Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

PMHE: a wearable medical sensor assisted framework for health care based on blockchain and privacy computing

Jindong Zhao, Wenshuo Wang*, Dan Wang, Xuan Wang and Chunxiao Mu

Abstract

Nowadays, smart medical cloud platforms have become a new direction in the industry. However, because the medical system involves personal physiological data, user privacy in data transmission and processing is also easy to leak in the smart medical cloud platform. This paper proposed a medical data privacy preserving framework named PMHE based on blockchain and fully homomorphic encryption technology. The framework receives personal physiological data from wearable devices on the client side, and uses blockchain as data storage to ensure that the data cannot be tampered with or forged; Besides, it uses fully homomorphic encryption method to design disease prediction models implemented by smart contracts. In PMHE, data is encoded and encrypted on the client side, and encrypted data is uploaded to the cloud platform via the public Internet, preventing privacy leakage caused by channel eavesdropping; smart contracts run on the blockchain platform for disease prediction, and the operators participating in computing are encrypted user data too. So, privacy and security issues caused by platform data leakage are avoided. The client-to-cloud interaction protocol is also designed to overcome the defect that fully homomorphic encryption only supports addition and multiplication by submitting tuples on the client side, to ensure that the prediction model can perform complex computing. In addition, the design of the smart contract is introduced in detail, and the performance of the system is analyzed. Finally, experiments are conducted to verify the operating effect of the system, ensuring that user privacy is not leaked without affecting the accuracy of the model, and realizing a smart medical cloud platform in which data can be used but cannot be borrowed.

Keywords: Blockchain, Homomorphic encryption, Smart contracts, Privacy computing, Smart medical

Introduction

In recent years, with the application of new technologies such as smart healthcare and mobile healthcare, medical data, such as electronic health records, clinical measures, personal health status records perceived by wearable sensors, have all shown explosive growth [1, 2]. In online medical system, the procedure of authorization distribution, transmission and processing of data involves not only data exchange and transmission technology, but also the privacy security of data source [3, 4] and the trust of multiple nodes participating in data sharing [5]. Furthermore, in the mobile and health medical service system, People's awareness of data privacy preserving is weak, and attackers tend to connect users' medical data with network behaviors, which makes the impact of medical privacy disclosure more serious [6].

With the development of blockchain technology, it has gradually been applied in the medical field. The main application areas of blockchain include the secure sharing and privacy protection of medical data, among which the privacy protection of medical and health data is the research focus [7]. The essence of blockchain technology is decentralization, which ensures that medical data will not be manipulated or damaged in an environment

*Correspondence: wangwsh202121@163.com

School of Computer and Control Engineering, Yantai University, Yantai 264005, China

of mutual distrust. The encryption algorithm provides anonymity for blockchain and protects the privacy of patients' information. In view of the privacy disclosure and information islands of medical data, the decentralized, distributed storage, anonymity and other features of blockchain not only guarantee the privacy and security of medical data, but also provide possibilities for the safe transfer of medical data, which has attached extensive attention of researchers in the medical field.

Blockchain ensures that data cannot be tampered with or forged. However, on the smart medical cloud platform deployed on Internet, users need to upload physiological data to the cloud platform through the public link. And then physiological data will be evaluated on cloud platform using the preset model to predict diseases and monitor health. There is a risk of data leakage during this process. In addition, models on cloud platforms require user data as raw formation for computing. Cloud platforms are untrusted third parties, and data may be leaked on cloud platforms.

In this paper, we present a smart medical cloud platform framework based on privacy computing, and its abbreviation is PMHE. The platform calculates users' physiological data that is encrypted on client side, and uses the results to predict diseases. It can protect users' privacy in communication and computing. Although the data is encrypted, the accuracy of the model is not affected.

Contributions

PMHE utilizing blockchain and fully homomorphic encryption technology builds an intelligent medical data and privacy protection framework to solve many problems. Those problems include privacy disclosure and data tampering caused by data loss and hacking.

The advantages mentioned above enable the data consumers and data providers to realize the trust transaction of medical data on the platform without the trust endorsement of the third-party platform [8]. People can enjoy the disease prediction, health monitoring and other medical services provided by the platform securely. The main contributions in this paper can be summarized as follows:

- Propose the PMHE framework. This is a privacy protection scheme of smart medical data based on blockchain technology and fully homomorphic encryption technology. Compared with most existing schemes, PMHE does not rely on any trusted third-party or tamper-proof hardware, but only involves portable wearable medical devices, blockchain and (untrusted) clouds. In terms of storage and computing costs, PMHE also makes significant improvements to existing medical data management solutions based on blockchain.
- Design interaction protocols with desired functionality for PMHE under the universally compostable framework. Generally, the computing model in the cloud involves complex operations, but holomorphic encryption algorithm only supports addition and multiplication. The interaction protocol allows any computation to participate in homomorphic evaluations.
- Implement PMHE scheme based on Hyperledger Fabric, and conduct comprehensive experiments to evaluate PMHE'S performance. The experimental results show that PMHE increases affordable communication costs and storage costs while providing safe, full-featured disease prediction and health monitoring functions.

Structure

This paper is divided into seven parts. Introduction section introduces the research motivation and contribution, Related Works and Preparation section introduces correlational researches and preparatory knowledge, System Architecture and Security Model section and The Framework of PMHE introduce the process and results of implementing PMHE, Discussion and Analysis section deeply studies the privacy security of PMHE, and Experiment and Evaluation section carries out specific experiments and analyzes experimental data. Finally, the paper is summarized in Summary and Conclusion section.

