Journal of Quantum Computing
DOI:10.32604/jqc.2021.016775
Article

Tech Science Press

IOTA-Based Data Encryption Storage and Retrieval Method

Hongchao Ma[1,*], Yi Man[1], Xiao Xing[2], Zihan Zhuo[2] and Mo Chen[3]

[1]Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2]National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, 100029, China
[3]Beijing Information Science and Technology University, Beijing, 100085, China
[*]Corresponding Author: Hongchao Ma. Email: mahongchao1994@163.com
Received: 20 May 2021; Accepted: 16 August 2021

Abstract: At present, the traditional blockchain for data storage and retrieval reflects the characteristics of slow data uploading speed, high cost, and transparency, and there are a lot of corresponding problems, such as not supporting private data storage, large data operation costs, and not supporting Data field query. This paper proposes a method of data encryption storage and retrieval based on the IOTA distributed ledger, combined with the fast transaction processing speed and zero-value transactions of the IOTA blockchain, through the Masked Authenticated Messaging technology, so that the data is encrypted in the data stream. The form is stored in the distributed ledger, quickly retrieved through the field index mechanism established by the data form, and the data operation is carried out on the chain. Experimental results show that this system has high storage, encryption and retrieval performance, and good practicability.

Keywords: IOTA; Masked Authenticated Messaging; storage and retrieval

1 Introduction

The NXT community first put forward the idea of combining DAG and blockchain [1] to solve the efficiency problem of the blockchain with chain structure [2]. Gao et al. summarized the scalability bottlenecks of blockchain [3], and proposed DAG-based solutions. At present, there are performance and functional problems in blockchain projects, which can be summarized as follows: First, due to the block generation mechanism and the storage structure of the chain, only one chain can exist in the whole network, leading to the generation of data blocks that cannot be executed concurrently, resulting in too long time of data storage; Second, due to the currency strategy of the blockchain, each transaction requires a certain fee, which acts as an incentive to the miner, making the data on the chain more costly; Third, the current distributed ledger is transparent to users and is not suitable for the preservation of private data. Fourthly, the data storage is mostly the transaction information of fixed format, which does not support the query through the data fields in the transaction. The data operability is low, and there is no traditional database query function.

The contribution of this paper is to propose a scheme for encrypting storage and fast retrieval of data files. There is no concept of data block in IOTA network, which is based on the structure of directed acyclic graph DAG, and the transaction speed is faster [4]. IOTA networks allow zero-value transactions and support the transfer of large amounts of data. For private data, by masking authentication messages using IOTA, the data can be stored in the IOTA network as an encrypted data stream [5]. The scheme proposed in this paper uses IOTA distributed ledger to accelerate the speed of data uploading and retrieval; The integrality and traceability of system data are guaranteed by the immutable property of distributed ledger; Use masking authentication messaging so that data on the chain is encrypted and stored as a stream of data; By establishing index and data channel mechanism, distributed ledger data can be quickly queried through data fields.

2 Related Work

In order to solve the problem of blockchain network storage and query in actual situations, many scholars have proposed many related methods.

In [6], Jiao proposed a blockchain database system framework, which uses an immutable index based on hash pointers to quickly retrieve the data in the block based on the index to achieve blockchain query verification testing the read and write performance of the database.

In [7], Florea uses the Raspberry Pi to collect data, stores and shares data in the IOTA distributed ledger. It is set that each message is assigned a tag, and the required data can be searched in the distributed ledger according to the tag.

In [8], Hawig et al. applied the IOTA distributed ledger to the information storage of medical and health equipment, using the IOTA Masked Authenticated Messaging to store data on the chain to prove It has the ability of IOTA to store, verify and immutable data.

In [9], Lindvall explained that IOTA uses a Merkle tree-based signature scheme and hash-based ternary and symmetric encryption to encrypt data to ensure the authenticity and confidentiality of data.

