on the nodes. Baggage processing information of each department is automatically recorded into the blockchain in real-time through RFID. Conversely, basic baggage information was encrypted by CP-ABE and then written to IPFS deployed at edge nodes. Only the returned address hash was written to the blockchain to enhance the scalability of the blockchain and protect the privacy of passengers. Additionally, smart contract provides interfaces for passengers and airlines, respectively. Passengers and airlines can query baggage tracking information according to RFID ID. In addition, airlines can conduct batch queries through flight number, but only those whose attributes conform to the access control policy can request basic baggage information. At the same time, smart contract calculates baggage requested data for each piece of baggage.

4. CSPBFT Consensus Mechanism

PBFT consensus algorithm is designed to solve the consistency problem of distributed systems with Byzantine nodes [17]. It mainly consists of consistency protocol, view change protocol, and checkpoint protocol, among which consistency protocol is the core. The consistency protocol of PBFT requires to complete two times node communication with complexity $O(N^2)$, which ensures that the algorithm can achieve consensus even if Byzantine nodes exist in the network. The process of PBFT is mainly shown in Figure 3. Client c is the sender of the request. The primary node receives the requests, sorts them, assigns numbers, and broadcasts them to replicas in the network. The replica is mainly responsible for receiving the messages sent by the primary node and other replicas, carrying out corresponding verification and operations and finally sending the consensus results back to the client [18]. However, there will be a lot of communication between nodes, which will affect the consensus efficiency. In this paper, based on the application scenario of consortium blockchain, we have modified the PBFT algorithm in the following aspects:

4.1. Consistent Protocol Simplification. In the absence of Byzantine nodes, a simplified consistency protocol is adopted to reduce communication traffic between nodes, as shown in Figure 4. The implementation process of the simplified consistency protocol is as follows:

(1) CS-request period

Similar to the request phase of the PBFT algorithm, the client sends a request message to the primary node. The message format is <CS-request, x, t, c > , where x is the main content requested by the client, t is the timestamp, and c is the identity information of client C.

(2) CS-preprepare period

After receiving the request message x, the primary node assigns a sequence number n to the received x and then generates a prepreparation message. The message format is << CS-preprepare, v, n, d, e > , c, x > . v is the view number, d is the hash calculation result of x, and n is the message number. c is the node integral data, which is used to change the class of the node, and e is the hash calculation result of c. Then, the primary node sends the preprepared messages to all nodes. The consensus node needs to verify the message content. If the verification passes, it will enter the next stage. Otherwise, it will change the view and replace the primary node.

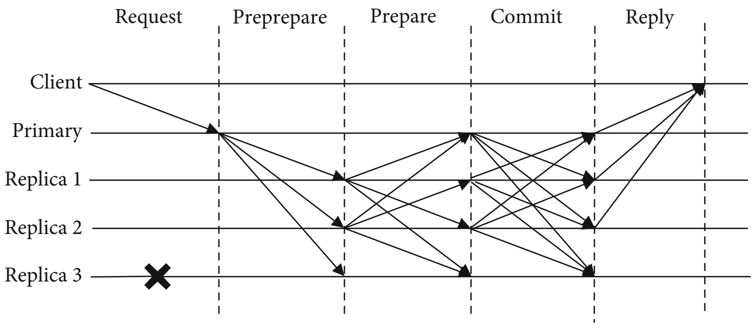


Figure 3: PBFT consistency protocol.

