Journal of Medical Systems (2019) 43: 305
https://doi.org/10.1007/s10916-019-1430-2

TRANSACTIONAL PROCESSING SYSTEMS

Med-PPPHIS: Blockchain-Based Personal Healthcare Information System for National Physique Monitoring and Scientific Exercise Guiding

Tong Zhou[1,2] · Xiaofeng Li[1,2] · He Zhao[1]

Received: 26 February 2019 / Accepted: 29 July 2019 / Published online: 14 August 2019

Abstract
The dissemination of electronic medical data among professional personnel has been perceived to be an important breakthrough for the discovery of new technologies and therapies for curing diseases. However, in the current medical data management, it is difficult to share medical data due to the fragmentation of medical data and the lack of effective sharing methods. On the other hand, the security of medical data is difficult to protect because the centralized data storage is vulnerable to attack and tampering. Therefore, we propose a model called Med-PPPHIS, which consists of a permission-less blockchain and a permissioned blockchain, named Med-DLattice, to serve the management of user's personal health information and form a chained protection mechanism for medical data. Med-DLattice features Directed Acyclic Graph (DAG) structure, where each account updates its Account-DAG asynchronously to other unrelated accounts. The Med-DLattice nodes can reach an efficient consensus with proposed DPoS-Quorum algorithm. Based on this model, by converting the medical data into on-chain tokens, a safe and efficient channel for data circulation is established, while the privacy of data is secured. We implement a prototype of Med-PPPHIS and introduce a blockchain-based closed-loop method for chronic disease management, which initially applies the model to national physique monitoring in Anhui Province, China. The performance of the model is evaluated by simulating 500 nodes on 25 AliCloud ECS virtual machines. Experimental result shows that Med-PPPHIS has low latency and high throughput, and the security analysis shows that the model is able to prevent Sybil attacks, DDoS attacks, etc.

Keywords Blockchain · Personal healthcare information system · Medical data tokenization · Chronic disease management · Electronic medical records

Introduction

In modern society, data carriers have evolved from traditional paper to electronic devices, especially in the medical field. The dissemination of electronic medical data among professional personnel is considered to be an important breakthrough in discovering new technologies and methods for treating diseases [1]. Therefore, the way to ensure that medical data (especially Electronic Medical Records that store patients' medical data (EMRs) and Electronic Health Records (EHRs)) is trustworthy and can be transmitted safely, efficiently and privately has become the focus of researches in both industry and academia. However, for some traditional medical information systems such as the traditional Electronic Medical Records (EMRs) or the Hospital Information System (HIS), a large amount of medical data is stored in the centralized database of each medical institution, making it difficult to transmit data between institutions and platforms, which leads to inconvenience for patients to seek medical care. For patients, privacy leakage is likely to occur in the storage and circulation of sensitive medical data. For hospitals, doctors are unable to analyze patients' conditions comprehensively and accurately because of the lack of the complete diagnosis and treatment history of patients. Meanwhile, an effective and safe sharing method is not available for the hospital to share the collected patient data. Centralized storage of medical data also requires

This article is part of the Topical Collection on *Transactional Processing Systems*

✉ He Zhao
zhaoh@hfcas.ac.cn

[1] Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China

[2] University of Science and Technology of China, Hefei 230026, China

high management costs, as well as unexpected risks such as data loss and leakage. For research institutions, the inability to effectively share medical data also slows the progress in medical research and the transformation of new theoretical achievements and medical technology. For medical insurance and other institutions, it is troublesome to carry out off-site settlement, the reimbursement formalities are complicated, and there is a probability that patient's medical data is falsified. In a word, two main problems are existing in the current situation of medical data management. First, it is difficult to share medical data:

(1) Medical data is stored in different database, even for the same patient;
(2) Effective and safe sharing methods and standards are lacking;

Second, it is difficult to secure medical data:

(1) Centralized medical data storage would present a single point failure;
(2) The privacy of sensitive medical data is not securely protected;
(3) Medical data is vulnerable to attack, tampering, and leakage;
(4) Labor costs and management costs are high.

The new blockchain technology has shown promising natures to solve these issues and advanced the biomedical and health care domains in various novel ways [2]. Blockchain is a cryptographically secure transactional singleton machine with shared-state [3], which contains an ordered list of records linked together through chains, trees or DAGs, etc., providing a trusted channel for the exchange of information and value. An example of simplified blockchain structure is shown in Fig. 1. Relying on the hash value of the previous block, all blocks are interlocked. If any block is tampered, it will trigger the whole changes of all hash pointers of subsequent blocks [4]. It is indeed because of its features such as decentralization, tamper-resistance and network data-sharing that blockchain has tremendous potential in the fields of data protection and tokenization [5–11].

Unlike cryptocurrencies which are created on and derived their values directly from the blockchains, digital assets are often issued by real world entities and blockchains are merely a medium to record their existence and exchanges [12]. Multichain [13] offers ledgers for storing and tracking asset history. IOTA [14] issues its token and offers its public ledger as a platform for micro-payment, which makes data been exchange among IoT devices. Previously, we proposed a method of data assetization and may help promote the data value transferring and data sharing among the Internet of Things based on Ethereum Smart Contracts [6].

This paper has combined our research foundation [15, 16] of the health informatics in the early stage, especially the management experience [17, 18] of chronic diseases. Assisting from the self-developed medical IoT devices, the blockchain technology is introduced in the process of national physique monitoring in Anhui Province, China. Medical data is collected to identify current or potential chronic diseases in the user, while the transfer and storage of data is encrypted. According to the collected data, the Chronic Disease Management Target (CDM-T) is determined, scientific, personalized exercise prescriptions (E-Prescription) will be prescribed, providing users with a safe, effective and private scientific health guidance for the management of chronic diseases.

The main contributions of this paper are as follows:

(i) We propose a model called Med-PPPHIS, which consists of a permissioned blockchain and a permission-less blockchain. The permissioned blockchain nodes store raw data and protect the storage information and fingerprint of the data (called data attribute information in the following description) on the chain, and periodically anchor the data snapshots to the public blockchain, so as to serve the user's health information management, and establish a chained protection mechanism for the medical data.
(ii) In the Med-PPPHIS, we propose a permissioned blockchain named Med-DLattice, in which each account updates its Account-DAG asynchronously to other unrelated accounts. The Med-DLattice nodes are able to

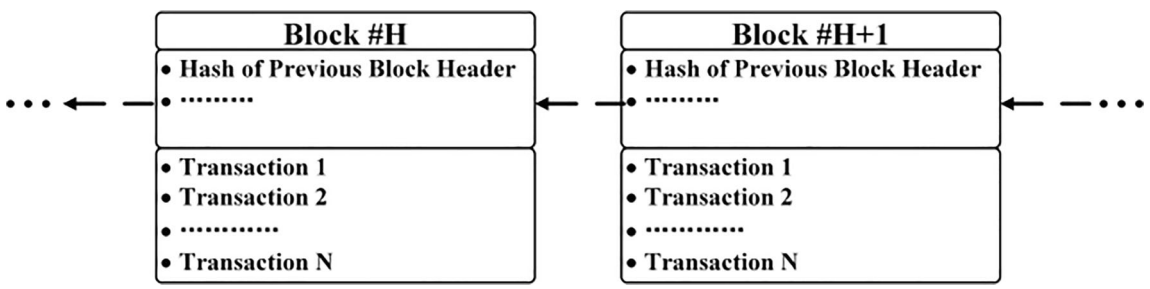

Fig. 1 An example of simplified blockchain. Each transaction is enclosed in a block. A block may contain multiple transactions and a hash value of the previous block's header, and thus forms a hash-chain

reach a consensus efficiently with proposed DPoS-Quorum algorithm.

(iii) We introduce a method of medical data tokenization based on Med-PPPHIS model. By converting the medical data into on-chain tokens, a channel for safe and efficient data circulation is established, while the privacy of the data is protected.

(iv) We implement a prototype of Med-PPPHIS and tentatively apply it to national physique monitoring in Anhui Province, China. A closed-loop method based on blockchain technology for chronic diseases management is proposed, providing users with a safe, effective and personalized scientific health guidance.

Related works

In this section, research trends about blockchain-based medical data management in industry and academia are outlined. EMR implementations are widespread and have been well recognized as costly investments [19]. Traditional electronic medical information systems (MISs) based on C/S or B/S and the later distributed MISs based on cloud storage [20, 21], have general problems such as centralized data storage, easy privacy leakage and data tampering. Esposito et al. [22] detailed the drawbacks of using cloud storage technology to establish a data sharing system in the medical field. They also raised the possible challenges of using blockchain technology in medical data sharing.

Since Bitcoin [23] was proposed by Satoshi Nakamoto in 2009, the blockchain technology has been increasingly valued. In industry, Philips, a medical device manufacturing industry, cooperates with Tierion, which is a start-up blockchain company, hoping to change the current medical condition of patients by the usage of blockchain technology [24]. Healthbank in Switzerland uses blockchain technology to handle health system business, ensuring the security of health data storage [25]. Change Healthcare in the United States claims that its blockchain smart healthcare network is now offering products with transparent claims management [26].

Alibaba Health is cooperating with Changzhou City to apply blockchain technology to the underlying architecture of Changzhou Medical Association, and has achieved the data interconnection of several local medical institutions [27]. Another Chinese company, Tencent, is connecting hospitals, pharmaceutical companies and users together based on its core technology, such as Tencent's payment, face recognition and blockchain technology. Safe circulations of electronic prescriptions are achieved and the entire process is traceable, which helps separation of dispensing and prescribing [28].

And in academia, Zyskind et al. used the blockchain technology for access control management and secure data storage [29]. Asaph Azaria et al. presented a blockchain-based data sharing system which was used as decentralized record management system to handle EMRs [10]. However, it is illegal to gather patient data and share them as rewards.

Fan et al. proposed a blockchain-based information management system to handle patients' information with improved consensus mechanism without large energy consumption and network congestion [9]. Zhou et al. proposed a blockchain-based medical insurance storage system based on extended Shamir's (t,n)-secret sharing [8]. Xia et al. proposed a blockchain-based data sharing framework that sufficiently addresses the access control challenges associated with sensitive data stored in cloud repositories among big data entities [7]. But the hospitals in the framework is reluctant to share the data to the third part due to the lack of incentives.

Li et al. proposed a novel patient-centric framework and a suite of mechanisms for access control of data to PHRs stored in semi-trusted servers and leverage ABE techniques to encrypt each patient's PHR file to achieve fine-grained and scalable data access control [30]. Guo et al. [31] preserved patient privacy and guaranteed the validity of EHRs encapsulated in blockchain based on an attribute-based signature (ABS) scheme with multiple authorities.

Med-PPPHIS model

Traditional blockchain-based medical management systems provides us two different data protection schemes. One only adopts the permissioned blockchain, the other only adopts permission-less blockchain. The former way is featured with high consensus efficiency and transaction throughput while lowers the security for protecting data. And the latter has higher security while the data protection cost also becomes higher. And medical data is relatively sensitive, its owner does not want it to be stored in the open database even if it is encrypted. The public exposure of encrypted data not only increases the risk of data leakage, but also increases the possibility of data being attacked.

Consisting of a permissioned blockchain and a permission-less blockchain, the Med-PPPHIS model proposed in this paper serves the user's personal health information management, as shown in Fig. 2. The permissioned blockchain stores the raw medical data in their nodes and protects the data attribute information on the chain, and periodically anchors the data snapshots of the permissioned blockchain to the permission-less blockchain, thereby forming a chained medical data protection mechanism from local cache to the permissioned blockchain and then to the permission-less blockchain.

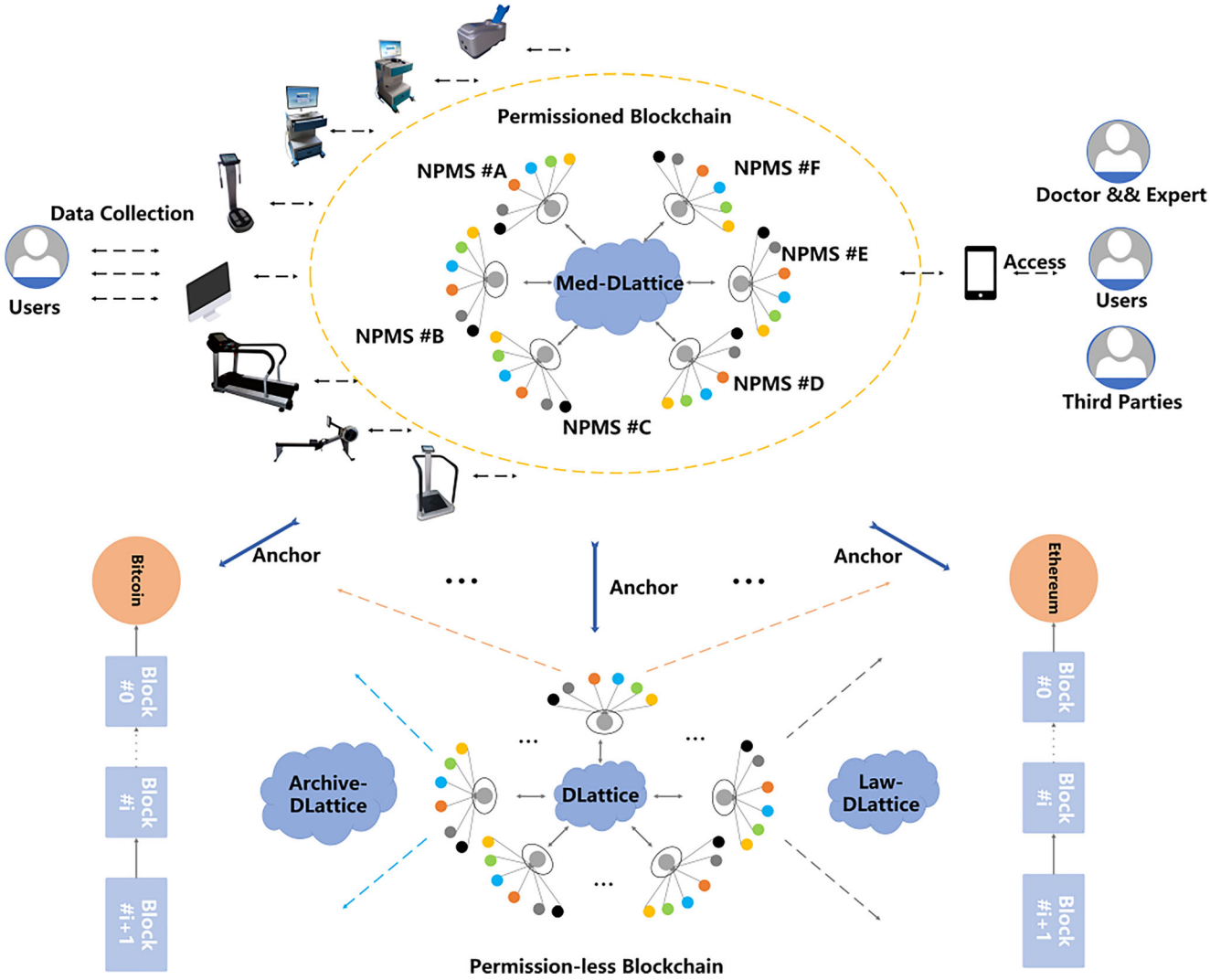

Fig. 2 Overall architecture of Med-PPPHIS model. Med-PPPHIS model is in the form of the "permissioned blockchain + permission-less blockchain". Med-DLattice is used as the permissioned blockchain and public blockchains, such as Bitcoin, Ethereum, DLattice, can be chosen as the public blockchain. This model can be applied to digital archives management, legal cases management and other specific scenarios