In [10], Sun et al. uses IOTA for the storage of environmental monitoring equipment data. The data is transmitted to the IOTA Tangle through the Masked Authenticated Messaging.

In [11], Zichichi et al. proposed a variety of blockchain cooperation methods, in which IOTA and IPFS are used to store and verify data from sensors or users themselves, and Ethereum is developed as a smart contract platform for coordinating data sharing and supply.

In [12], the open source project in the official IOTA Github uses IOTA to track various data of port cargo, and records it in the unalterable Tangle through the Masked Authenticated Messaging, ensuring the authenticity and integrity of the data, and simplify the work process.

3 Solution Architecture

The scheme proposed in this paper is divided into four main levels: client application layer, data index layer, index connection layer, and data storage layer.

At client application layer, users perform data storage and retrieval functions on the WEB client, and perform corresponding operations on the server side through HTTP requests according to their needs.

At data storage layer, use IPFS as the file storage database, which is a peer-to-peer (P2P) distributed file system [13]. The file is stored in the IPFS system, and the Hash value of the returned file is used as an index for downloading files from IPFS. Since IPFS does not have the ability to prevent tampering, the function of checking file consistency is added to the data index layer.

At data index layer, use IOTA Masked Authenticated Messaging for data on-chain. The Masked Authenticated Messaging is a module as the second layer data communication protocol of the IOTA network, which extends the functions of IOTA transactions and provides hash-based signature authentication and integrity verification [14]. Masked Authenticated Messaging sends each message to a different address, but there are detailed useful information to connect them [15].

At index connection layer, with the help of MySQL database or log file, establish a one-to-one or one-to-many mapping relationship between data and address, and quickly query the address of the distributed ledger through the data form.

4 Solution Design

This program mainly includes two methods, and different methods can be used for different data storage situations. The first option is mainly for the case where the data volume is small and fast retrieval is required; the second option is mainly suitable for the case where the data volume is large and the retrieval time is not too strict.

4.1 Solution 1: Accurate Retrieval of Small Data Storage

Solution 1 realizes the scenario of small data storage. IPFS is used as the storage layer, and the data is encrypted and stored through the Masked Authenticated Messaging mechanism. With the help of MySQL form, the data file and the distributed ledger address are in a one-to-one correspondence relationship.

4.1.1 Data Storage

This scheme is applied to the storage of small data volume. This scheme builds MySQL index for each data and stores block address with MySQL form. Table 1 shows the data storage algorithm of this scheme, and Fig. 1 shows the principle diagram of data storage.

Table 1: Small data storage algorithm

Solution 1: Data storage algorithm
1: # Data file preprocessing
2: fileHash = hash(File): Calculates the file Hash value for later verification
3: Input: File, fileHash, fileName, userName, operation
4: # Perform data storage
5: IPFSHash = ipfs.store(File): The file is stored in IPFS and returned the IPFS index
6: # Perform data on-chain and operation on-chain
7: app.post('/postIota', function (req, res)):
8: request={ fileHash, fileName, userName, operation };
9: # Modify the data to JSON format
10: jsonData = asciiToTrytes(request);
11: addData(jsonData):
12: iotaMam.attach(jsonData);
13: MySQLConnection.insert(addSql, SqlParams);
14: Output: “success” or “fail”