Related works and preparation

Related works

At present, in the field of smart healthcare, there are many researches on privacy preservation in the process of data sharing, but it is difficult to combine the privacy and availability of medical data. In 2019, Hylock et al. proposed a mixed-block blockchain framework to support immutable logging and editable patient blocks [9]. This framework presents patients and providers with access to consistent and comprehensive medical records by integrating a structured, interoperable design with patient-accumulated and generated data shared through smart contracts into a universally accessible blockchain. Proxy re-encryption (PRE) is used to improve the data security in [9], but when the medical data is trained in the cloud, the cloud service provider needs to decrypt the data using private key, and privacy is not fundamentally preserved. Ruijin Wang et al. in 2019 put forward a model of decentralized medical data sharing based on blockchain [10]. This model uses ring signature technology in blockchain to construct a private data storage

protocol, which can protect the privacy of medical data and patient identity. Although [10] implements medical information sharing with a strict access control management mechanism based on smart contract, it only shuffles the mapping between unencrypted medical data and data owner essentially. There is also a threat from online eavesdropping. In 2021, Jingwei Liu et al. proposed a privacy-preserving medical data sharing scheme based on consortium blockchain [11]. The pseudo-identity is used to preserve the users' privacy in [11], and malicious users can be tracked by the conditional anonymous tracing mechanism when malicious behavior occurs. However, [11] only lessens the damage after privacy disclosure and cannot realize real-time defense against privacy leakage like PMHE architecture. Zhou Zhengqiang et al. proposed a medical data security sharing scheme based on consortium blockchain in 2021[12]. The scheme uses consortium blockchain to store metadata and cloud storage to store ciphertext of medical data. In addition, the combination of time-limited smart contract and ciphertext-policy attribute-based encryption (CP-ABE) technology realizes fine-grained access control and secure storage of medical data. Although [12] ensures the privacy of medical data on blockchain in different ways, the operability of data is greatly reduced by using traditional encryption methods. Homomorphic encryption technology can solve this problem well.

The research on the combination of homomorphic encryption technology and machine learning is mainly focused on training the model with encrypted data. Crawford et al. [13] have made remarkable achievements in training small logistic regression models on encrypted data relying on operation "deep within the bootstrapping regime", but in his scheme, the number of features in training data is small to support disease prediction well. In 2018, Kim et al. adapted a novel homomorphic encryption scheme optimized for real numbers computation, and using new packaging and parallelization techniques, which solved the calculation problem of ciphertext logistic regression [14]. Kim leverages least squares approximation to substitute non-polynomial functions, which will greatly increase computation load inevitably. In 2020, Kangyu Lui realized logistic regression model training in encrypted data set [15]. Although Kim and Kangyu Lui evaluate the performance of the model by using real-world data sets, there is still a gap in accuracy between the machine learning with homomorphically encrypted data and that with original data. In addition, because the training of model involves a large number of calculations, introducing homomorphic encryption algorithm in training phase will greatly increase the time cost.

PMHE solves the problem of privacy disclosure in the field of smart healthcare and realizes the combination of privacy and availability of medical data. By introducing CKKS homomorphic encryption scheme, PMHE guarantees data privacy in communication and calculation, and homomorphism ensures that the medical data after encrypting can be further evaluated. Besides, computing cost of non-polynomial functions is reduced utilizing interaction protocols. Finally, experimental results show that the accuracy of AI model in PMHE is higher than existing scheme.

Blockchain and smart contract

The structural concept of blockchain was proposed as early as in the 1990s, and it was not until 2008 that Satoshi Nakamoto put forward the concept of blockchain for the first time in his published paper [16]. Blockchain can be regard as a decentralized database of blocks that can be added continuously, a self-referential data structure that is open, transparent, immutable, and traceable. Smart contract, known as Chaincode in Hyperledger, is essentially a program that runs on blockchain. And the code and data onto smart contract are stored on blockchain too. The problem of limited flexibility is solved by allowing authorized participants to manipulate applications and reach consensus on the blockchain. In addition, since the preparation of a complete smart contract involves many aspects such as privacy, security, legal issues and mechanism design [17], it is a key issue for practical application to design a safe smart contract which is fair, reliable and complies with specifications.

Fully homomorphic encryption and CKKS

As early as 1978, Rivest et al. first proposed the idea of Fully Homomorphic Encryption (FHE) [18, 19]. Since then, the study of Fully Homomorphic Encryption algorithm has been listed as a research difficulty in the field of cryptography. It was not until 2009 that Gentry constructed the world's first fully homomorphic encryption scheme based on lattice cryptography [20], which was homomorphic for fixed number of operation operations (usually called circuit depth). Then researchers began to improve on the basis of Gentry's work, and developed more perfect homomorphic encryption algorithm. CKKS homomorphic encryption scheme was proposed by Cheon et al. in 2017 [21]. It supports approximate floating-point operations and is one of the most important and suitable similar algorithms in homomorphic encryption field.

Homomorphic encryption (HE) is a cryptographic scheme that enables homomorphic operations on encrypted data without decryption. For arbitrary function f and message m, holomorphic encryption has the properties of Formula 1. *Enc* refers to the encryption algorithm, and *Dec* refers to the decryption algorithm.

$$Dec\left(f\left(Enc\left(m_1\right),Enc\left(m_2\right),\ldots,Enc\left(m_n\right)\right)\right)=f\left(m_1,m_2,\ldots,m_n\right) \quad (1)$$

Fully homomorphic encryption scheme supports both homomorphic addition and multiplication, and the number of operation rounds is unlimited. At present, homomorphic encryption technology plays an important role in privacy security of cloud computing system. After encoding and encrypting, the user stores the ciphertext in blockchain. Without the user's private key, the real plaintext data of the user cannot be obtained [22]. The detail process of full homomorphic encryption is shown in Fig. 1.

In the cloud computing system, the functions in the left box are implemented in client side, while the functions in the right box are implemented in cloud system. In operation procedure, encrypted plaintext is involved in homomorphic evaluations. After completion of operation using ciphertext, user can decrypt the result C using his/her private key, and the output of decryption operation is almost the same as computation result on original data.

Unlike other LWE-based encryption schemes, rescaling is introduced in CKKS scheme for managing the magnitude of plaintext [17]. And CKKS uses *Ecd*, *Dcd* and scaling factor Δ to map messages to plaintext. For example, the message is complex number, and the plaintext is an element on the cyclotomic polynomial ring $R^{-}=Z$ *[X] (ΦM (X))*. *Dcd* first divides the plaintext polynomial *m(X)* by factor Δ, then computes the function value of the plaintext polynomial at the root of the cyclotomic polynomial ring *ΦM (X)*, and rounds numbers to get the final complex number vector of message. *Ecd* is the inverse transformation of *Dcd* [24].

In PMHE framework, the cloud system is divided into cloud server and blockchain according to functions. The basic Web application is implemented on the cloud server, and the disease prediction AI model written by smart contract is run on the blockchain. The homomorphic operations in AI model are implemented with CKKS encryption scheme. Homomorphic encryption technology is a reliable tool and security basis for users to conduct data mining and analysis calculation in blockchain network [23]. Blockchain technology is combined with homomorphic encryption technology in PMHE, and CKKS encryption algorithm for approximate arithmetic is introduced to realize homomorphic evaluations on encrypted medical data.

System architecture and security model

This section describes the system architecture and security model in terms of desired functionality firstly, then, the design goals of privacy-protected solution defined with the desired functionality in PMHE is introduced. For the sake of description, Table 1 lists the symbols involved in PMHE.