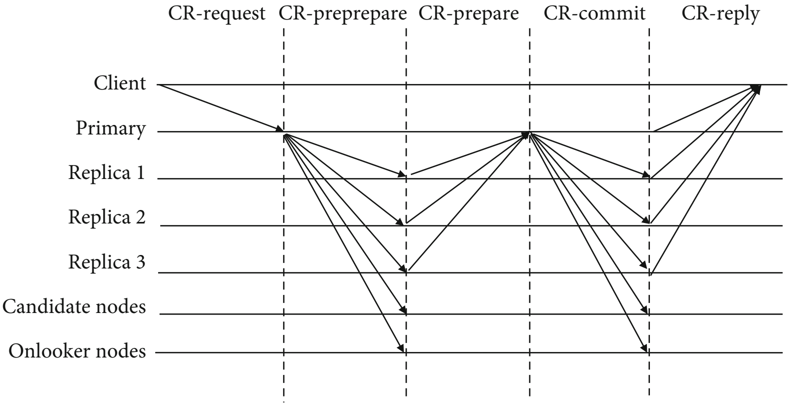


Figure 4: Simplified consistency protocol.

(3) CS-prepare period

After verifying the prepared message, the consensus node will judge the information in the verification certificate, including the correctness of blockchain transactions, block header information, and block height. Then, judging whether the integral data in the prepared message is the same as the local integral data, and if not, update the local integral information C. Then, a feedback message is generated and sent to the primary node, which format is <CS-back, $v, n, d, i>$, where i is the number of the node that sending the message.

(4) CS-commit period

If the primary node receives the feedback information sent by $2f + 1$ consensus nodes, and all the feedback information is the same, the primary node will package the feedback information and broadcast it to all nodes in the network. The message format is <CS-commit, $v, n, d, a>$. a indicates that the primary node has confirmed. If the primary node does not receive all the approval information, it will enter the complete consistency protocol process.

(5) CS-response period

Replicas verify whether the approval information of other nodes is correct. If all nodes have received block information, the transaction information will be added to the local memory. Class C nodes and class D nodes only receive consistent results, but do not give feedback.

4.2. Node Credit Score Mechanism. Selecting the primary nodes in the PBFT algorithm is random. It selects the primary nodes according to number sequence, which is more likely to appear malicious nodes because they are checked. If the malicious nodes are found, the view switching protocol will replace the primary nodes, resulting in a large amount of network communication overhead. In this paper, the PBFT algorithm is improved by node credit score mechanism.

Consortium blockchain requires participants to be authenticated before joining so its credibility and stability are more guaranteed than public blockchain. We apply node credit score mechanism in consortium blockchain. Nodes are selected according to credit scores to ensure higher computing power, wider bandwidth, and stability, which also improved the long-term operating efficiency of the system. With the running of the system, the credit scores and the proportion of malicious nodes in CSPBFT decrease continuously, thus effectively identifying and eliminating malicious nodes as well as ensuring that the algorithm can execute simplified consistency protocol most of the time. The algorithm takes the comprehensive strength of nodes as the initial score basis.

According to the credit scores, nodes are classified into class A, class B, and class C nodes. Class A node has the highest credit rating and takes priority as the primary node.

Table 2: Comparison of node permissions.

Credit rating	Taking priority as the primary node	Can act as the primary node	Taking priority as the replicas	Can act as the replicas	Getting consensus results
A	√	√	√	√	√
B	×	√	√	√	√
C	×	×	×	√	√
D	×	×	×	×	√

Secondly, class B nodes participate in consensus as replicas. It also can participate in the election of primary nodes when there is insufficient class A nodes. The node that just joined the system is class C node. The credit rating of class D nodes is too low, so they are not allowed to participate in consensus, but need to accept consensus results. Class C nodes are candidate nodes. When malicious nodes appear in the consensus nodes, the scores of malicious nodes decrease until they are excluded. According to the credit score situation, one node is selected from the class C nodes to join the class B node. If a node successfully participates in block generation, its credit score will increase by 1 point. On the contrary, if a node fails to generate blocks, 5 points of its score will be deducted. When the credit score is lower than 60, a node becomes class D node, which is not allowed to participate in consensus but can accept consensus results. The level of nodes will change in class A, class B, class C, and class D due to their behaviors. Class A and class B nodes are consensus nodes. According to the consistency protocol of the PBFT algorithm, when the client receives more than $f + 1$ consistent messages, it can be considered that the request is successfully executed, so the number of class A nodes is set to $f + 1$ and class B nodes is set to f. Class C nodes, as candidate nodes, do not participate in the system consensus process, whose number is uncertain. Class D nodes, as onlooker nodes, are not allowed to participate in consensus because of their low integral. They only accept consensus results so their number is uncertain. In the experiment, both the number of class C and class D nodes is set to $f/2$. Comparison of node permissions is shown in Table 2.