Assumptions

- Assumption 1. Med-PPPHIS assumes that honest users run bug-free software and the fraction of tokens held by honest users is above some threshold h (a constant greater than 2/3).
- Assumption 2. Med-PPPHIS assumes that at least $2 \times f + 1$ honest consensus-participating nodes are in the system, where f represents the number of byzantine adversaries.
- Assumption 3. Med-PPPHIS makes a "strong synchrony" assumption that most honest users can send messages that will be received by most other honest users within a known time bound δ_{term}. And this assumption does not allow network partitions.
- Assumption 4. Med-PPPHIS assumes that if some probability p is negligible, it means it happens with probability at most $O(1/2^{\lambda})$ for some security parameter λ. Similarly, if some event happens with high probability, it happens with probability of at least $1 - O(1/2^{\lambda})$.

Notions

Through this paper, we use these notions as shown in Table 1.

Permissioned blockchain

There are mainly two types of the permissioned blockchains: the private permissioned blockchain and the public permissioned blockchain. The public permissioned blockchain is also called the consortium blockchain, which consists of organizations, institutions or individuals with common goals. The consortium members participate in the consensus process and the data reading and writing permissions

Table 1 Notions and detailed description

Notions	Description
λ	Security parameters;
g	The generation of a cyclic group $\mathbb{G}$;
e	The bilinear map, $e : \mathbb{G} \times \mathbb{G}$;
$\mathbb{F}_p$	The finite field with character p;
N	The number of nodes in Med-PPPHIS model;
Q_e	The expected size of consensus committee;
Q_{ID}	The number of the honest identities;
∂	The number of stored secret shards;
t	The threshold of required shares to recover the secret;
Th_{anco}	Threshold of transaction blocks to be anchored;
$SSSS(t, N)$	Splitting up an original data into N shares. If and only if at least any t of N shares are collected, it should be easy to recover the original data.

are determined according to the consortium rules. The latest joined members need to be reviewed by other nodes already in the consortium [32] (Unless stated otherwise, the permissioned blockchain in this paper is refer to the public permissioned blockchain). This paper proposes a medical permissioned blockchain called Med-DLattice, in which national physique monitoring stations, hospitals, scientific research institutions, etc. can participate as members. The blockchain nodes store the raw medical data, and the data attribute information is protected on the chain. Thus, a secure and reliable underlying architecture is provided for data sharing and privacy protection while the medical data is tokenized.

Node and account (wallet)

In the model, each National Physique Monitoring Station (NPMS) is a consensus node in Med-DLattice, which is denoted as NPMS. Each NPMS communicates with each other though the Gossip Protocol. The ledgers in each NPMS, which are used to record the token assets and the medical data assets held by each user, are same. As a new NPMS is created, so is the corresponding consensus account *ConAC* at the same time. The account consists of a public-private key pair <*P*, *S*>. The public key *P* is called the account address, which is used to identify the identity of NPMS, and it is exposed to the whole network. Each NPMS has only one unique consensus account. It is worth noting that in Med-DLattice, the consensus node that originally had all the tokens is called Genesis Node, which is responsible for the allocation of tokens in the initial state of the permissioned blockchain, and the tokens are defined as *MDT*.

The *Account* is the main body of the actual user's participation in the model, and also consists of a public-private key pair <*P*, *S*>, where the public key *P* is used to identify the identity of the account and is exposed to the entire network; and the private key *S* is similar to the password in the ordinary system. The user holding the private key has the actual control of the account. The private key *S* can be used by the account to sign the transaction block or message to clarify the source of them. The *Account* includes normal accounts *NorAC* and consensus accounts *ConAC*. The normal is used to send and receive tokens and medical data assets, and to assign access and control permissions of the medical data. The *NorAC* in Med-DLattice may be ordinary users, doctors, scientific research institutions, even medical IoT devices. The *ConAC* has the same function as the *NorAC* except for the functions described above, and it is generally controlled by the administrator of NPMS.

A digital wallet may contain multiple accounts and the assets to which the accounts belong (including tokens and data assets). When the digital wallet is opened by the private key, it is possible to check the token balance and the data assets in the wallet, and to transfer token or share the data.

Transaction and block

A block contains only one transaction in this model, so it is called a Transaction Block, which is recorded as *TB*. The transaction blocks include Creating Transaction Blocks TB_{create}, Sending Transaction Blocks TB_{send}, Receiving Transaction Blocks $<TB_{receive} \mid TB_{deal}>$, and Authority Transaction Blocks TB_{auth}, etc., as shown in Fig. 3.

The TB_{create} is used to create accounts, including normal accounts and consensus accounts. The initial tokens of an account come from system allocation or other accounts. When a user (a medical IoT device, a doctor, a patient, a research institution, or a new NPMS) wants to join Med-DLattice, first he/she needs to purchase tokens *MDT* from other accounts, then a TB_{send} will be constructed by other accounts and sent to new users. Once the NPMSs reach the consensus, the new user is able to construct the corresponding $TB_{receive}$ to receive *MDT*, so that the new user will join to Med-DLattice. Users who have joined Med-DLattice may construct a TB_{send} and $TB_{receive}$ to send and receive tokens, or a TB_{send} and TB_{deal} to trade medical data assets. The transfer of both tokens and data require the confirmation of two transaction blocks. The users can share their own medical data with other users by the TB_{auth}. And the $TB_{delegate}$ is used to assign a station to wield its voting powers. It is worth noting that $TB_{delegate}$ only means that the station acts as the agent of the account to wield its voting powers, and the actual token assets in the account are not transferred. But if a station transfers its voting powers to another stations, it means that the station is actively offline.

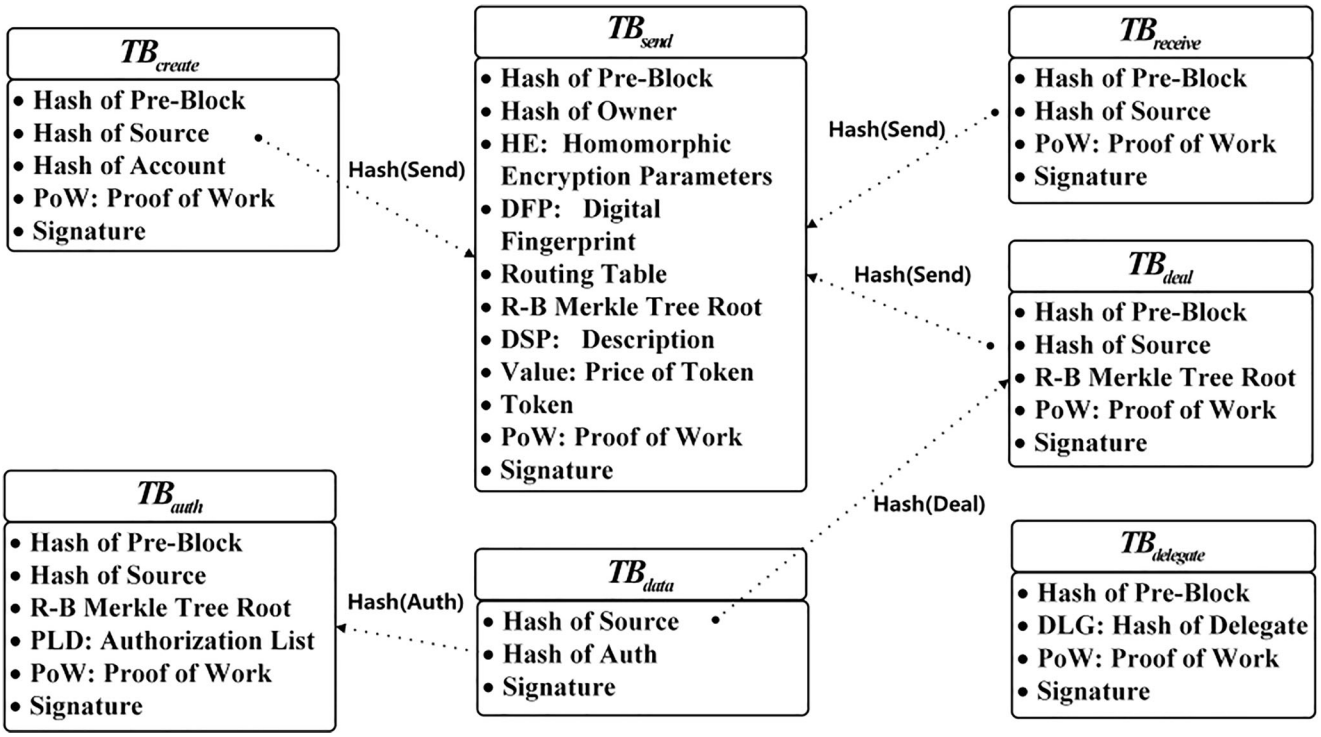

Fig. 3 Anatomy of transaction blocks

Med-DLattice

As shown in Fig. 4, Med-DLattice features DAG structure called Node-DAG. And each account has its own DAG structure, called Account-DAG. Account-DAG of all accounts are organized in the form of Merkle Patricia Tree (MPT) [33] in the Genesis Header. A Genesis Header and the Account-DAG of all accounts make up the Node-DAG. The public key of the NPMS is used as the Key, and the hash value of TB_{create} which is created as a Root Account Block (RAB) is used as the Value to jointly build the MPT.

The Account-DAG structure of each account is derived sequentially from its RAB, and is composed of the Token-Chain (T-Chain) and the Data-Tree (D-Tree). The income and expenditure of the tokens and the data assets of the account are recorded by T-Chain in the form of a unidirectional chain. D-Tree is a Red-Black Merkle Tree [34] combining with T-Chain, which stores the digital fingerprint of the data asset and corresponding access control permissions. D-Tree acts as an index to speed up the data query and data update. The digital fingerprint of the data is taken as the Key, while TB_{data} is used as the node to jointly build the Red-Black Tree. The Merkle Root of the Red-Black Tree is recorded in $H_{RBMerkle}$ of TB_{deal}.

DPoS-quorum consensus

In Med-DLattice, when a new station is created, forks are observed or even storing mistrust occurs, the DPoS-Quorum consensus algorithm proposed in this section is helpful for NPMSs to reach an effective consensus. In the consensus process, NPMSs could use Verifiable Random Functions (VRFs) [35] to verify whether they had valid consensus identities to participate in the consensus committee to solve the proposal according to the sum of the voting powers they hold and represent. If their identities are valid, the consensus vote will be conducted. When the number of votes collected by NPMSs exceeds the legal threshold, the consensus will be reached and the consensus process ends. Inspired by [36], the specific consensus process is divided into a setup phase and a consensus phase.