System architecture

As shown in Fig. 2, the PMHE framework involves five types of entities: portable wearable medical devices, client apps, cloud servers, disease prediction models, and the blockchain with smart contracts. In our proposal, we combine blockchain, machine learning and homomorphic encryption technology to obtain a scheme that can analyze medical data in blockchain network. The scheme uses homomorphic encryption technology to ensure both the privacy and availability of medical data.

- Portable wearable medical devices. Portable wearable medical devices mainly refer to portable electronic devices that can be directly worn in clinical or daily health monitoring [6]. The intelligent medical devices mainly used in this paper include smart bracelets, smart watches and ECG underwear. Portable wearable medical devices can continuously collect physiological data of the human body at anytime, anywhere and in any environment. It mainly collects such data as heart rate, ECG, respiration and blood pressure

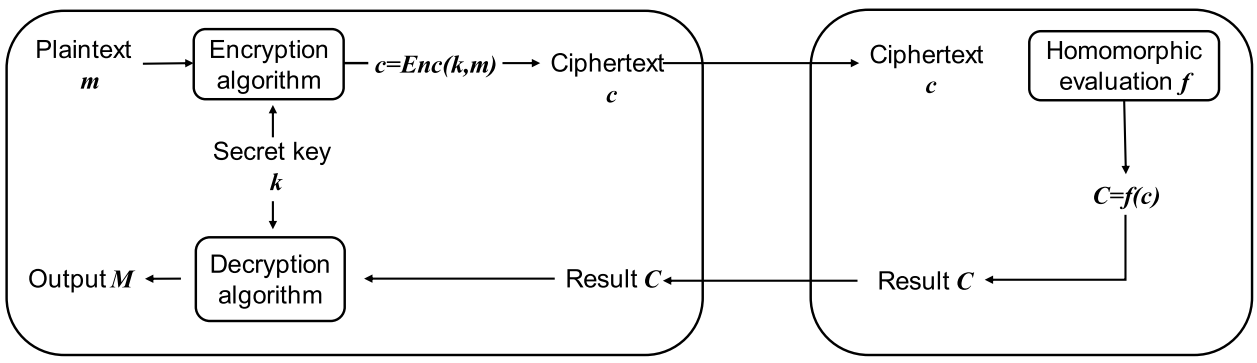

Fig. 1 Privacy homomorphic encryption

Table 1 PMHE symbol definition

Symbol	*Definition Description*
S	Server
C	Client
E	Encoding operation
D	Decoding operation
P	Original data
B	Plaintext
M	Encrypted plaintext
Enc	Encryption operation
Dec	Decryption operation
x	Unknown number
T	The intermediary result of homomorphic operation
X	The vector of user physiological feature
x_i	The value of user physiological feature
$N(x_i)$	The name of user physiological feature
F(x)	Basic elementary function (except for positive Exponential power functions)
pk	Public key
Req(pk)	Request public key
m_i	Intermediate modulus in *coefficient_modulus*
h	The number of modules in *coefficient_modulus*
NCH	The maximum number of homomorphic multiplications
NRE	The number of rescaling
ILE	The initial *"level"* of ciphertext

through sensors arranged in body surface, so as to prediction of diseases later.

- Client APP. It is used to receive data from wearable devices, perform preliminary filtering on original data, then encode and encrypt the filtered data, and finally upload the ciphertext to the cloud platform. In addition, it receives the data tuples after homomorphic operations returned from the cloud platform. If the data tuples contain resulting message, it decrypts the results to get the plaintext, and shows the corresponding health status for user. Last but not least, it generates public/private key pairs for secure communication and signature generation, and interacts with the server through the protocol to determine the format of the data to be uploaded.
- Cloud server. The cloud server, which does not need full trust, is responsible for receiving and processing the user's health data and returning the data tuples after homomorphic operations to the client. Model selection server is also deployed on the cloud server. The server determines the data format that the client needs to provide according to model selection algorithm. Besides, because the client does not know the meaning of plaintext results after decoding, the server also needs to send the specific meaning of each part of the plaintext results to the client.