5. Design and Implementation

As shown in Figure 1, the hybrid architecture has six specific phrases with 14 steps illustrated below. The notations are given in Table 3. It mainly includes the following steps:

5.1. Initialization. At this stage, these entities are initialized: RFID readers, RFID tags, blockchain nodes for different departments, IPFS in edge nodes, and CP-ABE encryption module. Blockchain nodes' access control levels are detailed in Table 4.

The data structure S is defined within a single block. It consists of H_i, public data, and encrypted private data. Public data includes baggage tracking information and baggage request data. Encrypted private data refers to basic baggage information encrypted by CP-ABE. Only the user whose private key completely matches the access control policy can decrypt and obtain passenger information.

Table 3: Notations.

Symbol	Description
S	Detail data of a block
p_i	Encryption policy
CT_i	Encrypted basic baggage information
H_i	IPFS address hash
SK_i	Private key of departments in different airlines

Table 4: Access control levels of blockchain nodes.

Blockchain nodes	Access	Consensus
Check-in department	Write and read	Yes
Security check department	Write and read	Yes
Sorting department	Write and read	Yes
Trucking loading department	Write and read	Yes
Air loading department	Write and read	Yes
Arrive department	Write and read	Yes
Passenger	Read	No
Administrator	Write and read	Yes

5.2. Encrypt the Basic Baggage Information

Step 1. The CP-ABE encryption module is initialized, and the corresponding private key is assigned according to the attributes of each part of the airport. For example, we set the property of check-in department in airport B to

```
propertyProperty 1 {
        {visitor Sorting 'airline =861202' 'department =05' 'identity =2001192'}
}
```

We set the property of sorting department in airport A to

```
propertyProperty 2 {
        {adminCheckin 'airline =861107' 'department =01' 'identity =1154442'}
}
```

Step 2. Before sharing data, airport A needs to build an encryption policy to achieve access control. We specify the encryption policy p_i which allows airport A to decrypt while airport B cannot as Rule1. After encrypted basic baggage information according to p_i, we get CT_i.

```
rule Rule1 {
        {(admin and (airline<861200 or Checkin))}
        {or (admin and 2 of (department >=01, identity >=1154442, Sorting))}
}
```

5.3. CT_i Is Uploaded to IPFS in Edge Nodes

Step 3. CT_i is uploaded to the IPFS cluster to ensure not only secure data storage but also colossal storage capacity.

Step 4. IPFS returns the address hash H_i to airport A. CT_i can be queried in IPFS by H_i.

5.4. H_i, RFID ID, and Data Keywords Are Recorded in Blockchain

Step 5. After airport A got H_i from the IPFS cluster, the RFID ID, data keywords, and H_i are uploaded to the blockchain together. Airport A can query the records in the blockchain according to the RFID ID.

Step 6. The consortium blockchain network returns the inquiry result according to the requirements of airport A.

5.5. Baggage Tracking and Complete Baggage Information Are Automatically Formed and Stored in Blockchain

Step 7. Airport A encodes the RFID ID, data keywords, and address hash and stores them in the user area of the RFID tag.

Step 8. Taking six departments as nodes, we build multichannel consortium chain to realize physical isolation for different companies' businesses as well as coarse-grained privacy protection. The program that automatically stores data in blockchain based on node-serialport, and RFID is deployed on each node. RFID readers were also deployed on the nodes. Node-serialport is a package of Node.js, which is used to read and write serial port data. It is the way to communicate with the RFID reader.

Step 9. When the RFID tags pass through six nodes, there are mainly three steps:

RFID readers receive the data from the user area of the RFID tag, which decodes and intercepts basic baggage information.