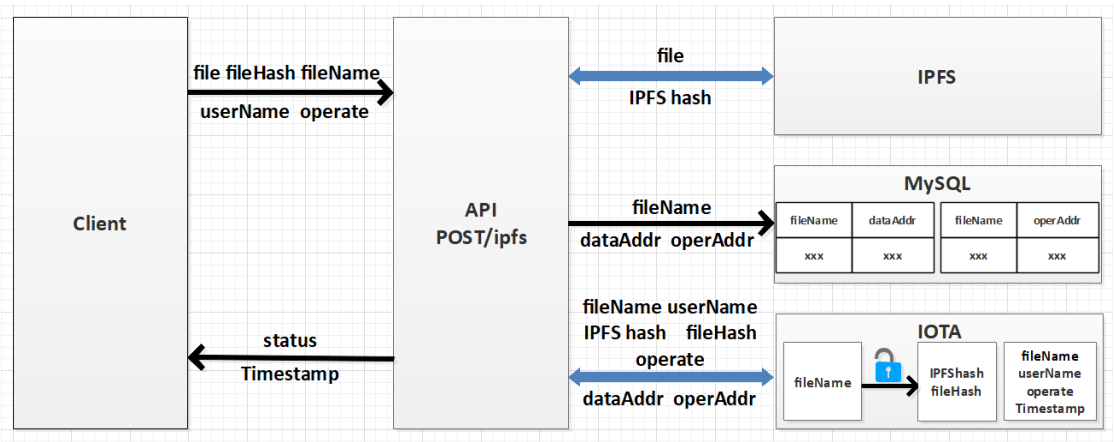

Figure 1: Solution 1: Data storage principle diagram

4.1.2 Data Retrieval

This scheme is based on the above storage channel mechanism and uses the algorithm proposed in this paper to replace the traditional ergodic search. The scheme can be used for random data retrieval function. Fig. 2 shows the principle diagram of data retrieval, and Table 2 shows the algorithm description of data retrieval.

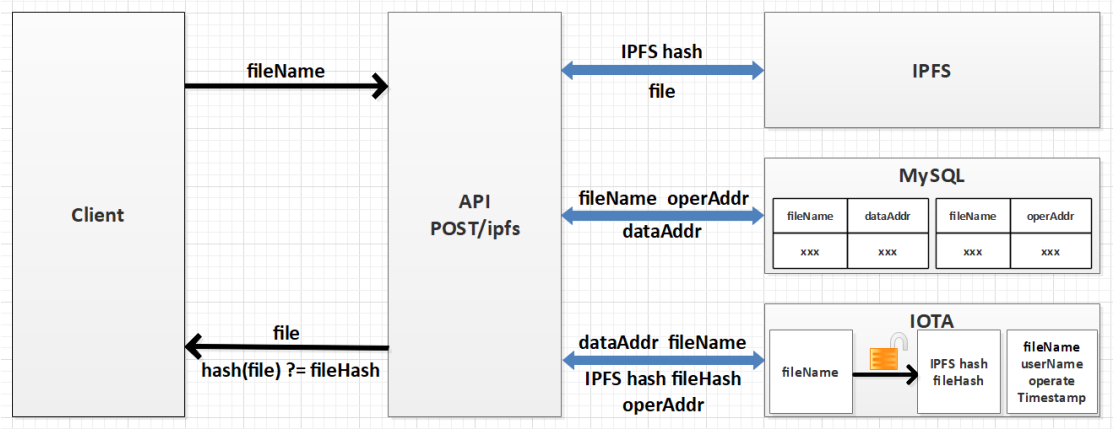

Figure 2: Solution 1: Data retrieval principle diagram

Table 2: Retrieval algorithm for small data volume

Solution 1: Data retrieval algorithm
1: # Search by file name or file name field
2: Input: fileName
3: # Perform data retrieval
4: app.post('/findIota', function (req, res):
5: iotaAddress = MySQLConnection.query(querySql, SqlParams);
6: findData(iotaAddress):
7: jsonData = iotaMam.fetch(iotaAddress, mamType, mamSecret);
8: # Decrypt and parse the data
9: data = JSON.parse(trytesToAscii(jsonData));
10: IPFSHash = data.ipfsHash; fileHash = data.fileHash;
11: # Perform download data
12: File = ipfs.download(IPFSHash): Download data files according to IPFS index
13: # Calculate the file hash and verify whether the file has been tampered with
14: curFileHash = hash(File);
15: If curFileHash == fileHash:
16: return File
17: Output: File or “fail”

4.2 Solution 2: Fast Retrieval of Large Data Storage

The second Solution realizes the scenario of large data storage. IPFS is used as the storage layer, and the entire data stream can be encrypted through the IOTA Masked Authenticated Messaging transfer and stored in their own independent addresses. In essence, it works like a radio, and only those with the correct frequency can listen.