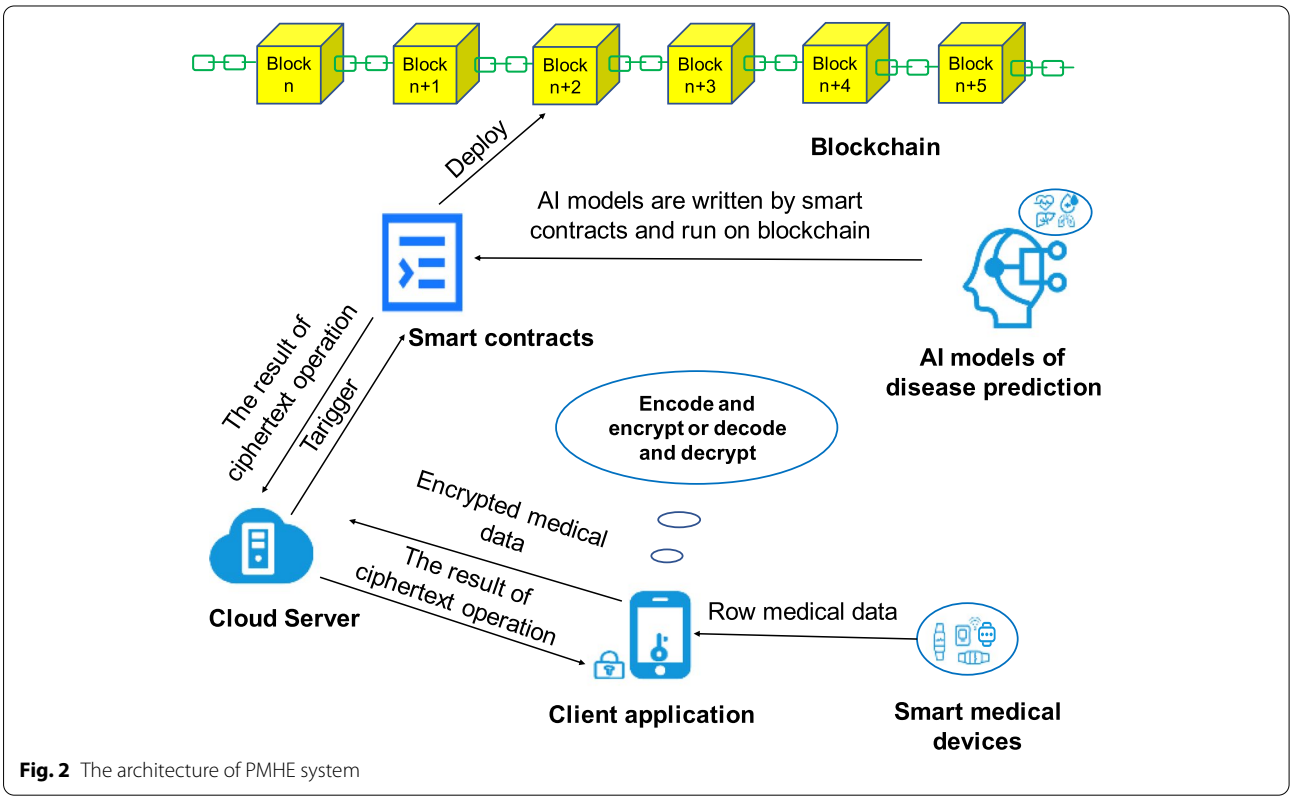

Fig. 2 The architecture of PMHE system

- Disease prediction model. Disease prediction models are written using smart contracts and run on blockchain. The model uses specific AI algorithm to analyze and calculate encrypted health data, so as to realize early warning and health monitoring. Through machine learning algorithm, the disease prediction model can not only be used for the monitoring and early warning of heart disease or hypertension, but also can be widely used for other field such as pregnant women care, the early warning of lung disease and Alzheimer's disease. The AI models of disease prediction in PMHE are based on logistic regression. Logistic regression algorithm has good classification ability [25]. It is a standard method of supervised machine learning to classify data, and it has been successfully used in medical research, for example to help determine whether a patient has a disease [26]. The predictive function of logistic regression is shown in Formula 2.

$$P(X) = \frac{1}{1+\exp\left(-\theta^T X\right)} = \frac{1}{1+\exp\left(-\sum_{i=0}^{n}\theta_i x_i\right)} \tag{2}$$

$X = (1, x_1, \cdots x_n) \in R^{n+1}$, $x_i (1 \le i \le n)$ refers to the user's physiological data in PMHE, which is collected by wearable devices or input by the user on the client, and n is the number of user features required by the model. $\theta = (\theta_0, \theta_1, \cdots, \theta_n) \in R^{n+1}$ refers to the regression coefficient. The model is obtained off-line from the training on unencrypted data set, and CKKS algorithm is introduced into the model to support prediction on encrypted feature data. The predictive function of logistic regression after introducing CKKS algorithm is shown in Formula 3.

$$Enc(P(X)) = \frac{1}{Enc(1)+\exp\left(Enc(-1)Enc\left(\theta^T\right)Enc(X)\right)} = \frac{1}{Enc(1)+\exp\left(Enc(-1)\sum_{i=0}^{n}Enc(\theta_i)Enc(x_i)\right)} \tag{3}$$

The blockchain with smart contracts. Smart medical care involves a large amount of personal health data. Traditional cloud storage uses traditional database systems to manage user information, which is easy to be tampered with and forged by malicious users, posing a great threat to data security. In PMHE, consortium blockchain is used to store user data, and AI prediction algorithms are written using smart contract. Medical data and the program will be stored synchronously at different nodes on the blockchain network, which ensures the failure of a single node will not cause the collapse of the entire system. Smart contracts also ensure the security and reliability of the AI model. Blockchain and smart contracts can avoid data loss, program change, data leakage and other problems, greatly ensure data security.

Security model

In cloud computing system, user data faces two risks about privacy disclosure. One is that off-line data may be eavesdropped during network transmission, while another is that data leakage will happen in the cloud.

- Network eavesdropping. Network eavesdropping is a kind of passive attack. The cloud system communicates with the client via the public link over Internet. Attackers can gain user data by eavesdropping communication channel between the client and the cloud system. The most powerful way to eliminate the threat of eavesdropping is to encrypt off-line data and then transmit ciphertext over the Internet.
- Data leakage. If the cloud server is not fully trusted, it may expose users' personal medical data. Or the cloud system could be trusted, but an attacker could still gain access to user data by hacking into the cloud system. Encryption is the most suitable solution to this security problem. In the cloud system, ciphertext is received and evaluated on AI model. Without a secret key, even if an attacker has access to the data, he or she cannot know the user's privacy.

The PMHE framework can achieve the following security objectives:

- Security. In the process of data up-chain and AI model calculation, the data is in ciphertext form, which ensures the safety of data processing. As for data storage, data is stored using blockchain to ensure data storage security.
- Privacy. Even if the data is eavesdropped during transmission or leaked in the cloud, the attacker can only obtain the ciphertext encrypted with the user's public key. Without knowing the user's private key, the attacker cannot obtain the valuable information contained in the data.

The framework of PMHE

PMHE is a cloud computing framework for smart medical systems based on blockchain and homomorphic encryption technology. The wearable medical device transmits the collected physiological data of the user to the client, and then the client encodes and encrypts the data. The encrypted plaintext is transmitted to the blockchain and triggers a smart contract on the blockchain.

The AI prediction model written using smart contract calculates the ciphertext and predicts the results. The client receives the resulting message after homomorphic operations from the cloud server, and finally gets the predictive plaintext by homomorphic decryption. According to the plaintext information, the client can predict the user's disease or evaluate the user's health status. The program framework of PMHE with CKKS is shown in Fig. 3.

The basic idea of PMHE is to use CKKS encryption algorithm to encrypt plaintext into ciphertext during medical data transmission, calculation, and storage. And the entire process of computing is encrypted to ensure data integrity, privacy and security. Therefore, it is necessary to improve each stage in order to use ciphertext to achieve its function.

Model conversion algorithm

CKKS encryption algorithm only supports addition and multiplication, but AI model involves some functional operations which do not meet homomorphism, such as exponential function, logarithmic function , square root function etc. Although some functions can be converted to addition and multiplication by polynomial fitting (for example, sigmoid can be fitted using polynomial approximation), a lot of ciphertext operations need to be performed in blockchain network. These extra operations will not only result in a loss of efficiency, but will more likely exceed the CKKS multiplication limit.

In PMHE, the AI model is converted into two basic operation units, basic elementary function (except for positive exponential power functions) and polynomial, which are stored in the stack. CKKS can directly execute polynomial operation, and the homomorphic addition and multiplication between operation units can also be supported on CKKS algorithm. The intermediary result outputted by CKKS algorithm is saved to participate in the subsequent homomorphic evaluations. The AI models of disease prediction running on PMHE are based on logistic regression, and the process of prediction function decomposition and CKKS execution is shown in Fig. 4.