The program automatically obtains baggage tracking information, blends it with basic baggage information, and requests to invoke the smart contract.

Smart contract compares data summaries of requests and records in the blockchain. If they are consistent, it allows data to be stored in blockchain.

5.6. Data Sharing and Access Control

Step 10. Airport B can query baggage handling status according to RFID ID or flight number, respectively. Meanwhile, the smart contract will count the baggage tracking request data, and the query times of each piece of baggage will be permanently recorded in the blockchain.

Table 5: System and module versions.

System and module	Versions
Ubuntu	18.04.3 LTS
Hyperledger Fabric	1.3
Docker	19.03.4
Docker-compose	19.03.4
Go	1.12.10
IPFS	0.4.13

Step 11. The consortium blockchain network returns the result to airport B, and airport B can obtain the baggage handling status.

Step 12. In case of lost or damaged baggage, if B demands basic baggage information, it should query its H_i of IPFS in the blockchain according to the RFID ID and then inquires CT_i in IPFS cluster through H_i.

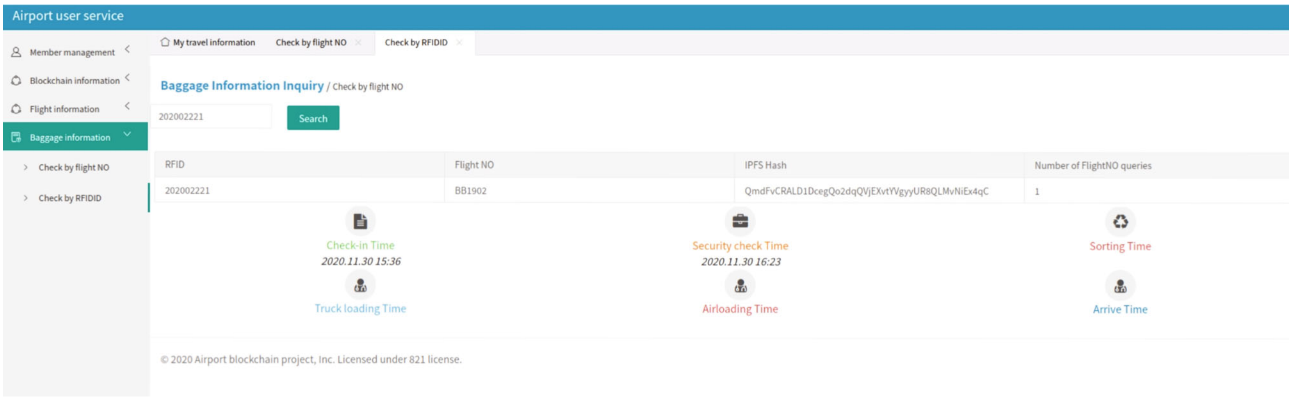
Step 13. The IPFS cluster finds the CT_i and returns query results, and airport B decrypts CT_i according to its private key SK_i. Only when SK_i accord with access policy p can airport B obtain the basic baggage information.

Step 14. Passengers can query real-time baggage tracking information according to the RFID ID. Meanwhile, the query times of each piece of baggage will also be recorded in the blockchain. Administrators can conduct not only batch or accurate retrieval, but also data interaction.

6. Experiments and Evaluation

6.1. Experimental Setup. In this section, we implement experiments to evaluate the performance of the proposed hybrid baggage tracking system that was prototyped on the Hyperledger Fabric. The specific configuration of the experimental platform and the experimental environment is as follows: the system is deployed on 2 hosts with intel corei7-9700@3.00 GHz processor and corei7-5500@2.40 GHz processor, 4 GB RAM, and we have two RFID readers. The system and module versions are shown in Table 5.