Setup phase

If a new NPMS (Public Key P_i) successfully joins to Med-DLattice (The current number of NPMSs is N, and their public-private keys are denoted as $<P_1, S_1>, \ldots, <P_n, S_n>$ respectively), it needs to calculate the Membership Public Key first, which is denoted as P_{mk}:

$$P_{mk} = a_1 \times P_1 + \ldots + a_i \times P_i + \ldots + a_n \times P_n,$$

where $a_i = h(P_i \| P_1 \| \ldots \| P_n)$; Then, Membership Signature, denoted as MS_i, is calculated:

$$MS_i = (a_i \cdot S_i) \times H(P_{mk}, i);$$

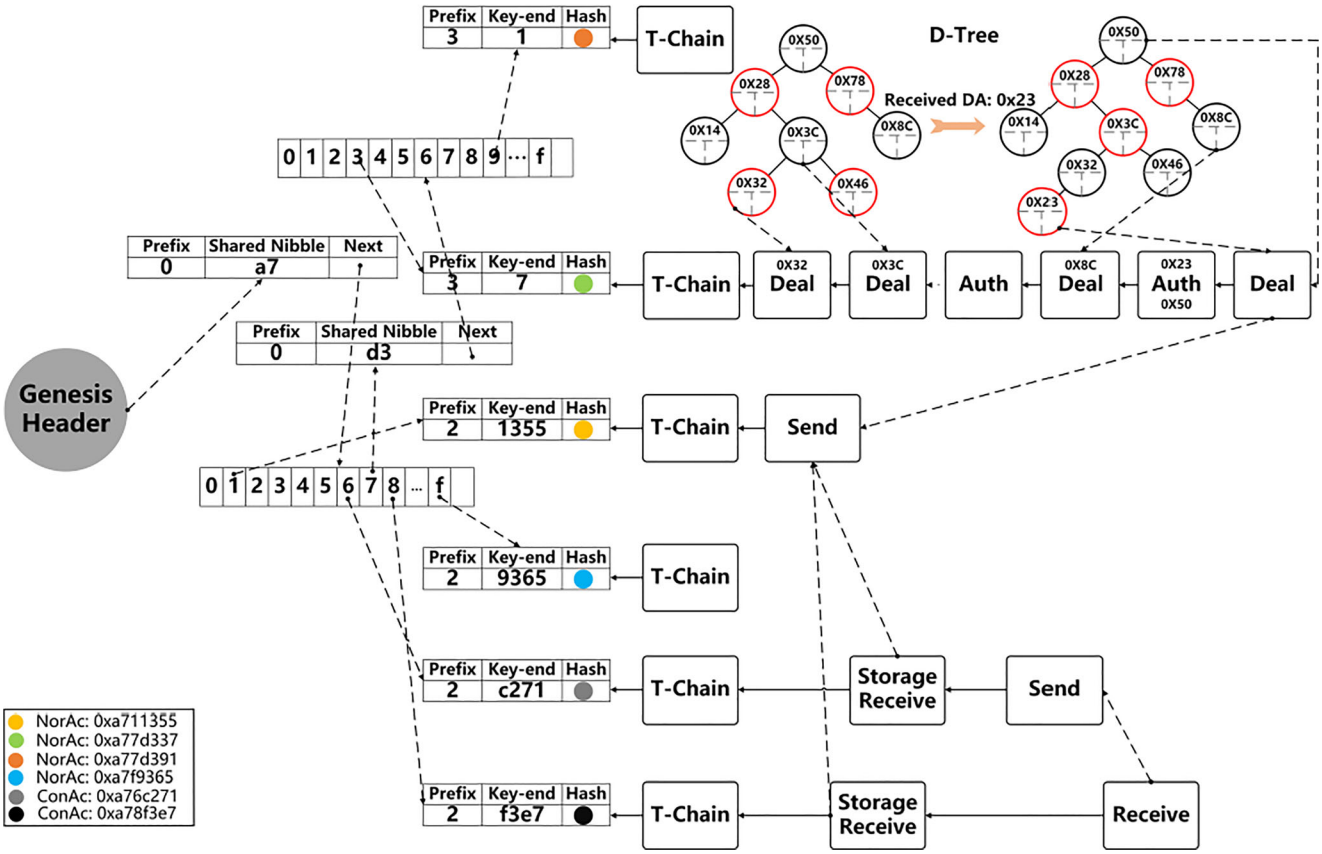

Fig. 4 Overview of Med-DLattice. Node-DAG consists of a Genesis Header and the Account-DAGs of all accounts. And the Account-DAG structure is composed of the T-Chain and the D-Tree. All accounts are organized in the form of MPT in the Genesis Header. An older Red-Black Merkle Tree of NorAc:0xa77d337 is depicted. After receiving a new DA (hash 0×23), the newer Red-Black Merkle Tree is shown in the Figure

Finally, it is exposed in Med-DLattice along with its public key. It is worth noting that the input x in $H(x)$ is hashed onto a curve. When a NPMS receives the public information $<P_i, MS_i>$ of a new NPMS, each NPMS needs to update their P_{mk} and the Membership Secret Key locally which is denoted as MK. Take the new NPMS as an example, what it needs to calculate is:

$$MK_i = (a_1 \cdot S_1) \times H(P_{mk}, i) + \ldots + (a_i \cdot S_i) \times H(P_{mk}, i) + \ldots + (a_n \cdot S_n) \times H(P_{mk}, i).$$

Consensus phase

- **Proposal Phase**

The following proposals may be created during the proposal phase:

– During the creation of a new station in Med-DLattice, the proposal is that whether the creation is allowed, recorded as $Pops_{create}$;
– During data anchoring, the proposal is the information that each account needs to be anchored (see "Permission-less Blockchain"), which is denoted as $Pops_{anchor}$;
– When a NPMS receives a fork set, in which each transaction block has the same previous block hash (H_{PRE}), the proposal is to choose a certain TB from the set for consensus and record as $Pops_{fork}$;
– When an account accesses the stored medical data and finds that no valid data acquisition service is available on the data storage station, or the data storage station finds that the account of data producer is unwilling to pay the data storage fee, the proposal is referred as $Pops_{storage}$ at this time.
– **ID Generation**

When a NPMS receives the proposal, it calculates whether it has valid consensus identities to participate in the consensus committee that resolves the proposal according to the sum of the tokens it holds and represents based on VRFs. If the consensus identity is valid, the vote for consensus will begin.

Algorithm 1. ConsensusIDGeneration() generates validate consensus identities.

Input: w : the sum of the tokens the NPMS holds and represents; ctx : the context information of the NPMS; $seed$: the hash value of proposal; $Pops$: the type of proposal;
Output: Consensus Identity

$ID_{cons} \leftarrow empty$
$p \leftarrow w / ctx.W$
$power \leftarrow 0$
$< hash, proof > \leftarrow VRF_{sk}(Seed \,||\, Pops)$
$index \leftarrow hash / 2^{len(hash)}$
while $index \notin [\sum_{k=0}^{power} B(k, ctx.C_E, p), \sum_{k=0}^{power+1} B(k, ctx.C_E, p)]$ do
  $power++$
end while
$ID_{cons}.pk \leftarrow ctx.pk; ID_{cons}.power \leftarrow power$
$ID_{cons}.hash \leftarrow hash; ID_{cons}.proof \leftarrow proof$
return ID_{cons}
End

ConsensusIDGeneration() (Algorithm 1) is used to generate consensus identities for an account based on the sum of the tokens it holds and represents, and calculate the voting powers. If the calculated voting powers of the consensus identity is 0, the identity is invalid. All valid consensus identities together constitute a consensus committee for resolving the proposal.

VerifyID() (Algorithm 2) is used to verify whether a consensus identity ID_{cons} is a valid member of a consensus Committee.

Algorithm 2. VerifyID() verifies a consensus identity whether in the consensus committee.

Input: w : the sum of the tokens the NPMS holds and represents; ctx : the context information of the NPMS; $seed$: the hash value of proposal; $Pops$: the type of proposal; ID_{cons} : the consensus identity;
Output: True or False

if $\neg VerifyVRF_{pk}(ID_{cons}.hash, ID_{cons}.proof, Seed \,||\, Pops)$ **then**
  return false
$p \leftarrow w / ctx.W$
$power \leftarrow 0$
$index \leftarrow hash / 2^{len(hash)}$
while $index \notin [\sum_{k=0}^{power} B(k, ctx.C_E, p), \sum_{k=0}^{power+1} B(k, ctx.C_E, p)]$ do
  $power++$
end while
if $ID_{cons}.power == power$ **then**
  return *true*
else
  return *false*
End

- **Voting**

The NPMSs with valid consensus identities will vote for a certain proposal, which is denoted as $Pops$. The signature can be obtained:

$$Sig_i = S_i \times H(P_{mk}, Pops) + MK_i,$$

A consensus vote, $<ID_{cons}, h(Pops), Sig_i>$, which consists of the signature Sig_i, the consensus identity ID_{cons} and the hash of the corresponding proposal, will be broadcasted to other NPMSs in permissioned blockchain.

- **Counting**

The NPMSs will count the consensus votes. If the number of votes for a proposal exceeds the legal threshold and its signature set is verified, the NPMSs reach a consensus on the proposal. The value of legal threshold depends on the security parameters, as shown in Table 2.

Denote the signature set of consensus votes as (SS, SP), where $SS = Sig_j + \ldots + Sig_k$, and $SP = P_j + \ldots + P_k$.

$$If\ e(g, SS) = e(SP, H(P_{mk}, Pops)) \cdot e(P_{mk}, H(P_{mk}, j) + \ldots + H(P_{mk}, k))$$

Then the NPMSs in Med-DLattice reach a consensus on $Pops$.

$$Proof: e(g, SS) = e\left(g, Sig_j + \cdots + Sig_k\right)$$
$$= e\left(g, S_j \times H(P_{mk}, Pops) + .. + S_k \times H(P_{mk}, Pops) + MK_j + .. + MK_k\right)$$
$$= e\left(g, S_j \times H(P_{mk}, Pops) + .. + S_k \times H(P_{mk}, Pops)\right) \cdot e\left(g, MK_j + .. + MK_k\right)$$
$$= e\left(S_j \times g + .. + S_k \times g, H(P_{mk}, Pops)\right) \cdot e\left(g, MK_j + .. + MK_k\right)$$
$$Meanwhile: e(g, MK_i) = e(P_{mk}, H(P_{mk}, i))$$
$$= e(g, (a_1 \cdot S_1) \times H(MK, i) + .. + (a_n \cdot S_n) \times H(MK, i))$$
$$= e(\{a_1 \cdot S_1) + ... + (a_n \cdot S_n)\} \times g, H(MK, i))$$
$$= e(a_1 \times P_1 + \cdots + a_n \times P_n, H(MK, i))$$
$$= e(MK, H(MK, i)) Thus: e(g, SS)$$
$$= e(SP, H(P_{mk}, Pops)) \cdot e(P_{mk}, H(P_{mk}, j) + .. + H(P_{mk}, k)).$$

Permission-less blockchain

The permission-less blockchain is also known as the public blockchain, in which any organizations or individuals can participate and have the permission to read and write blockchain data [32]. By periodically anchoring data snapshots of the permissioned blockchain to the public blockchain, a chained medical data protection mechanism from the local database cache to the permissioned blockchain and then to the public blockchain is formed. Med-DLattice proposed in the previous section is used as the permissioned blockchain, and DLattice [37], a public blockchain specially designed for data tokenization, which is proposed by our research team earlier, is used as the public blockchain. Other public blockchains, such as Bitcoin, Ethereum, also can be chosen as the public blockchain.

When the block height of all accounts in Med-DLattice reaches Th_{anco} or multiples of Th_{anco}, the NPMSs take turn to propose $Pops_{anchor}$, and anchor the snapshots of Med-

Table 2 The maximum and minimum number of total identities, honest identities, and malicious identities that the system may generate with different security parameters

Security Parameter	Expected Identity	Adversary		Honest		All	
		Min	Max	Min	Max	Min	Max
$\lambda = 20$	100	3	45	41	126	36	151
	200	14	74	104	224	136	271
	300	27	100	170	317	221	386
	400	41	126	239	499	308	499
	500	56	151	308	499	397	610
$\lambda = 15$	100	5	40	47	118	63	143
	200	17	68	112	213	146	259
	300	32	93	180	305	233	372
	400	47	118	251	394	322	483
	500	63	143	322	499	413	592
$\lambda = 10$	100	8	35	54	109	71	132
	200	22	61	122	201	158	245
	300	38	85	193	289	248	355
	400	54	109	266	377	340	463
	500	71	132	340	463	432	571

DLattice to the OP_RETURN [38] in Bitcoin, or the smart contract on Ethereum. The snapshot of each account is denoted as $Anco = <P, MDT, BH, RB-MT>$, where P is the public key of the account, MDT is the token held by the account, BH is the height of T-Chain, and $RB-MT$ is the Red-Black Merkle Root of the D-Tree. In DLattice, each NPMS in Med-DLattice corresponds to a normal account in DLattice, and TB_{send} will be constructed just like in Med-DLattice to anchor the snapshots data to the DLattice.

Methodology

Medical data collection and assembling

Data assembling is to assemble the raw data D_{raw} into a data structure that can be used by the model at generation source of data. This data structure is denoted as $D = (P_p, P_o, E_{p-o}(D_{raw}), T, Sig_{s-p}(D_{raw}))$, where P_p represents the public key of a data producer (e.g. medical IoT devices, patients, doctors, experts, etc.); P_o represents the public key of a medical data owner (e.g. patients, doctors); D_{raw} represents raw medical data, including continuous data (such as heart rate data collected by wearable devices) or non-continuous data (such as medical data collected by our self-developed medical IoT devices, as shown in Table 3), and records data (such as cases and prescriptions generated between doctors and patients); $E_{p-o}(D_{raw})$ indicates that the data producer uses the public key P_o of data owner to asymmetrically encrypt the raw data D_{raw}, and only the data owner who owns the corresponding private key S_o can decrypt the data to obtain D_{raw}; $Sig_{s-p}(D_{raw})$ indicates that the data producer uses its private key Sk_p to sign raw data, the signature can be verified only by its public key P_p; T represents the timestamp of the data generation.