Initialization is performed firstly. AI model is converted according to computation sequence and intermediary result T is assigned initial value *Enc(X)*. In the stack, e^x and $\frac{1}{x}$ are basic elementary function, and others are polynomial operation. CKKS algorithm performs the corresponding countermeasure according to the operation unit poped out of stack. When the operation is polynomial operation, smart contract substitutes T into the polynomial and executes homomorphic operations. Then, the temporary result is assigned to T. When the operation is an elementary function, the smart contract sends the function and intermediary result T to the cloud server in the form of data tuples. The cloud server retrieves the mapping table between public key and IP address based on the *pk* in the data tuples and obtain the client ip address. Ultimately, the cloud server sends the data tuples to the corresponding client. The client decrypts the intermediary result and decryption result t is substituted into the basic elementary function. Then the client encrypts the output of function and packages it into the data tuples. In the end, the encrypted output is assigned intermediary result T and participate in homomorphic operations in AI model.

The design of data tuples

In PMHE, the data interacts with each module in the form of tuples. The interaction protocol based on data tuples is described as follows:

1). The server sends the tuple template to the client.

$$P = \left\{N(x_0), N(x_1), \cdots N(x_n), Req(pk)\right\} \tag{4}$$

where $N(x_i)$ indicates the name of user physiological feature, and *Req(pk)* indicates the application for user public key.

2). The client responds to the server, encodes and encrypts the value of physiological features, and sends it to the cloud server with public key.

$$\begin{gathered}(1, \mathrm{x}_1, \mathrm{x}_2, \cdots \mathrm{x}_\mathrm{n}) \xrightarrow{E} \{\mathrm{B}_0, \mathrm{B}_1, \cdots, \mathrm{B}_\mathrm{n}\} \\ \xrightarrow{\mathrm{Enc}} \{\mathrm{M}_0, \mathrm{M}_1, \cdots, \mathrm{M}_\mathrm{n}\} \xrightarrow{\mathrm{add(pk)}} \left\{\mathrm{M}_0, \mathrm{M}_1, \cdots \mathrm{M}_\mathrm{n}, \mathrm{pk}\right\}\end{gathered} \tag{5}$$

$$P \rightarrow M \rightarrow S \tag{6}$$

where $\mathrm{B_x} = \mathrm{E(x)}, \mathrm{B_i} = \mathrm{E(X_i)}, \mathrm{M_i} = \mathrm{Enc(B_i)}, 0 \leq \mathrm{i} \leq \mathrm{n}$.

3). The server receives ciphertext data, creates the *pk-ip* mapping table or inserts the user's pk-ip into it, and uploads the data to the blockchain.

$$S \rightarrow B \tag{7}$$

4). When the off-stack operation is a basic elementary function, the data interaction in PMHE is as follows.

$$\left\{F(x), T, pk\right\} \rightarrow S \overset{pk-ip\,\mathrm{table}}{\rightarrow} C \tag{8}$$

$$D(Dec(T)) = t \tag{9}$$

$$Enc(E(F(t)) = T \tag{10}$$

$$\left\{T, pk\right\} \rightarrow S \rightarrow B \tag{11}$$

PMHE scheme

In PMHE, a data tuple is generated on the client side firstly. And then the data in data tuple is encrypted using

CKKS-PMHE

Client:

Init:

1. Set state: = INIT
2. Select CKKS parameters according AI model
3. Generate PK/SK pair
4. Set ep=0

Register:

1. Send ("register", $profile, $model, $key, $para). $profile includes the information about the user, model indicates which function does the user want to use. $key is the public key, while $para is the basic parameters of CKKS
2. Set state: = REGISTER.

Submit: Upon receiving ("start", $epoch, $tuple template) from cloud server.

1. Set state: = SUBMITTING
2. Collecting sensor data, computer corresponding operator of each data according $tuple template
3. Encode and encrypt all the data and package them into $tuple
4. Send ("Submit", $tuple) to cloud server until ep<$epoch
5. Set state: = SUBMITTED.

Diagnose: Upon receiving ("report", $result, c) from cloud server.

1. Set state: = DIAGNOSE
2. Decrypt $result to get the result T
3. Decode T to get real result Q
4. Query result criterion c using Q to get report

Cloud Server:

Init:

1. Set state: = INIT
2. Init AI Model collections: = {m_1, m_2, …m_i}
3. Init AI Results Criterion: = {c_1, c_2, …, c_i}, c_i = {<s_i: normal, s_i +1- s_i +delta: Mild Risk, s_i +delta: heavy Risk>}

Register: Upon receiving ("register", $model, $key, $para).

1. Select AI model m from collections according $model and query $epoch from m
2. Storing ($profile, m, $key, $para) locally
3. Send ("start", $epoch) to client c
4. Send ("init", $profile, m, $para, $key) to contract

Delivery: Upon receiving ("submit", $tuple) from client c.

1. Set state: = RECEIVING
2. Receive ("submit", $tuple) from the client until the number of $tuple less than $epoch, store the received data into a list $list
3. Send ("run", $key, $list) to contract
4. Set state: = RUN

Report: Upon receiving ("issue", m, $result) from contract

1. Select result criterion c according m
2. Send ("report", $result, c) to client c

Fig. 3 The program framework of PMHE with CKKS supporting

CKKS homomorphic encryption to hide valuable information. An ideal property of the CKKS scheme is that the ciphertext can be computed directly without prior decryption of the ciphertext. Therefore, privacy security, data authorization distribution and secure transmission of personal data are ensured.