Fig. 3 shows the data flow of the single-channel Masked Authenticated Messaging. The black blocks are the part of the stored data, scattered in the distributed ledger, through the Masked Authenticated Messaging, connected together at an additional level, such as the red connecting line, Knowing the channel entry address or end address, you can store or retrieve the entire message stream, which is convenient for storing and retrieving data in the IOTA distributed ledger.

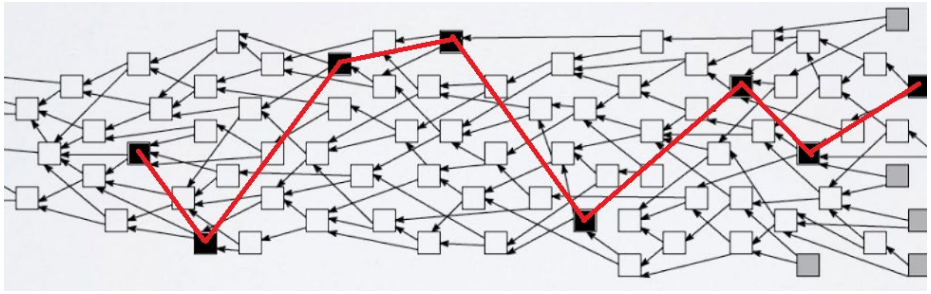

Figure 3: Data flow of single-channel Masked Authenticated Messaging

4.2.1 Data Storage

The solution is suitable for the storage of Marine data. The algorithm establishes a storage channel based on IOTA mask authentication message for each type of data, and then classifies the data into different data channels. This algorithm not only simplifies the complexity of the address information management of the data storage block, but also can classify the data files and improve the efficiency of subsequent retrieval work, which is very suitable for the scene of classified file storage. Table 3 shows the principle diagram of data storage, and Fig. 4 shows the principle diagram of data storage.

Table 3: Storage algorithms for large amounts of data

Solution 2: Data storage algorithm
1: # Data file preprocessing
2: fileHash = hash(File): Calculates the file Hash value for later verification
3: Input: File, fileHash, fileName, userName, operation
4: # Perform data storage
5: IPFSHash = ipfs.store(File): The file is stored in IPFS and returned the IPFS index
6: # Perform data on-chain and operation on-chain
7: # Obtain the channel address corresponding to the data file
8: XxxAddr = LogFile.get(fileName);
9: app.post('/postIota', function (req, res)):
10: request ={ fileHash, fileName, userName, operation };
11: # Modify the data to JSON format
12: jsonData = asciiToTrytes(request);
13: addData(jsonData):
14: message = iotaMam.create(XxxAddr, jsonData);
15: # Store in the corresponding Masked Authenticated Messaging channel
16: iotaMam.attach(message.payload, message.address);
17: mamState = JSON.stringify(message.state);
18: # Log file update channel address
19: LogFile.put(fileName, mamState.addr);
20: Output: “success” or “fail”