Medical data decentralized storage

A method for distributed data storage is proposed in this section. The encrypted original medical data is distributed in various NPMSs of Med-DLattice, and the data attribute information is recorded on the chain. The decentralized storage is thus achieved and single point failure is avoided. This is also the basis of medical data sharing and is an important part of the process of data tokenization. Comparing with traditional blockchains, such as Bitcoin and Ethereum, each NPMS only store part of the data, and the data shards need not be stored by each NPMS. It not only reduces the size of the blockchain, but also protects the privacy of the data, as shown in Fig. 5. Given below demonstrates an example that the collected health signs data of a user *Oer* (data owner) by health signs testing devices Dev_h (data producers) is distributed in various NPMSs of Med-DLattice:

Upload (from NPMS to blockchain)

- Step 1: Data Assembling. First, in the process of collecting medical data, the Dev_h assemble the raw data into the data structure specified in the model (see the "Medical Data Collection & Assembling"). The assembled data structure D_h will be asymmetrically encrypted by the public key P_{oer} of *Oer*. The encrypted data is denoted as ct_h;
- Step 2: Secret Splitting. Shamir's $SSSS(t, N)$ scheme [39] is used to split up ct_h into N shares, where $t = f + 1$. Dev_h

randomly samples a polynomial $F(x)$ of degree t over $\mathbb{F}_p$ as the following polynomial:

$$F(x) = a_0 + a_1 x + ... + a_{t-1} x^{t-1},$$

where $a_0, a_1, ..., a_{t-1} \in \mathbb{F}_p$. We denote that $a_0 = ct_h$. Let

$$f(x) = a_1 x + ... + a_{t-1} x^{t-1}.$$

Then, we have $F(x) = ct_h + f(x)$. And Dev_h generates a verification parameter set HE_g as follow:

Table 3 Medical IoT devices involved in Med-PPPHIS and key indexes collected from these devices. (*) represents a self-developed medical IoT device

Category	Device	Key Index
Health Signs Monitoring Equipment	Testing Instrument (*)	Heart Rate (HR), Subendocardial Viability Ratio (SEVR), Systolic Blood Pressure (SBP), Diastolic Blood Pressure (DBP), Augmentation Index (AI), etc.
	Pulmonary Function Testing Instrument (*)	Forced Vital Capacity (FVC), Forced Expiratory Volume in 1 Second (FEV1), etc.
	Bone Density Meter (*)	Bone Stiffness Index (STI), T-Value, Z-Value, etc.
	Body Composition Meter (*)	Body Mass Index (BMI), Body Fat Rate (BF), etc.
	National Physical Fitness Testing Instrument	Grip Strength, Vertical Jump, Push-up, Sit-up, Sit and Reach, Single Leg Stance *with Eyes Closed,* Step Test, etc.
Athletic and Functional Performance Assessment Equipment	Cardiopulmonary Endurance Testing Instrument (*)	Max Heart Rate (HR), Functional Capacity (F.C.)
	Balance Function Testing Instrument (*)	Center of Pressure (COP), Center of Pressure Velocity (COPV), etc.
Intelligent Fitness Equipment	Motion-Sensing Game Device	Exercise Event, Intensity, Duration, etc.
	Electric Smart Treadmill (*)	Exercise Intensity, Distance, Duration, etc.
	Indoor Rowing Machine (*)	Effect of Rowing, Stroke Frequency, Distance, etc.
Wearable Device	Heart Rate Monitor Watch (*)	Heart Rate (Max, Average, Min), etc.

$$HE_g = \{g, g^{a_0}, g^{a_1}, g^{a_2}, ..., g^{a_{t-1}}\},$$

where g is a base point of 254-bit Barreto-Naehrig curve (BN-curve). After that, Dev_h will take N secret numbers $scrt_1$, .., $scrt_i$, .., $scrt_n$ to generate N secret shards:

$$< scrt_i, F(scrt_1) >, ., < scrt_i, F(scrt_i) >, .,$$
$$< scrt_n, F(scrt_n) >,$$

Then the commitments of N shards will be calculated separately:

$$CMT_{scrt_i} = g^{F(scrt_i)};$$

- Step 3: Secret Shards Storage. Once the ct_h is divided into N shards according to the previous step, each secret shard is hashed separately:

$$H^i_{scrt} = h(F(scrt_i)),$$

where the output in $h(x)$ is a number. Then we have H^1_{scrt},.., H^i_{scrt},..,H^n_{scrt}. Each hash value exclusives or (XOR) the public key of the station the account in, then select a NPMS with a closer XOR distance to store the secret shards $F(scrt_i)$. The public key of the selected NPMS will be used to asymmetrically encrypt the shards (assuming that the public keys of the NPMSs storing N shards are P_1,..,P_i,..,P_n respectively):

$$EF_i = E_{ecies}\left(F(scrt_i) \| P_i\right);$$

Any ∂ of N shards are stored in NPMSs, where $\partial \in [2f+1, N]$, and each stored shard is denoted as $<scrt_i, EF_i>$. The value of ∂ is depending on the will of the account itself. The larger the ∂ is, the more copies are stored and the higher the security is, but the higher the storage cost is. Once the storage is done, the shard storage station will use its private key $S_{station}$ to sign the stored shards:

$$Sig_{scrt_i} = Sig_{BLS}\left(< scrt_i, EF_i > \| S_{station}\right).$$

And its hash value will be calculated:

$$H^i_{sign} = h\left(Sig_{scrt_i}\right),$$

Then hash value H^i_{sign} will be returned to Dev_h.

- Step 4: Secret Shards Anchoring. TB_{send} will be constructed by Dev_h, where HE_g is recorded in HE; the shards storage list and the hash value of signature of the stored

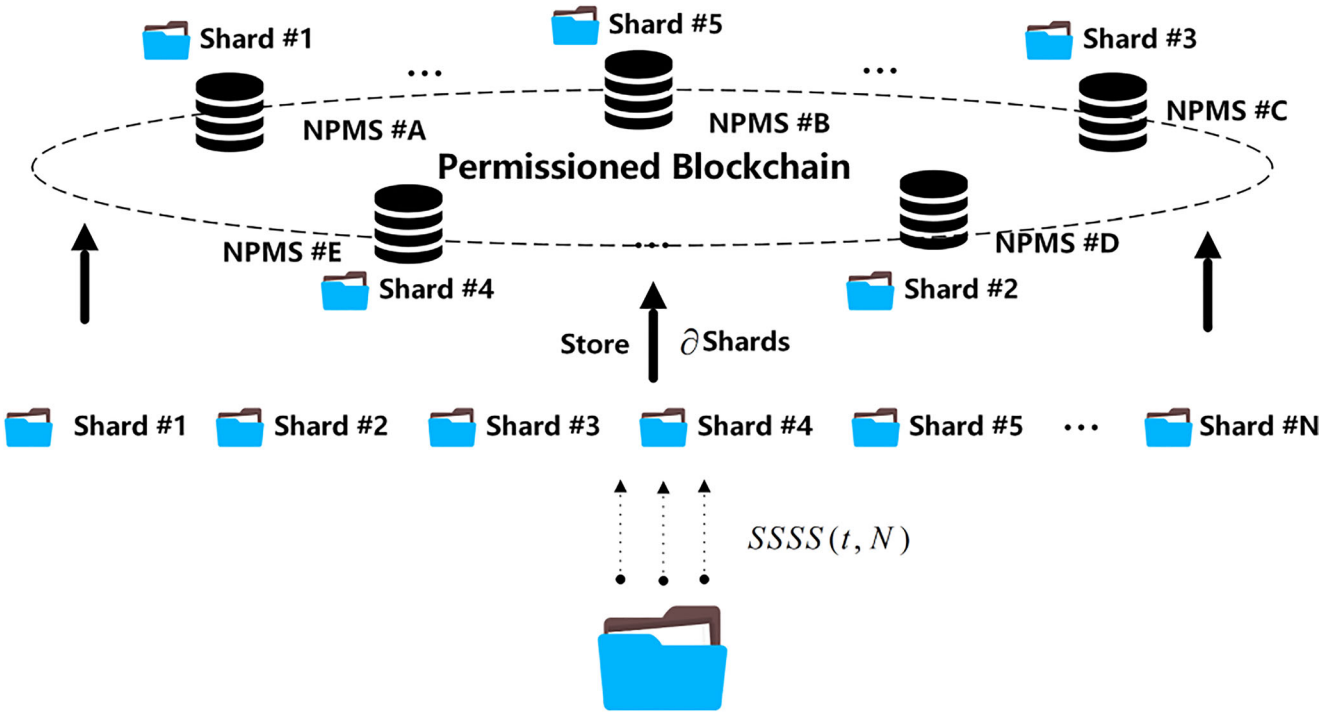

Fig. 5 Schematic diagram of distributed data storage. Shamir's $SSSS(t, N)$ scheme is used to split up an encrypted original medical data into N shares. And any ∂ of N shares are stored in NPMSs. Then, if and only if at least any t of N shares are collected, it should be easy to recover the encrypted original medical data

shards are recorded in $T_{routing} = \left\{P_i, scrt_i, CMT_{scrt_i}, H^i_{sign}\right\}$. Finally, TB_{send} will be signed and broadcasted.

Verify (blockchain)

- Step 1: Transaction Verification. The set of verification parameters stored in *HE* will be verified whether it's legal by each NPMS after they receive TB_{send}.

If

$$\begin{aligned} CMT_{scrt_i} &= (g^{a_0})(g^{a_1})^{scrt_i}(g^{a_2})^{scrt_i^2}\ldots(g^{a_{t-1}})^{scrt_i^{t-1}} \\ &= g^{a_0+a_1 scrt_i+a_2 scrt_i^2+\ldots+a_{t-1}scrt_i^{t-1}} \\ &= g^{F(scrt_i)} \end{aligned},$$

Then HE_g in TB_{send} is legal.

- Step 2: Storage Verification. When the data storage station receives TB_{send}, it locally decrypts the stored shards according to its public key $P_{receiver}$:

$$F(scrt_i) = DE_{ecies}\left(EF_i \| S_{receiver}\right),$$

Then HE_g in TB_{send} will be used to verify whether the stored shards are legal:

If

$$\begin{aligned} g^{F(scrt_i)} &= (g^{a_0})(g^{a_1})^{scrt_i}(g^{a_2})^{scrt_i^2}\ldots(g^{a_{t-1}})^{scrt_i^{t-1}} \\ &= g^{a_0+a_1 scrt_i+a_2 scrt_i^2+\ldots+a_{t-1}scrt_i^{t-1}} \end{aligned},$$

Then the stored shards are legal.

Retrieve (from blockchain to NPMS)

- Step 1: If *Oer* wants to obtain the health data collected by Dev_h, first, it shall construct TB_{deal} base on TB_{send} to receive the data, then it can separately obtain the secret shards from the NPMSs according to the $T_{routing}$ in TB_{send};
- Step 2: The public key P_{oer} of *Oer* will be used by the station for asymmetric encryption of the stored shards EF_i:

$$EF(scrt_i) = E_{ecies}\left(F(scrt_i) \| P_{oer}\right),$$

Then it will be signed by the station's private key S_{sender}:

$$Sig_{F(scrt_i)} = Sig_{BLS}\left(EF(scrt_i) \| S_{sender}\right),$$

and construct a message:

$$< P_{sender}, EF(scrt_i), Sig_{F(scrt_i)} >,$$

to send to *Oer*.

- Step 3: *Oer* receives ∂ shards from NPMSs, and then use its private key S_{oer} to retrieve the shards $F(scrt_i)$:

$$F(scrt_i) = DE_{ecies}\left(EF(scrt_i)\|S_{oer}\right).$$

Oer can reconstruct a polynomial of degree t by Lagrange Interpolating [38]:

$$\begin{cases} a_0 + a_1 scrt_1 + ... + a_{t-1} {scrt_1}^{t-1} = F(scrt_1) \\ a_0 + a_1 scrt_2 + ... + a_{t-1} {scrt_2}^{t-1} = F(scrt_2) \\ \qquad\qquad ..... \\ a_0 + a_1 scrt_t + ... + a_{t-1} {scrt_t}^{t-1} = F(scrt_t) \end{cases}$$

Then $ct_h = F(0)$, and we decrypt it with the private key S_{oer}:

$$D_h = DE_{ecies}\left(ct_h\|S_{oer}\right),$$

D_h is the raw health data.

Medical data tokenization

After the medical data is distributed stored, the data producer and the data owner record the data attribute information on the chain by transaction, and the data is tokenized. Therefore, the data owner can easily manage and use their data assets and speed up the circulation of data. This is the basis for data transfer, sharing, and permission management. Take for example a doctor *Dor* makes an E-Prescription for the user *Oer*, the specific process of data tokenization is described in this section.

- Step 1: The E-Prescription is issued by *Dor* and is assembled into data struct D_{e-p}. And the D_{e-p} is divided into serval shares and distributed in NPMSs (see "Medical Data Decentralized Storage"). Then, TB_{send} will be constructed and broadcasted to the system;
- Step 2: Each NPMS will verify whether the verification parameters and signature in TB_{send} are legal after they receives TB_{send}; each shard storage station also verifies whether the stored secret shards are legal;
- Step 3: After verification, the TB_{send} is added to their T-Chain by each NPMS. If there is a fork, it will be added after the consensus is reached (see "Double Spending Attack");
- Step 4: Each shard storage station is able to construct $TB_{receive}$ to get the corresponding storage fee;
- Step 5: If the *Oer* is online and finds there's an unreceived E-Prescription, he/she should construct TB_{deal} to receive it and pay the fee, meanwhile the corresponding TB_{data} should be constructed on his/her D-tree, and the update of the red-black merkle tree is completed. At this time, the E-Prescription issued by *Dor* has been stored in the user's digital wallet as data assets, thus completing the tokenization of medical data.