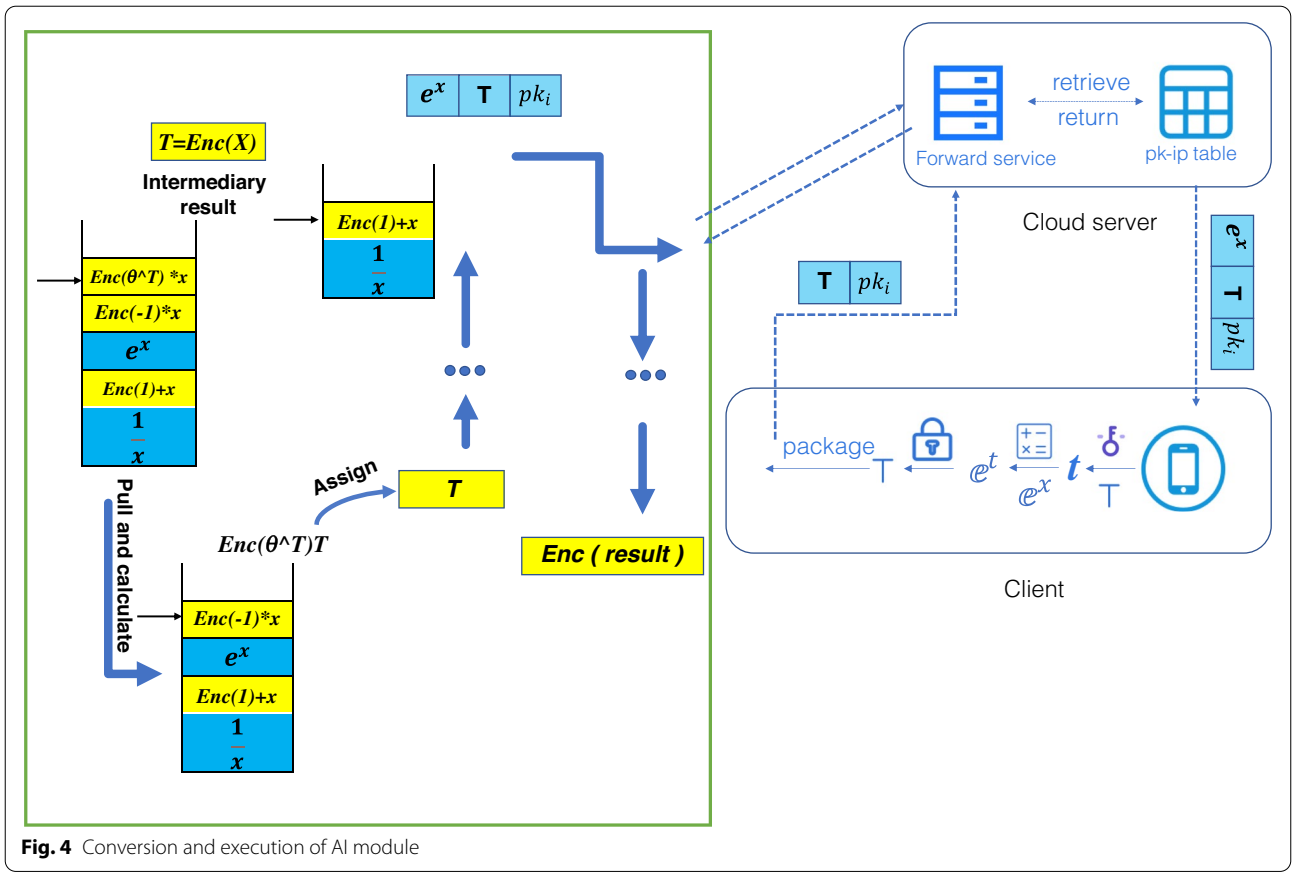

Fig. 4 Conversion and execution of AI module

1) INITIALIZATION. In the initialization phase, the user registers with the client APP, connects the client to the wearable device, and generates the public/private key pair. The public key represents the unique identity and is used to protect data in homomorphic encryption algorithms. The private key is used to sign the submitted data, verify the user's identity, and decrypt the calculation results returned by the cloud.
2) SUBMIT. AI model is deployed on the blockchain and implemented as a smart contract. The client obtains the physiological data collected by the wearable device, encrypts them with the public key after coding, and submits them to the cloud application. The cloud calls the corresponding smart contract, performs the data calculation, and saves the data to the blockchain.
3) PREDICTION. Based on the physiological data collected by wearable devices, ciphertext is calculated in the AI model. After the calculation, the result is obtained. In the calculation process of AI model based on homomorphic encryption technology, the valuable information contained in the data cannot be observed, which ensures the privacy of user data.
4) REPORT. Transforming physiological indicator data collected by wearable devices into disease prediction results with medical value and providing users with low-cost, high-quality and high-precision medical information is the meaning of report. For example, intelligent heart health assessment algorithm can efficiently and accurately predict whether a person has coronary heart disease, and is not affected by the diagnostic ability of the doctor. However, the result in encrypted formation cannot be used for diagnostics, so the server cannot understand it and sends the result to the client.
5) DIAGNOSIS. After the client receives the operation result of the ciphertext, it decrypts and decodes it with the private key to get the unencrypted AI model result, and then compares the model reference criteria to get the diagnostic result.

Smart contracts

Smart contracts can be used to accomplish more complex business logic when more business and application requirements need to be fulfilled in the blockchain.

Essentially, smart contract is pieces of executable code that run in a blockchain, so it has the same decentralized and autonomous characteristics as blockchain. In the PMHE framework, the principle of full homomorphic encryption based on smart contracts is as follows [27]:

1) *setup*(*m*): It refers to the function that selects AI model according to parameter *m*.
2) *receive*(*List*(*tuple*)): It refers to receive a list of encrypted data tuples from the client.
3) *send*(*pk*, *result*): It refers to send the results of the AI model operation on encrypted data to the client.
4) *add(pk, ciphertext1, ciphertext2)*: It refers to the homomorphic function of addition, which can be used by users to perform addition calculations on ciphertext data
5) *mult(pk, ciphertext1, ciphertext2)*: It refers to the homomorphic function of multiplication that users can use to multiply on ciphertext data.

The formal description of the FHE-Contract algorithm is as follows:

- Input: The encrypted data tuples uploaded by the client.
- Output: The result of the homomorphic calculation on the encrypted data.

The specific algorithm description is shown in Fig. 5.

AI models in PMHE

PMHE supports disease prediction on encrypted physiological information. The specific realization and prediction process of disease prediction model will be discussed in the section.

Firstly, PMHE trains unencrypted AI models using traditional machine learning methods on unencrypted public data sets. Specifically, PMHE obtains regression coefficient θ in Formula 2. Then, a smart contract that supports the operation sequence in the operation stack in Fig. 4 will be programmed. In the smart contract, the addition and multiplication operation are the most basic operation, and all of them are implemented using CKKS to support prediction on encrypted physiological data. When client uploads the encrypted feature data *Enc(X)* to the cloud server, the smart contract is triggered. The operations defined in the smart contract are executed sequentially according to the operation order in the stack until the encrypted result *Enc(result)* is finally obtained. The client decrypts *Enc(result)* to obtain the predicted result. Figure 5 illustrates the detailed execution logic of the AI model on the blockchain. During the whole process, the user's physiological data is encrypted, which fundamentally ensures privacy.