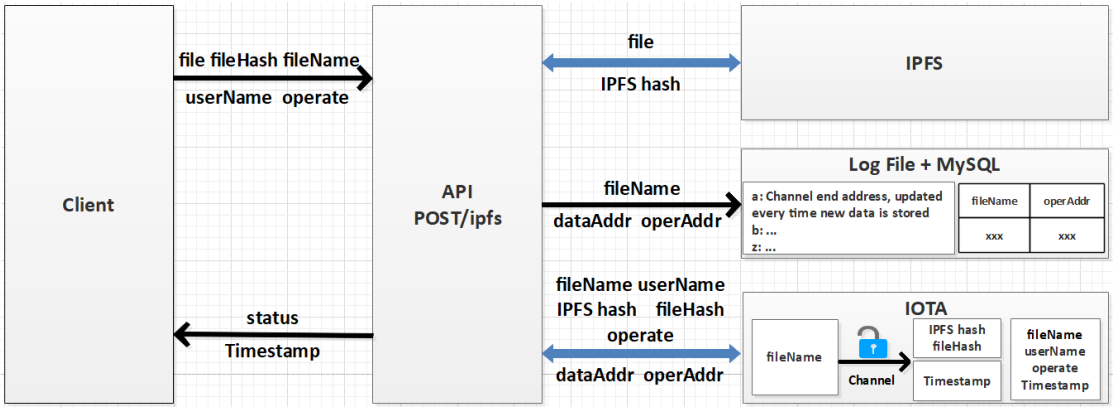

Figure 4: Solution 2: Data storage principle diagram

4.2.2 Data Retrieval

Based on the above storage channel mechanism, this scheme uses the algorithm proposed in this paper to replace the traditional ergodic search and carry out classified query on the data with different characteristics to improve the query efficiency. Table 4 shows the algorithm description of data retrieval, and Fig. 5 shows the principle diagram of data retrieval.

Table 4: Retrieval algorithms for large data volumes

Solution 2: Data retrieval algorithm
1: # Search by file name or file name field
2: Input: fileName
3: # Perform data retrieval
4: # Obtain the channel address corresponding to the data
5: XxxAddr = LogFile.get(fileName);
6: app.post('/findIota', function (req, res):
7: findData(iotaAddress):
8: value[] = iotaMam.fetch(iotaAddress, mamType, mamSecret);
9: value[].forEach(function(v, i):
10: # Decrypt and parse the data
11: messagei = JSON.parse(trytesToAscii(v));
12: If messagei.fileName == fileName
13: IPFSHash = messagei.ipfsHash; fileHash = messagei.fileHash;
14: # Perform download data
15: File = ipfs.download(IPFSHash): Download data files according to IPFS index
16: # Calculate the file hash and verify whether the file has been tampered with
17: curFileHash = hash(File);
18: If curFileHash == fileHash:
19: return File;
20: Output: File or “fail”

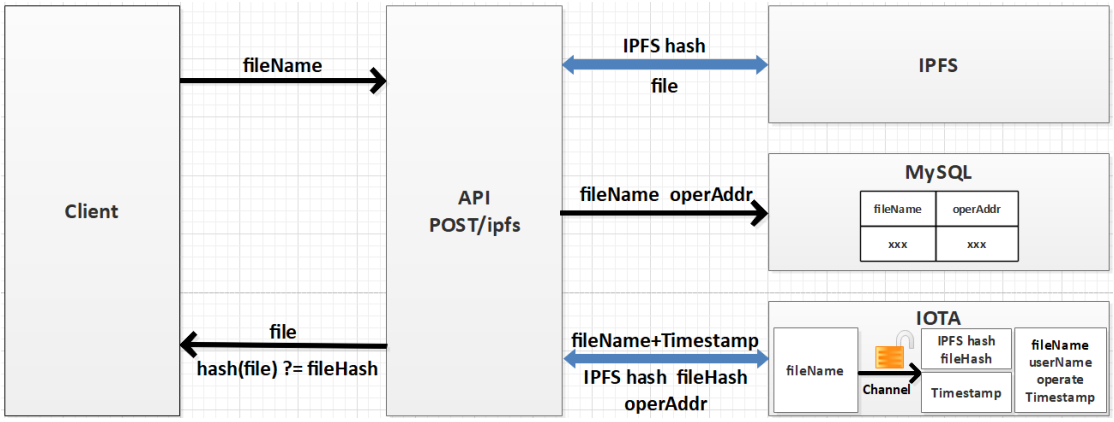

Figure 5: Solution 2: Data retrieval principle diagram

5 Performance Evaluation

The experimental platforms used in this experiment are Windows(Intel® Core™i7-8750H CPU@2.20 GHz 2.21 GHz) and Ubuntu18.04 virtual machine, Use private-iota-testnet to build the IOTA private chain test network, use mam.client.js-master to write applications, interact with the IOTA private chain network, and complete the corresponding functions of the solution in this article.