6.2. Experimental Results. (1)*Query Real-Time Baggage Tracking Information According to the RFID ID.* In the first experiment, we log into the blockchain network as administrators. Not only can we conduct bulk queries according to flight number but also accurate queries by RFID ID. We can get the time of baggage arrival in each department, flight number, and IPFS address hash corresponding to basic baggage information, as shown in Figure 5. At the same time, this retrieval behavior will be permanently recorded by blockchain network, and the number of queries will plus one.



Baggage Information Inquiry / Check by flight NO

202002221

RFID	Flight NO	IPFS Hash	Number of FlightNO queries
202002221	BB1902	QmdFvCRALD1DcegQo2dqQVjEXvtYVgyyUR8QLMvNiEx4qC	1

Check-in Time
2020.11.30 15:36

Security check Time
2020.11.30 16:23

Sorting Time

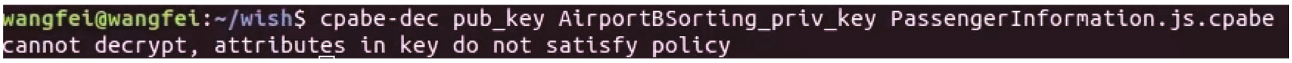
Truck loading Time

Airloading Time

Arrive Time

© 2020 Airport blockchain project, Inc. Licensed under 821 license.

Figure 5: Query real-time baggage tracking information by RFID ID.



```
wangfei@wangfei:~/wish$ cpabe-dec pub_key AirportBSorting_priv_key PassengerInformation.js.cpabe
cannot decrypt, attributes in key do not satisfy policy
```

Figure 6: The unauthorized airport department could not decrypt.

(2)*Test of Access Control*. In the second experiment, only when the department's property conforms to the access control policy can it decrypt basic baggage information downloaded from IPFS. For instance, the access policy we developed in attribute-based access control allows all departments in airport A to decrypt basic baggage information while airport B cannot, as shown in Figure 6.

(3)*System Performance*. We conducted four rounds of tests to measure the average transaction processing time, the average time for RFID data to store in blockchain, and the average time to reject retrieval that does not comply with the access control policy. To evaluate the scalability of our system, we also carry out extensive system performance evaluations by increasing the number of baggage and departments. The average transaction processing time from 4 rounds of tests remains at around 0.40 s when the number of departments increased from 4 to 20 as shown in Figure 7. The average time to reject retrieval that does not comply with the access control policy from 4 rounds of tests remains at around 50 ms as shown in Figure 8. The average time for RFID data to store in the blockchain from 4 rounds of tests remains at around 0.58 s when the number of baggage increased from 6 to 1000 as shown in Figure 9. The results of the experiments show that when the number of baggage and departments increased, transaction time and response time did not change significantly, which demonstrates that the performance of the proposed system is scalable. The prototype system can realize baggage processing information stored in the blockchain in real time and automatically as well as produce response in a few hundred milliseconds, which makes it suitable in practical baggage tracking systems.

6.3. CSPBFT Performance. In this section, we compare PBFT and CSPBFT consensus algorithms in terms of communication latency, communication overhead, and operational efficiency through experiments. This experiment simulates a multinode consortium blockchain system by JAVA.

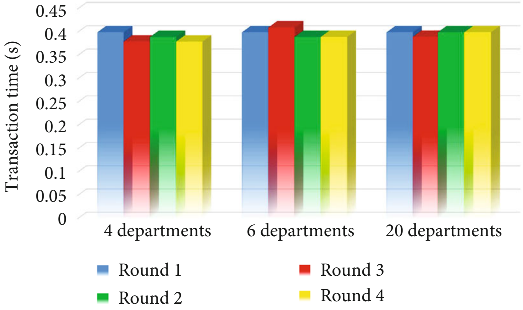


Figure 7: The average transaction processing time.