Discussion and analysis

In this section, we firstly give a formal security proof of privacy under the universally composable framework, and then analyze PMHE performance from a communication and computing perspective.

FHE-Contract

```
1.  model=FHE.Setup(m)
2.  iresult=receive(data)
3.  while (stack is not null) do
4.    operation=stack.pop()
5.    if operation is multiplication do
6.      iresult=FHE.mult(pk, iresult)
7.    end if
8.    if operation is addition do
9.      iresult=FHE.add(pk, iresult)
10.   end if
11.   if operation is elementary function do
12.     Send(pk, <iresult, func>)
13.     iresult=receive(truples)
14.   end if
15. end while
16. return   iresult
```

Fig. 5 FHE-Contract algorithm

Privacy security

Unlike traditional cloud computing application frameworks, PMHE consists of three components: client, cloud system and blockchain. By using the CKKS algorithm, the system realizes the function of data security protection in communication and calculation. The task of client is to generate keys, select CKKS algorithm parameters. Besides, it is responsible for encoding and encrypting user data. All those operations ensure the security of data transmitted over the Internet, and the user data participating in operations of the AI model is encrypted too. The server sends the calculation result on cryptographic state to the client. The client gets an approximation which is extremely close to the real result. By comparing the result with the reference criteria, we can get a diagnostic report.

In the entire process of data processing, encrypted data is transmitted on the public link between the server and the client, and cloud server and smart contract deal with ciphertext, which ensure the security of data in communication link and model operation. Based on the security of CKKS algorithm, brute force attack on the ciphertext still faces the problem of computational infeasibility, and the user privacy is guaranteed fundamentally. Based on the security of the Hyperledger Fabric, the data on blockchain is difficult to be tampered with, which ensures the security of data storage. In conclusion, user's privacy and medical data are protected furthest.

Performance analysis

Compared with the traditional smart medical cloud platforms, PMHE ensures data security. The data is encoded to meet the CKKS algorithm, which causes an increasing of data size. In the calculation process, each multiplication involves rescaling and re-linearize operation [28], which results in more calculated workload. According to the interaction protocol, the tuple transmitted in PMHE is several times larger than the original data. As a result, PMHE adds additional overhead in terms of communication, computing, and storage.

Suppose there are twenty items in user physiological indicator.

$$X = \{x_0, x_1, \ldots x_{19}\} \tag{12}$$

Storage cost

In PMHE, the cloud server system only records the user's basic information and the model parameters used by the user, but does not record specific user data. Therefore, the storage cost of cloud servers is not high.

User data is stored in blockchain (Hyperledger Fabric's state database). In each uploaded tuple, what we need to store is only twenty pieces of basic user data. Each data consists of 8192 floating-point numbers (equivalent to *polynomial_modulus*). The total storage required is about 1.25 MB, which is quite acceptable.

Computing cost

In PMHE, computing tasks are assigned to the client device and the blockchain. Encoding, encryption, decryption and decoding of user data are completed on the client, while the AI model runs on smart contract of blockchain. During client initialization, the operation of generating key pair causes extra computation time. In addition, each sensor data needs to be encoded as plaintext (encoded message). The length of plaintext is polynomial modulus times longer than original data. After encrypting, the length of encrypted plaintext is twice as long as plaintext. Compared with traditional clients in cloud computing system, each data-upload costs more computing time, but the interval of one minute for uploading data can be ignored.

The AI model performs homomorphic operations on encrypted data [21] on blockchain network, and most of these operations are multiplication. Since the operators are polynomial, and the operations such as rescaling and relinearize are required for each multiplication, the computational complexity is greatly increased. However, blockchain itself is a distributed system, made up of many nodes, which allows computing to take place without centralization. The structural characteristics of blockchain can effectively relieve the computational stress.

Communication cost

At communication stage, the first interaction between the medical device and the cloud requires setting interaction parameters for each other, and the main cost in the rest time is uploading encrypted medical data. Wearable devices usually upload data at a frequency of from 1 time per minute to 1 time per 10 minutes, depending on the user's physical condition. Even at the highest frequency, we assume that the data tuple contains twenty physiological indicators. The system need upload approximately 1.25MB per minute, and the network speed requirement is about 175kbps, which is generally acceptable in the WiFi environment or 4G environment.

Experiment and evaluation

The PMHE prototype has been implemented on Hyperledger Fabric and conducted comprehensive experiments to evaluate its feasibility and performance in storage and computation.

Experiment setting and parameters selection

Experiment setting

Latigo-ckks [29] is a CKKS framework implemented with GO language, which is also the preferred language for smart contract on Hyperledger Fabric development.

In the experiment, the *DELL R720* was used as the server and 8 virtual machines were built with virtualization technology. One is used as a Web application server to manage user information, receive user data, and invoke smart contracts on blockchain. The others are used to build a Hyperledger Fabric environment.

Hardware server configuration:

- CPU: E5-2697 v3 @ 2.60 GHz
- Memory: 16*16GB 1600-MHz DDR3

Virtual Machine configuration:

- OS: CentOS 8.0
- CPU: 4 vCPUs
- Memory: 16G

The client test was conducted on a mobile phone running the Android operating system.

The configurations of mobile phone are as follows:

- OS: Android 11
- CPU: Dimensity 1110 2.6GHZ
- Memory: 8 GB

The implementation of the PMHE scheme consists of five parts: portable wearable medical device, client, cloud server, blockchain and smart contract which implements disease prediction model. The computing results of AI model are used to predict or diagnose the health status of users. Because the model implemented with CKKS executes approximate arithmetic, there is an error between the output of decryption algorithm and the real results. Therefore, the focus of the experiment is the computational efficiency of the client, the execution efficiency of the smart contract and the accuracy of AI model.

CKKS is a public key encryption system, which has all the characteristics of public key encryption system, such as public key encryption, private key decryption, etc. Therefore, the following components are needed in the program:

- Keygenerator: Generating the key
- Encryptor: Encrypting data with a public key
- Decryptor: Decrypting ciphertext with a private key
- Evaluator: Executing homomorphic evaluations

According to the design of PMHE, the first three modules are realized in the client, and the evaluator module is the AI model essentially.

Parameters selection

CKKS requires three preset parameters: poly_modulus_degree, coefficient_modulus, and scale.