5.1 Performance of Data Uploading and Retrieval

In order to verify the performance of the solution proposed in this paper, we use the way of comparing with the existing scheme to conduct experiments in two aspects of the time consumed by data storage and data retrieval. Test for a small data storage solution, here called solution A; the other is a researched IOTA data on-chain solution, here called solution B. Solution B interacts with the IOTA network through the JavaScript file of tangly.js (For details, see https://github.com/ BitBounti/tangly). It can realize the function of data on the chain and the function of retrieving data through data fields.

Fig. 6 shows the relationship between the average time of data uploading and the amount of data. Solution A is significantly better than solution B in terms of uploading time. The uploading time of solution A is basically not affected by the amount of data, and its performance is stable. Scheme B's chain time is greatly affected by the amount of data, and as the data increases, the performance becomes worse.

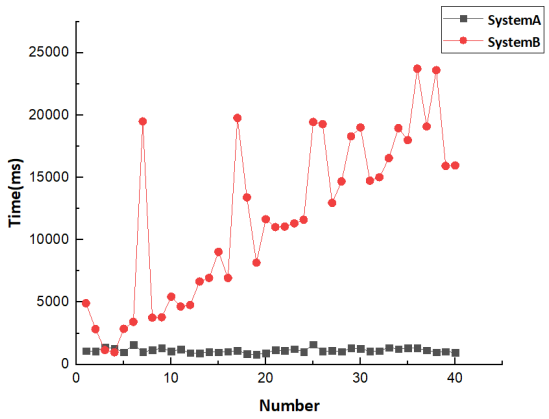

Figure 6: The relationship between the average time of data uploading and the amount of data

As shown in Fig. 7, the relationship between the average time of data retrieval and the amount of data. It can be seen from the figure that the retrieval time of Scheme A has nothing to do with the amount of data, and the performance can reach millisecond level, and the performance is superior; the retrieval time of

Scheme B is proportional to the amount of data, the larger the amount of data, the more time required, and the worse the performance.

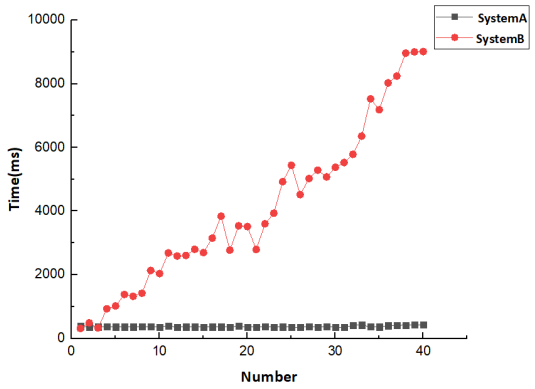

Figure 7: The relationship between the average time of data retrieval and the amount of data

5.2 Safety Analysis

The security analysis of this system mainly revolves around data integrity and data privacy.

1) Integrity: For data on the chain, the data can be stored in an immutable form through the IOTA distributed ledger, which strictly guarantees the integrity of the data and cannot be changed at will.

2) Privacy: For private encrypted data, a mask authentication transmission mechanism is adopted, the transaction address is encrypted through the root node of the Merkle tree, and the sidekey password encrypts the data. The data is stored in the distributed ledger in the form of an encrypted data stream. Only by knowing the root node and the password can the data be unlocked, which strictly guarantees the privacy of the data.