Efficient and secure medical data sharing

In order to share the data, Proxy Re-encryption scheme is used to control the access of medical data after the data is converted into valuable data assets in the permissioned blockchain. If the user *Oer*wants to share their health data $D_h$with the doctor *Dor*, the detailed process should be as follows:

- Step 1: *Oer* constructs TB_{deal} to receive D_h collected by the health signs testing devices Dev_h. According to the method in "Medical Data Decentralized Storage", the ct_h which is asymmetrically encrypted by the public key P_{oer} of *Oer* can be recovered from the shards storage NPMSs.
- Step 2: If *Oer* wants to share the data with *Dor*, he/she should use the public key P_{dor} of *Dor* to calculate the re-encrypted key:
- $$Re{-}Key_{o-d} = RE{-}Key\left(Sk_{owner}\|Pk_{doctor}\right);$$
- Step 3: *Oer* constructs TB_{auth} to record $Re-Key_{o-d}$ in its *Pld* in form of $<P_{dor}, Re-Key_{o-d}>$, and broadcasts it after signing. Once TB_{auth} is confirmed by Med-Lattice, *Dor* first obtains $T_{routing}$ from TB_{send} according to corresponding TB_{auth}. Then he/she gets ct_h according to $T_{routing}$. At last, he/she decrypts ct_h by using its private key S_{dor} and the re-encrypted key recorded in *Pld* of TB_{auth}. We have:

$$D_h = DE_{ecies}\left(RE{-}Enc\left(ct_h\|Re{-}Key_{o-d}\right)\|S_{dor}\right);$$

At this point, the access control permission of D_h granted by *Oer* are obtained by*Dor*, thereby completing the sharing of data.

Blockchain-based closed-loop approach for chronic disease management

For now, there are 28 NPMSs as members participating in the medical permissioned blockchain in Anhui Province, China. Based on the proposed Med-PPPHIS model, a preliminary exploration and application is been carried out in the process of national physique monitoring, while a closed-loop approach based on blockchain for chronic disease management is proposed. The flowchart is shown in Fig. 6.

In the closed-loop method, the user's medical data will be collected by various medical IoT devices with user's authorization and be used to identify current or potential chronic diseases. According to the collected data, the Chronic Disease Management Target (CDM-T) is determined, scientific and personalized E-Prescription will be formulated. The data owners have become the real dominant of data by blockchain technology. Not only the privacy, validity, and security of data is protected, but the circulation speed of data between devices, users, and doctors is also accelerated. Meanwhile, the closed-loop structure is helpful to track users' health conditions accurately and sustainably.

This section specifically describes the closed-loop method based on blockchain for chronic disease management, where the users are denoted as *Oer*, the doctors/experts are denoted as *Dor*, the health questionnaire devices are denoted as Dev_q, the health signs monitoring devices are denoted as Dev_h, the athletic and functional performance assessment devices are denoted as Dev_s, the intelligence fitness devices are denoted as Dev_e, and the wearable devices are denoted as Dev_w.

- Step 1: No matter Dev_q, Dev_h, Dev_S, Dev_e, or *Oer* and *Dor*, these entities first create a digital wallet in Med-PPPHIS to generate the corresponding public and private key <*P*, *S*>;
- Step 2: *Oer* uses Dev_q to fill in the health questionnaire (including Physical Activity Readiness Questionnaire (PAR-Q), Cardiovascular Risk Assessment Questionnaire, Exercise Contraindications Screening Questionnaire, Questionnaire for common chronic diseases that are not covered by physical examination devices). After Dev_q uses the P_{oer} of *Oer* to asymmetrically encrypt these questionnaires D_{qus} and stores the ciphertext ct_q in a distributed manner, the TB_{send} will be constructed and broadcasted in Med-DLattice;
- Step 3: *Oer* first constructs TB_{deal} in his/her T-Chain to receive TB_{send}, and then constructs TB_{auth} to grant *Dor* to access D_{qus};
- Step 4: *Dor* obtains D_{qus} from *Oer* according to TB_{auth}, and evaluates whether the user is suitable for chronic disease management;
- Step 5: Dev_h and Dev_s collect the health data D_{health} and athletic ability data D_{sport} of *Oer* separately, and obtain ct_h and ct_s by using asymmetrical encryption. Then the encrypted data are also stored in a distributed manner, and TB_{send} will be constructed to anchor D_{health} and D_{sport} on Med-DLattice;
- Step 6: *Oer* constructs TB_{deal} to receive D_{health} and D_{sport} separately, and then grants the access permissions of D_{health} and D_{sport} to *Dor* by corresponding TB_{auth};
- Step 7: According to the multi-source information sorting rules proposed by our research team [40, 41], *Dor* determines the *Oer*'s CDM-T; the parameters of the E-Prescription are determined by D_{sport}. Then *Dor* asymmetrically encrypts the E-Prescription, the attribute information of which is anchored on the chain after the encrypted data is stored;
- Step 8: *Oer* does exercise according to the E-Prescription. If the exercises need Dev_e, the exercise data $D_{exercise}$ is collected by Dev_e, and the real-time heart rate data D_{hr} during the exercise is collected by Dev_w; if barehanded exercises (running, swimming, etc.) are included, the $D_{exercise}$ is obtained by Dev_w. The medical IoT devices asymmetrically encrypt $D_{exercise}$, and the encrypted data will also be stored and then anchored in the chain. After *Oer* receives $D_{exercise}$, he/she will grant the access permission to *Dor*;
- Step 9: *Dor* will evaluate the effect of chronic disease management according to the feedback from user. The screening for other chronic diseases will depend on whether the former CDM-T is achieved, if not, repeat Step 5–8, thereby forming a closed-loop structure.

We can see from each step of the closed-loop method (Fig. 7) that the transfer of medical data requires authorization from the data owner, and data is encrypted in the process of storage or circulation between multiple roles, so as to build a secure and reliable circulation channel to ensure the privacy of users. Once the medical data is tokenized on the chain, the flow of data between multiple roles is stimulated, thus accelerating the efficiency of data sharing.

Implementation and evaluation

We implement Med-PPPHIS model and the goals of our evaluation are twofold. We firstly make security and efficiency analysis about Med-PPPHIS. The second goal is to compare Med-PPPHIS to other related systems, including MeDShare, MedRec, etc.

Implementation

The implementation of all components of Med-PPPHIS model is shown in Figs. 8 and 9, including a permissioned blockchain, digital wallets and a blockchain explorer. We implement a prototype of permissioned blockchain, Med-DLattice, in Golang [42], and choose LevelDB [43] as the backend database to store the blockchain data and raw medical data. A gossip network is constructed by using go-libp2p library (go-libp2p-pubsub) [44]. The Med-DLattice periodically anchors the block snapshots of the permissioned blockchain to the public blockchain, DLattice, which was designed by our research team previously. At the same time, we use the Express framework of NodeJS [45] to develop digital wallets for users and doctors respectively, and use Redis [46] as a cache. The users' digital wallet, shown in Fig. 8a, provide

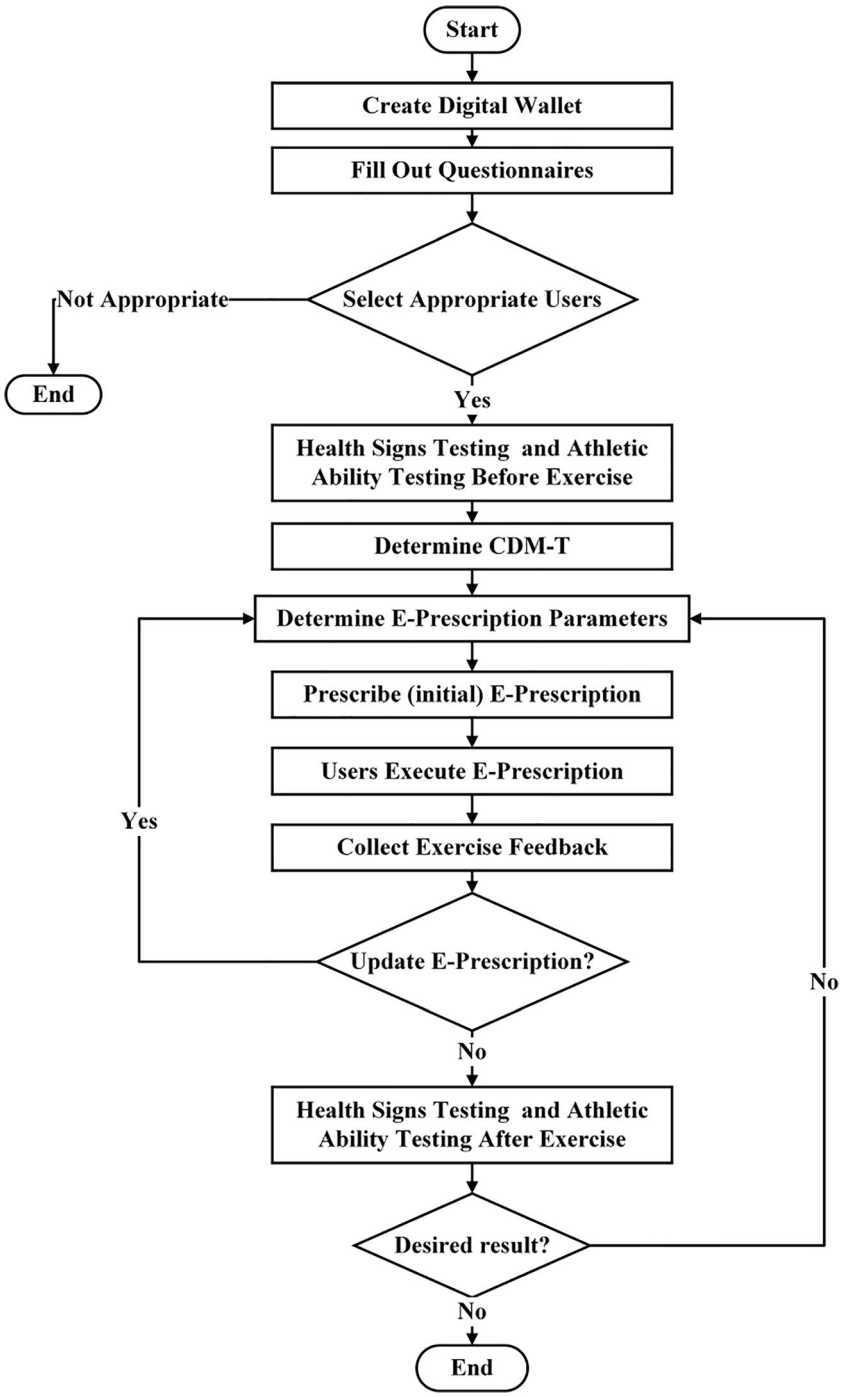

Fig. 6 Flowchart of the closed-loop approach based on blockchain for chronic disease management

users with the data asset query service as well as the transfer of token asset and data sharing. The interface of doctors' digital wallet is shown in Fig. 8b. Except for the same basic functions as users', it also can determine users' the Chronic Disease Management Target and make exercise prescriptions. We also develop a blockchain explorer in NodeJS to provide detailed information about Med-DLattice. The blockchain data stored in each Account-DAG in the Med-DLattice can be enquired through the blockchain explorer, as shown in Fig. 9.

Elliptic Curve Integrated Encryption Schema (ECIES) [47] is used for asymmetric encryption. We use SHA-256 for a hash function and use the VRF outlined in Goldberg [48]. The signature algorithm adopts Elliptic Curve Digital Signature Algorithm (ECDSA) [47] and Boneh-Lynn-Shacham (BLS) signature scheme [49]. For proxy re-encryption, we use the AFGH algorithm [50] to accomplish access control. Table 4 shows the parameters in our prototype of Med-PPPHIS model.

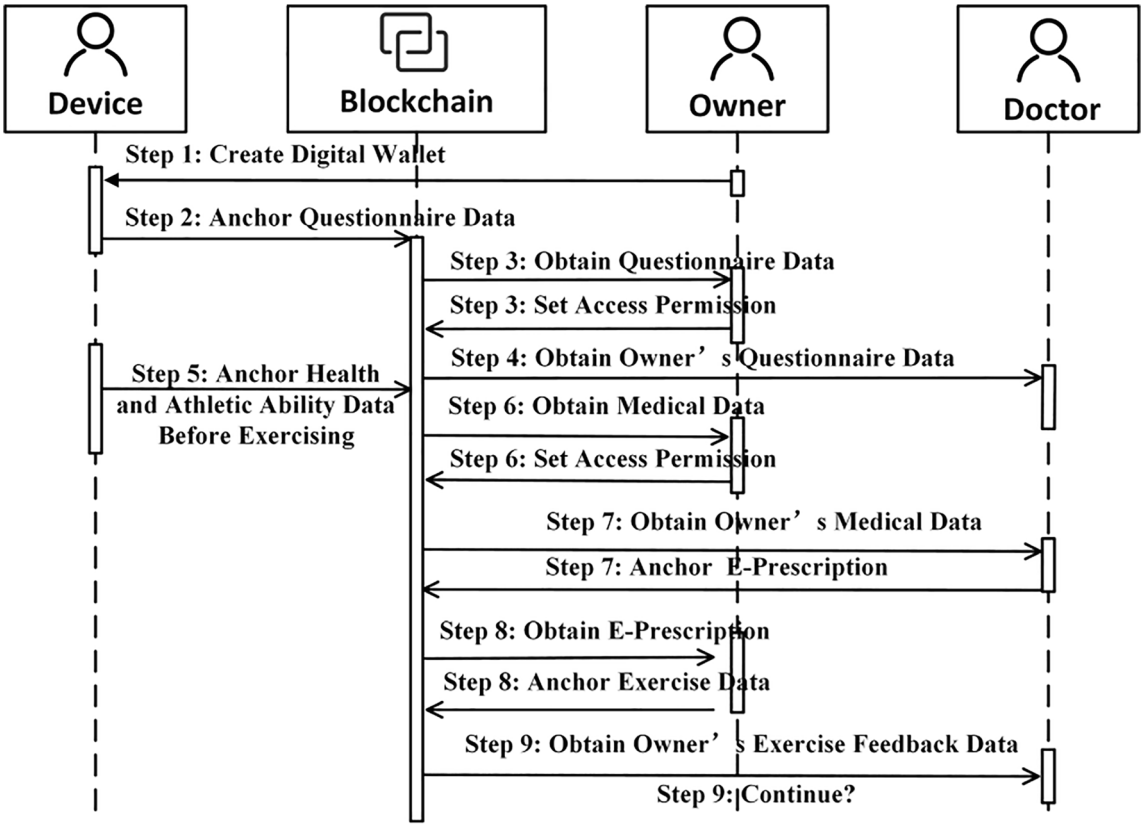

Fig. 7 Sequence diagram of the closed-loop approach based on blockchain for chronic disease management

Security analysis

In this section, we provide security analysis for how Med-PPPHIS prevents potential threats and works securely based on several assumptions clarified in "Assumptions".