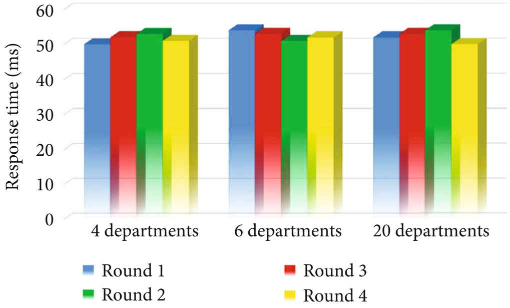


Figure 8: The average time to reject retrieval that does not comply with the access control policy.

(1) *Communication Latency*. Communication latency refers to the time interval between the client sending a transaction request to the primary node and the client confirming the completion of consensus, which is an essential parameter for evaluating the performance of consensus algorithm. Reducing communication latency can improve the efficiency and practicability of the system. In this experiment, the total number of nodes in the system is taken as an

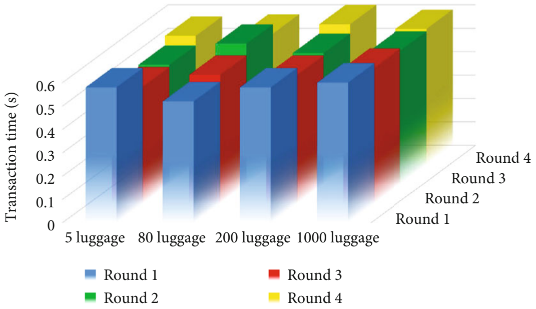


FIGURE 9: The average time for RFID data to store in blockchain.

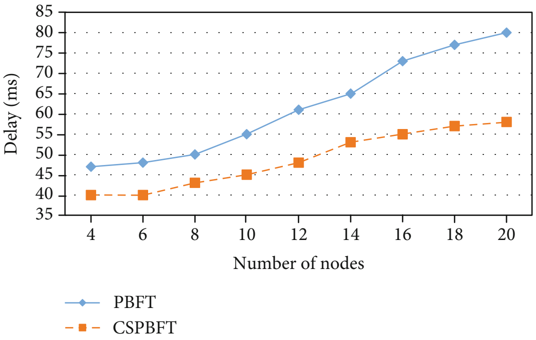


FIGURE 10: Communication latency contrast without Byzantine nodes.

experimental variable. The number of nodes increases from 4 to 20. The step size is 2. Transactions are carried out under different node numbers, and the average value in different states is taken as the final value of the communication latency. As shown in Figure 10, in the absence of Byzantine nodes, the CSPBFT algorithm implements a simplified consistency protocol, which is superior to the PBFT algorithm in communication latency. With the increase of the number of nodes, the CSPBFT algorithm has lower communication latency growth rate and better stability. However, in the presence of Byzantine nodes, the communication latency of the CSPBFT algorithm increases obviously due to the switching of consensus protocols, as shown in Figure 11. However, through the node credit score mechanism, the fault nodes in the system can be effectively identified and eliminated, so that the algorithm can execute the simplified consistency protocol most time, thus improving the long-term operational efficiency of the system.

(2) *Communication Overhead*. The PBFT algorithm has three core stages, which are preprepare stage, prepare stage, and commit stage. The maximum tolerance number of fault nodes in the blockchain is f, and the number of nodes i in the system should not be less than $3f+1$. Considering the consensus efficiency of the system, i is usually $3f+1$. Assuming that all nodes in the network communicate normally, the calculation of total message volume in the three-stage consensus process of PBFT is shown in Formula (5).

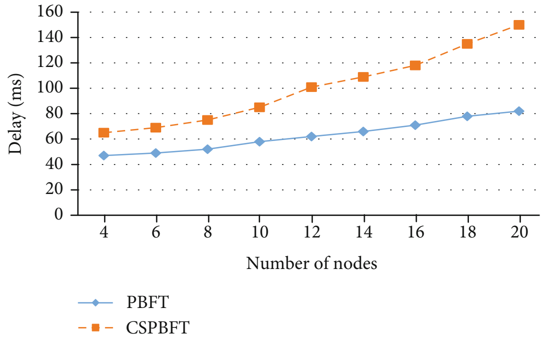


FIGURE 11: Communication latency contrast when Byzantine nodes exist.