6 Concluding Remarks

This paper uses IOTA blockchain technology to propose a data encryption storage and retrieval scheme. Solve the problems of traditional blockchain data, such as slow speed, high cost, no support for private data encryption, and no field retrieval.

The disadvantage of this scheme is that the system function is relatively single, the client's request is not processed with high concurrency, and the simulation environment is only built in the private chain test network; the storage and retrieval of large data volume in the second scheme stays in the theoretical analysis process. Because of the need for more detailed and larger workload operations, no tests were performed. In the follow-up research, the system will be improved, and the performance of the second plan will be tested to improve the system performance.

Acknowledgment: The research of this article is supported by the National Key Research and Development Program "Biological Information Security and Efficient Transmission" Project, Project Letter No. 2017YFC1201204.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Yuan, "IOTA principle," in *DAG Blockchain Technology*, 1st ed., vol. 1. Bejing, China: Machinery Industry Press, 2018, pp. 156–186.

[2] Q. Du, *DAG May Be the Real Blockchain 3.0*. China: IOTA China Community, 2017. [Online]. Available: https://www.iotachina.com/dagblockchain.html.

[3] Z. F. Gao, J. L. Zheng, S. Y. Tang and Y. Long, "State of the art survey of consensus mechanisms on DAG-based distributed ledger," *Journal of Software*, vol. 31, no. 4, pp. 1124–1142, 2020.

[4] S. Popov, "The tangle," *White Paper*, vol. 1, no. 3, pp. 1–11, 2018.

[5] S. Pinjala and K. Sivalingam, *In the Internet of Things, Why These Two Features of IOTA Can Kill Other Competing Public Chains*. China: IOTA Community, 2019. [Online]. Available: https://www.iotachina.com/wulianwangzhongiotadezhelianggetexingweishenmekeyimiaoshaqitajingzhenggonglian.html.

[6] T. Jiao, "Blockchain database: A queryable and tamper-proof database," *Journal of Software,* vol. 30, no. 9, pp. 2671−2685, 2019.

[7] B. C. Florea, "Blockchain and Internet of Things data provider for smart applications," in *Mediterranean Conf. on Embedded Computing*, Budva, Montenegro, pp. 2671−2685, 2018.

[8] D. Hawig and C. Zhou, "Designing a distributed ledger technology system for interoperable and general data protection regulation–compliant health data exchange: A use case in blood glucose data," *Journal of Medical Internet Research*, vol. 21, no. 6, pp. 1–13, 2019.

[9] M. Lindvall, *How is Authenthicity and Confidentiality Maintained for MAM Channels on the IOTA Tangle?* Norwegian: The IMT1003 subject at the Norwegian University of Science and Technology, 2004. [Online]. Available: https://varden.info/doc.php?id=7.

[10] S. J. Sun andX. C. Zheng, "Indoor air-quality data-monitoring system: Long-term monitoring benefits," in *Sensors.* Basel, Switzerland: MDPI, 2019, pp. 1–18.

[11] M. Zichichi and S. Ferretti, "A distributed ledger based infrastructure for smart transportation system and social good," in *Annual Consumer Communications & Networking Conf.*, Las Vegas, NV, USA, 2020.

[12] IOTA Official, *Iota Cargo Tracking System*. China: IOTA China Community, 2018. [Online]. Available: https://www.iotachina.com/iota-trade-poc.html.

[13] J. Z. Wang, "Implementation of a data sharing platform based on blockchain and IPFS," Ph.D. dissertation, Zhejiang University of Commerce and Industry, 2018.

[14] I. Inhuman, *Masked Authenticated Messaging Detailed Introduction*. China: IOTA China Community, 2018. [Online]. Available: https://www.iotachina.com/iotamaskedauthenticatedmessage.html.

[15] P. Handy, *Introducing Masked Authenticated Messaging*. China: IOTA China Community, 2018. [Online]. Available: https://blog.iota.org/introducing-masked-authenticated-messaging-e55c1822d50e.