Double spending attack

Double-spending is the core problem faced by any cryptocurrency, where an adversary holding $1 gives his $1 to two different users [51]. Med-PPPHIS prevents double-spending by Fork Penalty, Consensus Deposit and DPoS-Quorum consensus.

First, Fork Penalty is reserved in each Transaction Block. When a double-spending occurs, the NPMSs solve the problem by the DPoS-Quorum consensus (see "DPoS-Quorum Consensus"). When $\lambda = 10$ and $Q_e = 200$, Q_{ID} should be satisfied as follows:

$$\begin{cases} Q_{ID} > 2 \times a_{\max} \\ Q_{ID} \leq h_{\min} \\ 2 \times Q_{ID} > all_{\max} \end{cases}.$$

If and only if a consensus can be reached in a proposal, $Q_{ID} = 123$, as show in the Table 2, (Maximum, minimum and total identities generated by malicious, honest and all NPMSs are denoted as $a_{\max}$, $h_{\min}$ and $all_{\max}$ respectively). The Q_e represents the expected size of consensus committee and the Q_{ID} represents the number of the honest identities.

All NPMSs participating in the consensus have the opportunity to obtain the reserved Fork Penalty. If malicious stations are found, their Fork Penalty will be confiscated by other consensus-participating stations. If the Fork Penalty of a NPMS is not enough to meet the minimum requirement, the NPMS is no longer allowed to participate in consensus, which means that it cannot earn consensus benefits anymore.

Sybil attack

If there is no trusted public key infrastructure in a system, a malicious node can simulate many virtual nodes, thereby creating a large set of sybils. An entity could create hundreds of nodes on a single machine [52].

However, since the identities of NPMSs in consensus process are created in proportion to their tokens, adding extra nodes into the network will not gain an attacker extra vote. And if a new NPMS wants to participate in Med-DLattice, it has to wait until another NPMSs to review. Therefore, there is no advantage to be gained via a Sybil attack.

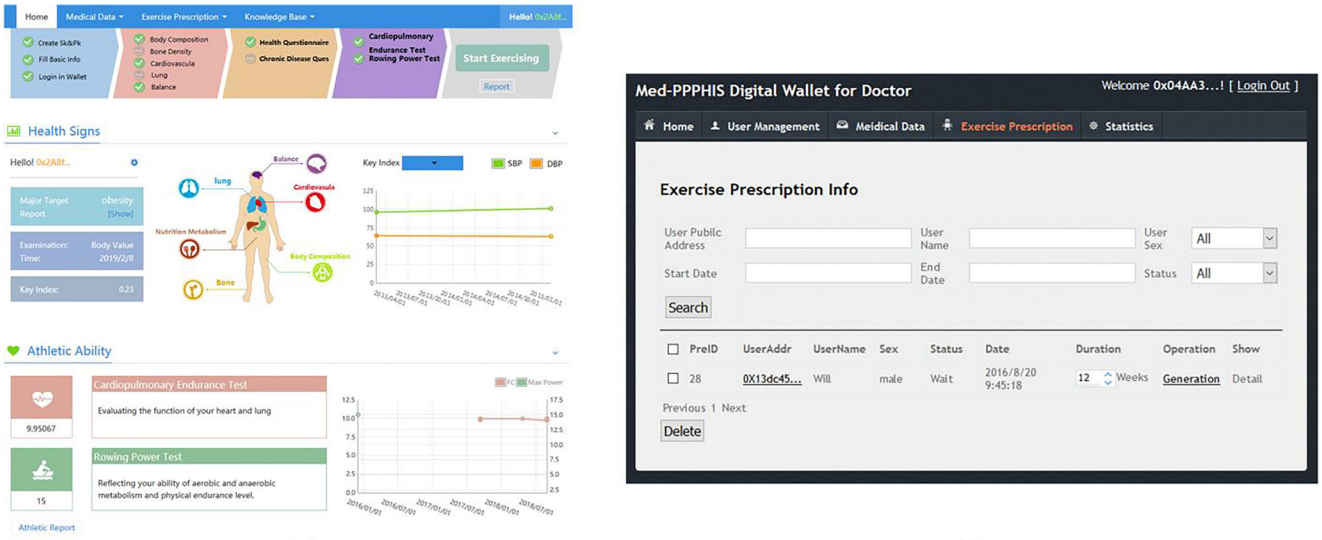

Fig. 8 **a** Interface of the users' digital wallet; **b** shows the interface of the doctors' digital wallet

DDoS attack

A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic [53].

The Verifiable Random Function (VRF) is used to generate consensus identities secretly and locally. Theses identities satisfy that each node can only calculate its own consensus identities instead of being calculated in advance by other nodes, while other nodes can verify these identities only after being broadcasted. Therefore, a consensus identity is non-interactively selected based on VRF, which has a posteriority to prevent DDoS attacks and collusion between identities.

Storage challenge

In the process of data storage and acquisition, there is a probability that a NPMS charges storage fees but not provide data services (the NPMS may go offline, or lose stored data), or a data producer stores data without paying (not all NPMSs that store the data are listed in $T_{routing}$ of TB_{send}).

This model prevents dishonest producers and malicious stations by setting Storage Deposit and starting storage challenge based on DPoS-Quorum consensus. When data producers are dishonest, the NPMSs propose:

$$Pops_{storage} =< P_s, P_p, scrt, EF, PoW, Sig >,$$

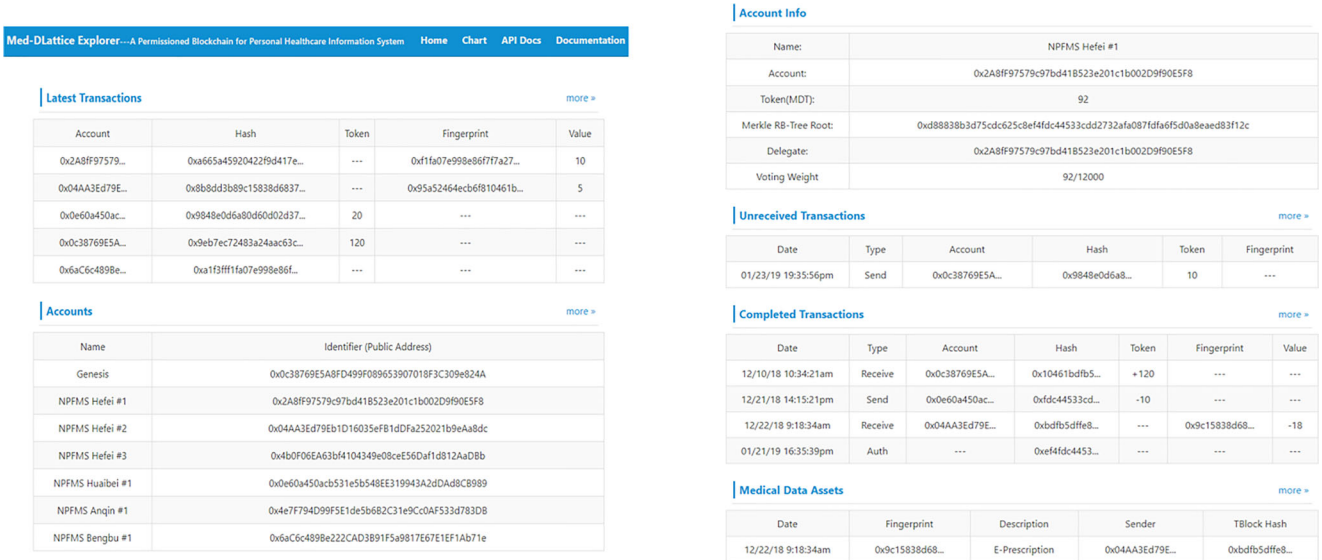

Fig. 9 Interface of Med-DLattice explorer that provides detailed information about Med-DLattice

Table 4 Implementation parameters

Parameter	Meaning	Value
λ	Security Parameter	10
h	Weight of *MDT* held by the honest NPMSs in the total *MDT*	4/5
MDT_{total}	Total *MDT* in Med-DLattice	12,000
Q_e	Expected size of consensus committee	200
τ_{good}	Threshold of the honest NPMSs in total NPMSs	2/3
Q_{ID}	Number of the honest identities	123
δ_{term}	Maximum time of each term	20

where P_s represents the public key of the data storage station, P_p represents the public key of the data producers and <*scrt*, *EF*> represents the stored secret shards;

When stations are malicious, the proposal is:

$$Pops_{storage} =< P_s, P_p, H_{send}, PoW, Sig >,$$

where H_{send} represents the hash value of TB_{send} and $T_{routing}$ in the TB_{send} can verify the NPMS indeed stores the secret shards.

If a storage challenge is reached between the stations, the Storage Deposit of the challenged station will be obtained by the stations participating in the storage challenge. If the Storage Deposit of a NPMS is not enough to meet the minimum requirement, the NPMS will no longer provide storage services, which means the benefits of storing data are not available for it.

Hostage byte

The hostage byte attack is a storage-specific attack where malicious NPMSs refuse to transfer shards, or portions of shards, in order to extort additional payments from data owners [54].

Data owners should protect themselves against hostage byte attacks by storing ∂ secret shards in NPMSs, and they only need t shards to recover the data, where $\partial \in [2f+1, N]$ and $t = f+1$. Redundant storage is not a complete solution for this attack, but addresses the vast majority of practical applications of this attack. Defeating redundancy requires collusion across multiple malicious nodes, which is difficult to execute in practice.

Table 5 Medical Data Types and Sizes in Med-PPPHIS Model

Data Type	Raw Data Size(B)	Encrypted Data Size(B)
Questionnaire Data	≈118	≈418
Health Signs Data	≈801	≈1, 548
Athletic Ability Data	≈362	≈826
E-Prescription Data	≈1,890	≈3, 334
Other Data	Depend on the specific situation	

Data protection chain

Once the data producer generates the data or the data owner obtains the data, the data will first be saved in the local database as a cache so that it can be obtained directly on the next access. After the data producers store the data distributedly, the data attribute information is anchored on the permissioned blockchain. However, the possibility of the collusion among the member stations to tamper the data is not excluded in the permissioned blockchain (Although the possibility is relatively small), therefore, the Med-PPPHIS model periodically anchors the data snapshots of Med-DLattice to the public chain. The anchored data snapshots on the public chain protect the data on Med-DLattice, and the data on the Med-DLattice further verifies the data cached in the local database, thus forming a chained medical data protection mechanism from local cache to the permissioned blockchain and then to the public chain. The risk of data being falsified and forged will be greatly reduced, providing a guarantee for the tokenization of medical data.

Forgery and modification attack

The original medical data is signed by data producer and encrypted with the public key of data owner at generation source of data, thus other nodes in the permissioned chain fail to be falsified either in transferring process or storing process. Then, the data owner will receive the successfully verified data through the transaction block. Once being discovered as falsified data, the data is to be denied, following that, the data producers will not get relevant benefits as they expect. Shamir's secret sharing mechanism makes a part of the data be stored in each permissioned blockchain node. Consequently, even the collusion occurred among all the malicious nodes, the integrated encrypted data can neither be obtained nor be falsified. On top of that, thanks to the consensus mechanism, the agreement on the blockchain structure among the nodes will be reached, and the data will also not be falsified through fork. At the same time, the snapshot of permissioned blockchain could be anchored in the public blockchain, resulting lower probability to be falsified. Therefore, the proposed Med-PPPHIS model is secure against data forgery and modification attack.