$$M_{\text{pbft}} = 6f(3f+1). \tag{5}$$

As for CSPBFT, in the main four-stage consensus process of preprepare stage, prepare stage, commit stage, and response stage, when all nodes in the network communicate normally, the calculation of message volume to generate a new block is as follows.

$$M_{\text{cspbft}} = 9f. \tag{6}$$

When the primary node fails and needs view conversion, the replicas need to communicate view conversion information through pairwise interaction. For PBFT, the communication amount is $9f^2$ at this time. After completing the view change, the primary node needs to send a view confirmation message to the replicas, and the communication amount is $3f$. Combined with the view change probability p, the average total communication amount of the PBFT algorithm is as follows:

$$C_{\text{pbft}} = 18f^2 + 6f + p\left(9f^2 + 3f\right). \tag{7}$$

For CSPBFT, the average total communication amount is

$$C_{\text{cspbft}} = 9f + p\left(4f^2 + 2f\right). \tag{8}$$

Therefore, the ratio Q of communication amount between CSPBFT and PBFT is as follows:

$$Q = 18f^2 + 6f + p\left(9f^2 + 3f\right)\left\{9f + p\left(4f^2 + 2f\right)\right\}. \tag{9}$$

The visual graph of Q is obtained by MATLAB. The value of p ranges from 0 to 1, the step size is 0.1, the value of f ranges from 3 to 33, and the step size is 3, as shown in Figure 12.

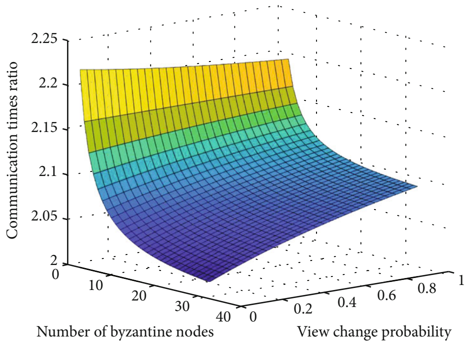


FIGURE 12: Communication overhead contrast.

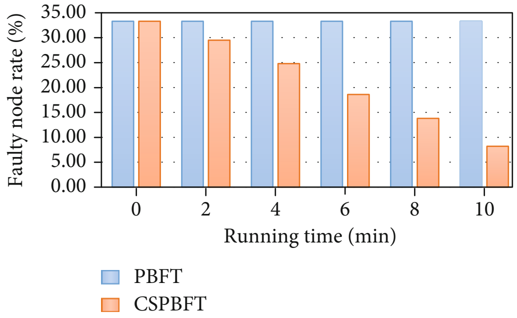


FIGURE 13: Faulty node rate.

It can be seen from the figure that no matter how the values of p and f change, the value of Q is always less than 1. This suggests that the communication volume of CSPBFT is always less than PBFT. With the increase of nodes, the Q decreases gradually, which indicates that the communication overhead performance of the CSPBFT algorithm is still better than PBFT in multinode environment. In addition, the CSPBFT algorithm introduces node credit score mechanism to evaluate node behavior, which reduces the probability of Byzantine node. Therefore, the CSPBFT algorithm has better performance in communication overhead in the practical process.

(3) *Operating Efficiency*. One of the purposes of CSPBFT design is to improve the long-term operational efficiency of the system. With the running of the system, the credit scores and the proportion of malicious nodes in CSPBFT decrease continuously, thus effectively identifying and eliminating malicious nodes in the system. However, the faulty node rate in PBFT does not change. Figure 13 shows the change of the faulty node rate between PBFT and CSPBFT for a long time. Therefore, through simplified consistency protocol, CSPBFT can generate blocks more efficiently than PBFT.