Efficiency analysis

We run several experiments with different settings on AliCloud servers to measure the storage, latency and

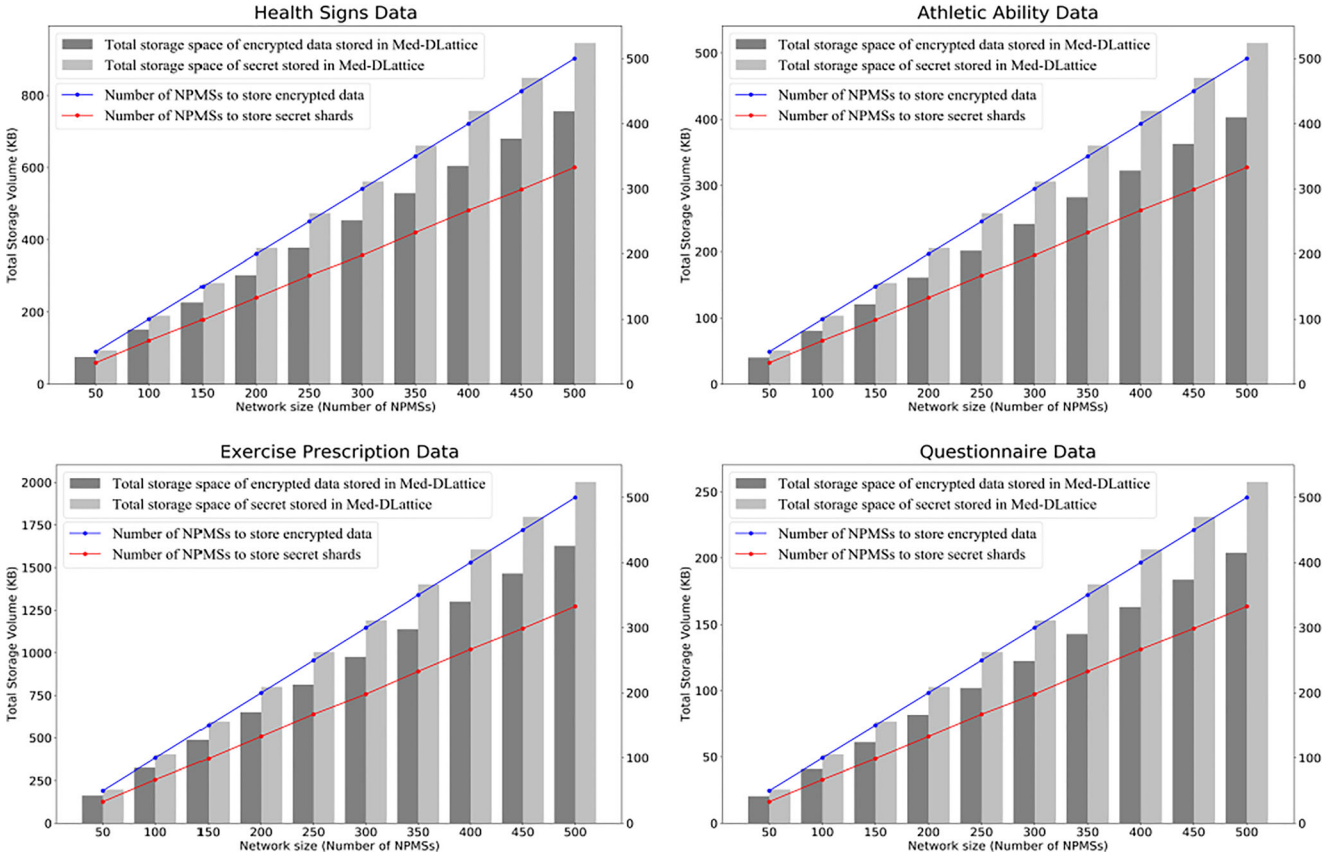

Fig. 10 Comparison between storage space of NPMSs storing ∂ secret shares and that of encrypted medical data stored in each NPMSs

throughput of Med-PPPHIS. Each AliCloud ECS instance has 4 AliCloud vCPUs and 8 GB of memory.

Storage

In traditional blockchain systems, integrated and same blockchain data need to be stored in each blockchain node. However, the encrypted original medical data in the proposed model is divided to N secret shards, of which any ∂ are distributed in NPMSs, which means these shards are stored in at most ∂ NPMSs. Although the storage space for secret shards is larger than that required by each NPMS to store all the data as shown in Fig. 10, each NPMS is only capable of storing data yet fails to recover original medical data independently, if and

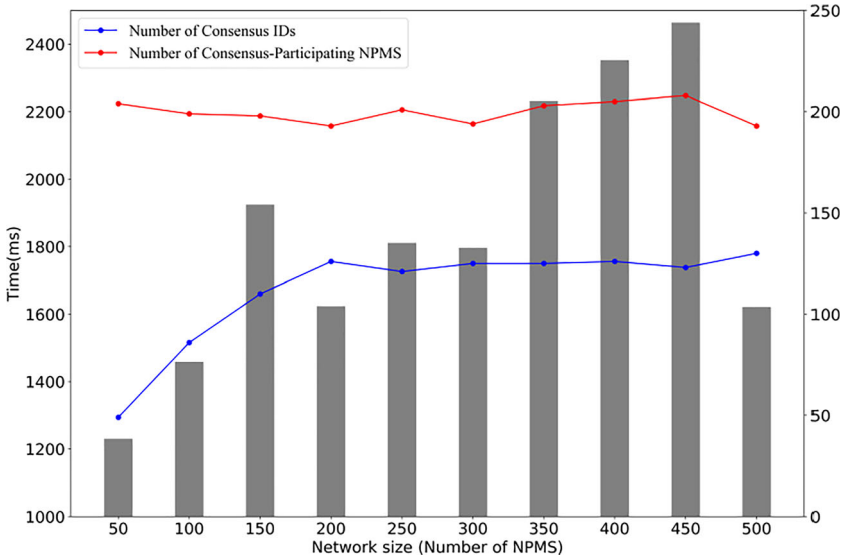

Fig. 11 Latency to reach consensus using DPoS-Quorum with 50 to 500 NPMSs. The red broken line represents the number of identities participating in the consensus, and the blue line represents the number of consensus-participating NPMSs

Table 6 Total time for verifying 1000 consensus votes and average time for verifying each consensus vote by different signature algorithms

Count = 1000	Total (ms)	Avg (ms)
ECDSA [46]	227.7	0.2278
Fp254BNb [48]	999.9	0.9999
DQ-Fp254BNb	5.5976	0.0056
Fp3982_1 [48]	2700.5	2.7005
DQ-Fp3982_1	7.7285	0.0077
BLS12_381 [48]	2516.6	2.5167
DQ-BLS12_381	7.7323	0.0073

if only at least any t of N shares is collected, the original data can be recovered successfully. It guarantees that the byzantine nodes cannot collude to obtain original data, thus protecting the privacy of the data. Meanwhile, only the attribute information of the data is recorded on the chain, which downsizes the blockchain significantly. It is worth noting that the size of secret shards gained by different Shamir's Secret Sharing Scheme. [55] is adopted in the implementation. Table 5 lists the size of raw medical data and the encrypted data.

Latency

Latency is the amount of time that it takes from the creation of a transaction until the initial confirmation of it being accepted by the network [56]. The latency of transaction in Med-DLattice is instantaneous, so we just consider the consensus latency in this section. The consensus latency is composed of three parts: the generation time of consensus identities, the propagation time of consensus messages, and the counting time of consensus votes. During the experiment, when the number of NPMSs increases from 50 to 500, using up to 25 ECS instances, and the corresponding voting weight decreases from 240 to 24, the consensus latency is shown in Fig. 11. Each AliCloud ECS instance is shared by at most twenty nodes.

It is worth noting that in the process of counting consensus votes, if the signatures are signed by ECDSA, they need to be verified separately for each vote. However, in DPoS-Quorum, these signatures can be first aggregated and then verified together, greatly reducing the time of signature verification. Time for verifying 1000 consensus votes' signature by using ECDSA, normal BLS and DPoS-Quorum Consensus respectively, are listed in Table 6. It can be seen from the comparison results that the signature algorithm in "DPoS-Quorum Consensus" (DQ-Fp254BNb, DQ-Fp3982_1 and DQ-BLS12_381) greatly reduces the time of signature verification, whether or not compared with ECDSA or normal BLS.

Throughput

One of the end goals of blockchain is to replace the current infrastructure (like financial backend of many institutions around the world, which handles thousands of transactions per second (TPS)), it will need to scale to meet and/or exceed the TPS. A higher throughput will also open the doors to more interesting and intensive applications of blockchain technology [57].

During the experiment, each NPMS will run on 25 ECS instances accordingly, to build the network of Med-DLattice. We measured the throughput of different types of transaction blocks separately as shown in Fig. 12. According to the experimental results, the throughput is not the same due to the different size of TB. It is worth noting that the size of TB_{auth} and TB_{send} depends on the actual circumstance. Because as the amount of authorization increases, the size of TB_{auth} to rise (only contains an authorization in this experiment); Similarly, with the number of NPMSs increases, the storage information

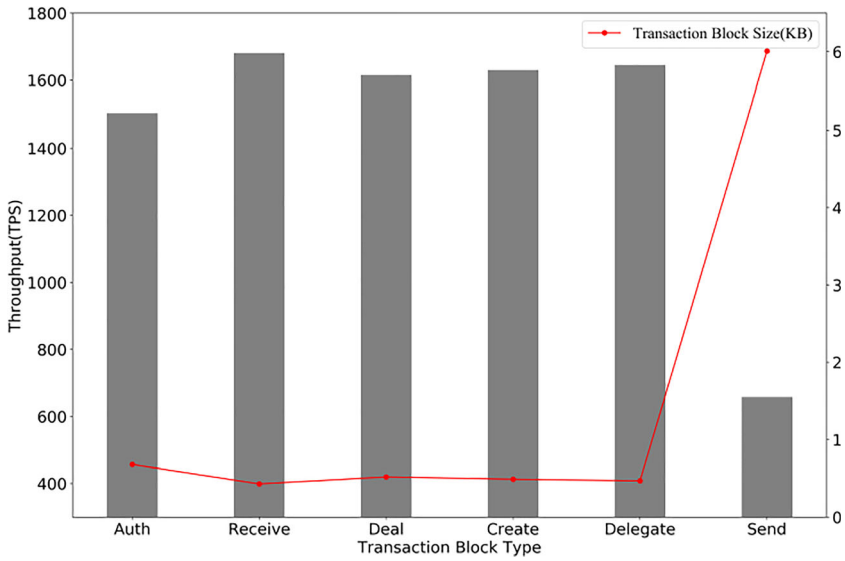

Fig. 12 Throughput of different transaction block. The red broken line represents the size of transaction blocks

Table 7 Comparison between Med-PPPHIS model and current cutting-edge solutions

	Blockchain-Based	Underlying Blockchain	Tamper-Resistance	Privacy Protection	Off-chain Storage	Incentive	Latency (S)	Throughput (TPS)
[20]	N	–	N	N	–	–	–	–
MedRec [10]	Y	Ethereum variant	Y	N	N	Y	Depend on underlying consensus algorithm and blockchain structure	
MeDShare [7]	Y	Ethereum variant	Y	Y	N	N		
MedBlock [9]	Y	Their own blockchain	Y	Y	N	N		
MIStore [8]	Y	Ethereum + PBFT	Y	Y	N	N		
SIFF [60]	Y	Fabric Blockchain	Y	Y	N	N		
TP-HER [61]	Y	Ethereum	Y	Y	N	N		
This Work	Y	Med-DLattice	Y	Y	Y	Y	≈1.46	≈1000

contained in TB_{send} will increase correspondingly (the storage information of 17 secret shards are stored in this experiment). Compared to the livenet data of Bitcoin and Ethereum, (Bitcoin network processes up to 3 TPS (From Block Height 556,800 to 556,810) [58] and Ethereum processes up to 127 TPS (From Block Height 7,002,602 to 7,002,612) [59]), the throughput of Med-DLattice is quite impressive.

Comparison to related system

Table 7 below compares our Med-PPPHIS model with other existing systems and literature presented in this paper. By contrast, we find that existing blockchain-based medical systems store raw medical data on the chain and lack effective incentives to encourage users to participate in the system (the incentives here refer not only to the rewards of mining, but also to motivate all users to participate in the system). MedRec [10] incentivize miners to participate in their network based on two incentivizing models. The first is based on the Ethereum's inherent incentives and the second rewards the participants with anonymous medical data. However, it is illegal to gather patient data together and share them as rewards. In the proposed model, all users willingly participate in the system and share the medical date by tokenizing the medical data on the chain, so as to gain benefits. Regarding latency and throughput, each blockchain-based medical system's performance is limited by the underlying blockchain architecture. As stated in MeDShare [7] and MedBlock [9] paper, the latency is 1286.73 s and 925.12 s respectively when 100 users participate in the system. Although the latency includes all steps

Table 8 The current problems and the corresponding resolution methods from the proposed model

Dimensions	Problems	Solutions
Data Protection	1. Privacy disclosure	1: The encrypted medical data is divided into several secret shards and stored in Med-DLattice nodes dispersedly. The unrelated users cannot retrieve the original medical data, thus protecting data privacy;
	2. Without access control	2: Only the owner of medical data has access control permission of data. And the allocation of access control is realized by proxy re-encryption technology;
	3. Hacker attack	3: After the decentralized storage, the data attribute information is protected on Med-DLattice, and the snapshot of Med-DLattice is anchored to the public chain periodically, so as to prevent it from being attacked by hackers;
	4. Data loss	4: Medical data is divided into several secret shards and distributed in Med-DLattice nodes randomly and redundantly to prevent data loss;
Data Usage	1. Data fraud	1: Medical data is stored in each node of Med-DLattice, while data attribute information is anchored on the public ledger, so it is difficult to forge data.
	2. Data abuse	2: Accessing medical data requires permission from the data owner to prevent the misuse of data;
	3. Information asymmetry	3: As a public ledger, Med-DLattice records medical data on the chain and makes it public, which solves the problem of information asymmetry between multiple parties and avoids false statements, improper charges and other problems;
User Engagement	1. Managing health data inconveniently 2. Sharing data inefficiency	1 and 2: By converting the medical data into on-chain tokens, the circulation speed of data between multiple parties is also accelerated, and the development of biomedical and health care domains is advanced.

required to process a request by all entities in the system, compared with proposed model in this paper, the gap is still obvious. The current problems and the corresponding resolution methods from the proposed Med-PPPHIS model are shown in Table 8.

Conclusion

In this paper, we propose a model called Med-PPPHIS, which consists of a permissioned blockchain and a public blockchain, to serve the management of user's personal health information. As the permissioned blockchain in Med-PPPHIS, the Med-DLattice features a DAG structure, where each account updates its own Account-DAG asynchronously to other unrelated accounts, thus improving the throughput of the permissioned blockchain. The core of Med-DLattice is to use the proposed DPoS-Quorum consensus to help the nodes reach an effective consensus. The Med-DLattice nodes store raw medical data and record the data attribute information on the permissioned blockchain, and periodically anchor the data snapshots to the public blockchain, so as to form a chained protection mechanism for medical data. Therefore, Med-PPPHIS has the characteristics of decentralization, transaction transparency, tamper-resistance, low latency and high throughput. By converting the medical data into on-chain tokens, a safe and efficient channel for data sharing is established, while the privacy of data is protected. The above advantages of Med-PPPHIS effectively address the pain points in current medical data management. We implement a prototype of Med-PPPHIS and introduce a closed-loop method for chronic disease management based on blockchain, which initially applies it to national physique monitoring in Anhui Province, China. This model can also be applied to digital archives management, legal cases management and other specific scenarios.

The shortcoming of this paper is that: (i) keyword query for medical data is not supported. If users have a large amount of medical data, the data query on the chain is not efficient enough; (ii) cancellation of data access permission is not supported; (iii) in Med-PPPHIS model, the public chain anchored by the permissioned chain needs to be specified in advance and cannot be dynamically selected.

In the future research, we hope to solve the problems mentioned above, especially the dynamical selection of public blockchain, which means the permissioned blockchain is able to dynamically select the most suitable public chain for anchoring based on comprehensive assessment of the health status of the public chain, the anchoring cost and other factors.

Funding This study was funded by the National Natural Science Foundation of China (No. 61602435), Natural Science Foundation of Anhui Province (No. 1708085QF153), and Anhui Provincial Science and Technology Major Project (No. 16030901057).

Compliance with ethical standards

Conflict of interests Tong Zhou, Xiaofeng Li and He Zhao declare that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. health research: the early personal health record experience, Journal of Medical Internet Research 12(2):1–10, 2010. https://doi.org/10.2196/jmir.1356.
2. Kuo, T. T., Kim, H. E., and Ohno-Machado, L., Blockchain distributed ledger technologies for biomedical and health care applications. J Am Med Inform Assoc 24(6):1211–1220, 2017. https://doi.org/10.1093/jamia/ocx068.
3. Wood, G., Ethereum: A secure decentralised generalised transaction ledger, *Ethereum project yellow paper*, 2014, Available: http://www.ibm.biz/blockchainhealth, Accessed Nov. 2018.
4. Shao, Q., Jin, C., Zhang, Z., Qian, W., and Zhou, A., Blockchain: Architecture and research Progress. Chinese Journal of Computers 41(5):969–988, 2018. https://doi.org/10.11897/SP.J.1016.2018.00969.
5. Zhou, T., Li, X., and Zhao, H., EverSSDI: Blockchain-based framework for verification, authorization and recovery of self-sovereign identity using smart contracts. Int. J. Computer Applications in Technology In Press.
6. Sheng, N., Li, F., Li, X., Zhao, H., and Zhou, T., Data capitalization method based on blockchain smart contract for internet of things. Journal of Zhejiang University (Engineering Science) 52(11):2150–2153, 2018. https://doi.org/10.3785/j.issn.1008-973X.2018.11.014.
7. Xia, Q., Sifah, E. B., Asamoah, K. O., Gao, J., Du, X., and Guizani, M., MeDShare: Trust-less medical data sharing among cloud service providers via Blockchain. IEEE Access 5(99):14757–14767, 2017. https://doi.org/10.1109/ACCESS.2017.2730843.
8. Zhou, L., Wang, L., and Sun, Y., MIStore: A Blockchain-based medical insurance storage system. Journal of Medical Systems 42(8):149, 2018. https://doi.org/10.1007/s10916-018-0996-4.
9. Fan, K., Wang, S., Ren, Y., Li, H., and Yang, Y., MedBlock: Efficient and secure medical data sharing via Blockchain. Journal of Medical Systems 42(8):136, 2018. https://doi.org/10.1007/s10916-018-0993-7.
10. Azaria, A., Ekblaw, A., Vieira, T., and Lippman A., MedRec: Using Blockchain for medical data access and permission management. Presented at *International Conference on Open and Big Data*, 2016. Available: http://ieeexplore.ieee.org/document/7573685/.
11. Tang, H., Zhou, T., Zhao, H., Zhao, Z., Wang, W., and Zhang, Z., Archives data protection and sharing method based on Blockchain. Journal of Software:1–15, 2019. https://doi.org/10.13328/j.cnki.jos.005770.
12. Dinh, T. T. A., Rui, L., Zhang, M., Chen, G., Chin, B., and Wang, J., Untangling Blockchain: A data processing view of Blockchain systems. IEEE Transactions on Knowledge & Data Engineering (99):1–1, 2017. https://doi.org/10.1109/TKDE.2017.2781227.
13. Multichain: Open platform for blockchain applications, Available: https://www.multichain.com/, Accessed Dec. 2018.
14. Serguei, P., The tangle, Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/

45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf/, Accessed Sep. 2018.
15. Xu, H., Zhou, T., Ma, Z., and Zhou, D., The study on comprehensive evaluation system of health signs based on internet. Journal of Biomedical Engineering Research 32(4):217–223, 2013. https://doi.org/10.19529/j.cnki.1672-6278.2013.04.004.
16. Xu, J., Zhao, H., Wang, W., Zhang, Z., Li, X., Zhou, T., and Ding, Z., National Physical Fitness Monitoring System. Computer Systems & Applications 26(10):61–66, 2017. https://doi.org/10.15888/j.cnki.csa.005980.
17. Zhou, T., Yuan, M., Zhao, H., Wang, W., Zhang, Z., and Ma, Z., Chronic disease tracking client based on iOS. Computer Systems & Applications 25(9):73–78, 2016. https://doi.org/10.15888/j.cnki.csa.005319.
18. Ma, D., Tan, H., Zhao, H., Zhou, T., Wang, W., Zhang, Z., and Li, X., National Physical Monitoring and scientific fitness exercise guidance client based on iOS. Computer Technology and Development 27(12):161–165, 2017. https://doi.org/10.3969/j.issn.1673-629X.2017.12.035.
19. Lemai, N., Emilia, B., Linh, T., and Nguyen, Electronic health records implementation: An evaluation of information system impact and contingency factors. International Journal of Medical Informatics 83(11):779–796, 2014. https://doi.org/10.1016/j.ijmedinf.2014.06.011.
20. Hassan, M. M., Lin, K., Yue, X., and Wan, J., A multimedia healthcare data sharing approach through cloud-based body area network. Future Generation Computer Systems 66(May):48–58, 2017. https://doi.org/10.1016/j.future.2015.12.016.
21. John, H., Gunnar, R., Kaori, S., Kenrick, T., George, O., Annah, W., Shahnaaz, S., and Tomohiko, S., Implementation of a cloud-based electronic medical record for maternal and child health in rural Kenya. International Journal of Medical Informatics 84(5):349–354, 2015. https://doi.org/10.1016/j.ijmedinf.2015.01.005.
22. Esposito, C., Santis, A. D., Tortora, G., Chang, H., and Choo, K. K. R., Blockchain: A panacea for healthcare cloud-based data security and privacy? IEEE Cloud Computing 5(1):31–37, 2018. https://doi.org/10.1109/MCC.2018.011791712.
23. Nakamoto, S., Bitcoin: A peer-to-peer electronic cash system, Available: http://bitcoin.org/bitcoin.pdf, 2008.
24. Tierion And Philips Bring Blockchain Technology to Healthcare Sector, Available: https://bitcoinist.com/tierion-philips-bring-blockchain-techn-ology-healthcare-sector/, Accessed: Jan. 2019.
25. Healthbank, Available: https://www.healthbank.coop/, Accessed Aug. 2018.
26. Change Healthcare, Available: https://www.changehealthcare.com/, Accessed Sep. 2018.
27. Alibaba's Online Health Service to Pilot Blockchain Solutions for Health Treatments in Changzhou, Available: https://www.yicaiglobal.com/news/alibaba%E2%80%99s-online-health-service-pilot -blockchain-solutions-health-treatments-Changzhou, Accessed Sep. 2018.
28. Tencent introduced blockchain medical prescription: shaping the future of China's healthcare, Available: https://bcfocus.com/news/hacker-hacks-wannabe-hackers-the-most-ridiculous-crypto-story-ever/6188/. Accessed Sep. 2018.
29. Zyskind, G., Nathan, O., and Pentland, A. S., Decentralizing privacy: Using Blockchain to protect personal data. In: *IEEE Security and Privacy Workshops*, 2015. https://doi.org/10.1109/SPW.2015.27.
30. Li, M., Yu, S., Zheng, Y., Ren, K., and Lou, W., Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption. IEEE Transactions on Parallel & Distributed Systems 24(1):131–143, 2013. https://doi.org/10.1109/TPDS.2012.97.
31. Guo, R., Shi, H., Zhao, Q., and Zheng, D., Secure attribute-based signature scheme with multiple authorities for Blockchain in electronic health records systems. IEEE, 2018. 10.1109/ACCESS.2018.2801266.
32. On Public and Private Blockchains, Available: https://blog.ethereum.org/2015/08/07/on-public-and-privateblockchains/, Accessed Oct. 2018.
33. Zhou, T., Li, X., and Zhao, H., DLattice: A permission-less Blockchain based on DPoS-BA-DAG consensus for data tokenization. IEEE Access 7:39273–39287, 2019. https://doi.org/10.1109/ACCESS.2019.2906637.
34. C. Wong, Patricia Tree, Available: https://github.com/ethereum/wiki/wiki/Patricia-Tree. Accessed: Mar. Dec., 2018.
35. Red-Black Merkle Tree, Available: https://github.com/amiller/redblackmerkle. Accessed Nov. 2018.
36. Micali, S., Rabin, M., and Vadhan, S., Verifiable random functions. In Proceedings of the 40th annual IEEE Symposium on Foundations of Computer Science (FOCS), New York 1999. 10.1109/SFFCS.1999.814584.
37. Stepan, BLS signatures: better than Schnorr. Available: https://medium.com/cryptoadvance/bls-signatures-better-than-schnorr-5a7fe30ea716. Accessed: Dec. 24, 2018.
38. Antonopoulos, A. M., Mastering bitcoin: Unlocking digital cryptocurrencies. O'Reilly Media, Inc, 2014.
39. Shamir, A., How to share a secret. Communications of the ACM 22(11):612–613, 1979. https://doi.org/10.1145/359168.359176.
40. Wang, Y., Cao, Q., Zhang Z., Wang, W., Liu, B., Chen, M., Li, X., Tang, C., Zhan, L., Sun, Y., and Ma. Z., A system of generating exercise prescription based on multi-source information, China Patent No. CN104077737A.
41. Cao, Q., Wang, Y., Chen, Y., Ding, Z., Li, M., Xu, J., Zhao, H., Li, X., He, Z., Xu, Y., Ma, B., Sun, Y., and Ma, Z., A system of inferencing exercise target based on multi-source information, China patent no. CN104123445B.
42. Golang 1.1.5, Available: https://golang.org/.Accessed: Jan. 03, 2019.
43. An implementation of the LevelDB key/value database in the Go, Available: https://github.com/syndtr/goleveldb, Accessed Dec. 2018.
44. Libp2p, Available: https://github.com/libp2p. Accessed: Dec. 2018.
45. NodeJS 11.8.0, Available: https://nodejs.org/en/, Accessed Dec. 2018.
46. Redis 5.0.3, Available: https://redis.io/, Accessed: Dec. 2018.
47. An implementation of ECIES and ECDSA in Go, Available: https://github.com/ethereum/go-ethereum/tree/master/crypto, Accessed: Jan. 2019.
48. An implementation of VRF in Go, Available: https://github.com/r2ishiguro/vrf/, Accessed: Jan. 2019.
49. An implementation of BLS in Go, Available: https://github.com/dfinity/go-dfinity-crypto, Accessed Jan. 2019.
50. AFGH Proxy Re-encryption, Available: https://github.com/zerodb/zerodb-afgh-pre, Accessed Jan. 2019.
51. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., and Zeldovich N., Algorand: Scaling byzantine agreements for cryptocurrencies, In: Proceedings of the 26th symposium on operating systems principles, pp 51–68, ACM, 2017. 10.1145/3132747.3132757.
52. Douceur, J. R., The Sybil attack. In Proceedings of the 1st international workshop on peer-to-peer systems (IPTPS'02), Springer, Berlin, 2002, 10.1007/3-540-45748-8_24.
53. DDOS Attack, Available: https://en.wikipedia.org/wiki/Denial-of-service_attack, Accessed Oct. 2018.
54. Storj Labs Inc., Storj:A Peer-to-Peer Cloud Storage Network Available: https://storj.io/storjv3.pdf, Accessed Jan. 2019.
55. An implementation of Shamir's Secret Sharing Algorithm in Go, Available: https://github.com/SSSaaS/sssa-golang, Accessed Jan. 2019.
56. A. Grigorean, "Latency and finality in different cryptocurrencies," Accessed: Jan. 04, 2019. Available: https://hackernoon.com/

latency-and-finality-in-different-cryptocurrencies-a7182a06d07a. Accessed Dec. 2018.
57. Zilliqa: A High Throughput Scalable Blockchain? Available: https://medium.com/@curiousinvestor/zilliqa-a-high-throughput-scalable-blockchain-60e355d873c5. Accessed: Jan. 04, 2019.
58. Bitcoin Explorer, Available: https://btc.com/. Accessed: Jan. 03, 2019.
59. Ethereum Explorer, Available: https://etherscan.io/. Accessed: Jan. 03, 2019.
60. Chen, Y., Ding, S., Xu, Z., Zheng, H., and Yang, S., Blockchain-based medical records secure storage and medical service framework. J Med Syst 43:5, 2019. https://doi.org/10.1007/s10916-018-1121-4.
61. Cao, S., Zhang, G., Liu, P., Zhang, X., and Neri, F., Cloud-assisted secure eHealth systems for tamper-proofing EHR via blockchain. Information Sciences 485:427–440, 2019. https://doi.org/10.1016/j.ins.2019.02.038.