7. Conclusion

In this paper, we have proposed a system named BEI-TAB that utilized RFID combined with consortium blockchain to realize the real-time tracking information automatically stored in the blockchain, which not only avoids data leakage but also improves the industrialization level of the airport. In addition, we took advantage of multichannel architecture realized physical isolation of different businesses and coarse-grained privacy protection. At the same time, we utilized CP-ABE and IPFS to store basic baggage information in edge nodes so as to improve the scalability of blockchain and provide fine-grained privacy protection. To this end, we have deployed a testbed with both software and hardware to evaluate the performance of transaction processing time and speed. The experiments showed that our system is scalable, which makes it suitable to be incorporated in secured and real-time baggage tracking. Besides, we improved the PBFT algorithm to CSPBFT which adopted a simplified consistency protocol to reduce the communication traffic between nodes in the absence of Byzantine nodes. The Byzantine nodes in the system are effectively identified and excluded by the node credit score mechanism, so that the algorithm can execute the simplified consistency protocol most time, thus improving the long-term operational efficiency of the system.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] IATA, “Resolution: RFID baggage tracking set for global deployment,” https://www.iata.org/en/pressroom/pr/2019-06-02-05/.

[2] H. Ding, X. Li, Y. Cai, B. Lorenzo, and Y. Fang, “Intelligent data transportation in smart cities: a spectrum-aware approach,” *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2598–2611, 2018.

[3] H. Ding, C. Zhang, Y. Cai, and Y. Fang, “Smart cities on wheels: a newly emerging vehicular cognitive capability harvesting network for data transportation,” *IEEE Wireless Communications*, vol. 25, no. 2, pp. 160–169, 2018.

[4] T. Ahmed, T. Calders, and T. B. Pedersen, “Mining risk factors in RFID baggage tracking data,” in *2015 16th IEEE International Conference on Mobile Data Management*, pp. 235–242, Pittsburgh, PA, USA, June 2015.

[5] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, “Integrated blockchain and edge computing systems: a survey, some research issues and challenges,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1508–1532, 2019.

[6] H. Huang, J. Lin, B. Zheng, Z. Zheng, and J. Bian, “When blockchain meets distributed file systems: an overview, challenges and open issues,” *IEEE Access*, vol. 8, pp. 50574–50586, 2020.

[7] A. Singh, S. Meshram, T. Gujar, and P. R. Wankhede, “Baggage tracing and handling system using RFID and IoT for airports,” in *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pp. 466–470, Pune, India, December 2016.

[8] M. S. Jerry, M. M. Vijay, and T. K. Tulshiram, "Carousel security management and cargo deck tracking of passenger baggage using wireless technology," in *2016 IEEE Bombay Section Symposium (IBSS)*, pp. 1–6, Baramati, India, December 2016.

[9] M. Johnson and A. Gilman, "Real-time baggage tracking using a modified background subtraction algorithm," in *2012 19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, pp. 200–204, Auckland, New Zealand, 2012.

[10] Q. Gao and P. Liang, "Airline baggage appearance transportability detection based on a novel dataset and sequential hierarchical sampling CNN model," *IEEE Access*, vol. 9, pp. 41833–41843, 2021.

[11] S. Wang, Y. Zhang, and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems," *IEEE Access*, vol. 6, pp. 38437–38450, 2018.

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

[13] A. Juels, "RFID security and privacy: a research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[14] R. Norvill, B. B. FizPontiveros, R. State, and A. Cullen, "IPFS for reduction of chain size in Ethereum," in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1121–1128, Halifax, NS, Canada, 2018.

[15] H. Ding, Y. Guo, X. Li, and Y. Fang, "Beef up the edge: spectrum-aware placement of edge computing services for the Internet of Things," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2783–2795, 2019.

[16] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, Berkeley, CA, USA, May 2007.

[17] K. Lei, Q. Zhang, L. Xu, and Z. Qi, "Reputation-based Byzantine fault-tolerance for consortium blockchain," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 604–611, Singapore, December 2018.

[18] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo, "Efficient Byzantine fault-tolerance," *IEEE Transactions on Computers*, vol. 62, no. 1, pp. 16–30, 2013.