Parameter *poly_modulus_degree* must be a number of powers of 2, such as 1024, 2048, 4096, 8192. Larger value supports to perform more complex calculations, but will increase the size of the ciphertext. The number of moduli in *coeff_modules* determines the number of homomorphic multiplications that can be performed. The last item in *coeff_modules* is the special modules, whose value should be equal to the maximum value of the intermediate modules, and the intermediate modules should be as close to *scale* as possible. The relationship between the *max coeff_modulus bit-length* and *poly_modules degree* is shown in Table 2 [29].

Parameter *scale* can scale floating point numbers, and executing multiplication on encrypted data causes *scale* to double. To control the inflation of *scale*, the operation of rescaling needs to be performed [17]. Each rescaling operation consumes one intermediate modules m_i.The reduction of *scale* is about 2^{m_i},because the intermediate modules is close to *scale*, the *scale* remains stable, which ensures final accuracy. Before each computation, we should ensure that the data participating in the homomorphic evaluation is on the same *"level"*. The *"level"* of data can only be lowered rather than raised and is reduced by one after the operation of rescaling, which means the number of homomorphic multiplications is finite, and the order of calculation in model design is very important. When *"level"* falls to zero, multiplication cannot be performed. The relationship between *h*, *NCH*, *NRE* and *ILE* is shown in Formula 13.

$$NRE = ILE = log(NCH + 1) = h - 2 \tag{13}$$

Different disease prediction models have different requirements on computation complexity, computation time and computation security. Therefore, two groups of encryption parameters are selected according to the

Table 2 Relationship between parameters

poly_modulus_degree	*max coeff_ modulus bit-length*
1024	27
2048	54
4096	109
8192	218

above selection principal and actual needs. According to Formula 13, the user can choose parameters1 or parameters2 when *NCH* in AI model does not exceed 3. When a certain item in the model requires *NCH* more than 3, only parameters2 can be selected. *NCH* does a preliminary filtering of the available parameters. And when *NCH* cannot determine a unique parameter, the user can choose the parameter as he or she wishes.

Parameters1:

$$poly\_module\_degree = 8196;$$

$$coeff\_modulus = \{60, 40, 40, 60\}; scale = 2^{40}$$

Parameters2:

$$poly\_module\_degree = 8196;$$

$$coeff\_modulus = \{50, 30, 30, 30, 50\}; scale = 2^{30}$$

Since the AI model running on PMHE requires no more than twenty user physiological features in practical application, twenty items are set in the data tuples in test.

Table 3 Computing time of every stage on client

Procedure	*Average Time(Microsecond)*	
	Parameters1	**Parameters2**
Generating the private key	3696	3724
Generating the public key	8845	14,666
Generating encryptor	412	468
Generating encoder	1810	2208
Generating decoder	1972	3243
Encoding	6187	9778
Encryption	8499	12,062
Decryption	14,172	18,759
Decoding	495	571

PMHE time consumption testing

Client testing

The size of public and private key file, the length of plaintexts after encoding and the length of encrypted plaintexts are highly related to the *poly_modulus_degree* and the *coefficientt_modulus*. The larger the *poly_modulus_degree,* the higher the security is, but the time of key generation, encoding and encryption will increase. In the case of parameters1 and parameters2, thirty experiments were conducted respectively, and the average time consumption of each operation is obtained. The time statistics for clients to generate key, encoding, and encryption are shown in Table 3.

In client, the private key, public key, encryptor, encoder and decoder can be reused. The client encodes and encrypts the tuple before uploading it. Table 3 shows that encoding and encrypting a data tuple containing 20 user characteristics take less than 0.03 s. And it takes no more than 0.02 s for the user to decode and decrypt the result. Client operations do not add much time consumption.

AI model testing

In PMHE framework, the AI model is implemented by smart contract, the operations on smart contract contain only addition and multiplication. Each item in polynomial is the multiplication of user data ciphertext and polynomial coefficients ciphertext. When the ciphertext is multiplied, the size of the ciphertext increases dramatically and the computation time increases accordingly. In order to reduce the size of the ciphertext and obtain faster computing speed, relinearize operation is required. Meanwhile, after every multiplication, rescaling operation is required to control the growth of the *scale* and reduce the error. The AI model running time statistics are shown in Table 4.

Table 4 Encryption time statistics

Procedure	*Average Time(Microsecond)*	
	Parameters1	**Parameters2**
Generating the secret key of relinearize operation	121,565	200,155
Single addition operation	10,792	10,987
Multiplication operation between ciphertexts	34,941	49,108
Multiplication operation between ciphertext and plaintext	11,288	12,203
Square operation of ciphertext	34,339	48,247
The time of relinearize and rescaling operation	121,497	187,206
Total	334,422	507,906
The number of tests	30	30

Error size analysis of CKKS algorithm

The core function of the smart medical system is disease prediction and health monitoring, and finally diagnosis is made according to the output of decryption algorithm. CKKS algorithm based on approximate arithmetic, and noise is introduced into the operations. As a result, there is an error between final result and real value. To figure out the error size, we divided the original data into three groups, one-digit, double-digit, and three-digit. Because the number of multiplications in the AI model exceeds three times, parameters1 supports a maximum of three multiplications and does not meet the calculation requirements, we select parameters2. Each experiment was repeated fifty times. The experimental results are shown in Fig. 6.

Figure 6 shows that the error between the final results and the real results is very small, and the maximum error is no more than 0.02%. In the worst case, the cumulative calculation error of the user's twenty encrypted features is less than 0.4%. Since logistic regression is a binary model, if the distance between the output of the prediction function and the threshold value exceeds 0.4%, the error has no effect on the prediction result. CKKS algorithm can fully meet the needs of disease prediction and health monitoring.

AI model accuracy testing

In PMHE, users can select different disease prediction models through model selection algorithm in cloud server. In order to comprehensively evaluate the accuracy of the models, we chose five different data sets, which are Low Birth Weight Study(LBW) [30], Heart_UCI [31], Umaru Impact Study(UIS) [32], Breast-Cancer-Wisconsin(BCW) [33] and NhanesIII [34]. The specific data sets information is shown in Table 5. We selected 75% of the data set for training to obtain the initial AI model, and introduced the CKKS algorithm into the initial AI model to obtain CKKS-AI model. The accuracy of initial AI model and CKKS-AI model is shown in Fig. 7. Besides, we compared the scheme in this paper with that of Lui K [15], and the result is shown in Fig. 8.

Experimental results show that the accuracy of CKKS-AI model is close to that of the initial AI model, for reasons described in Section "Error size analysis of CKKS algorithm". And CKKS-AI model has higher accuracy compared with the model trained in encrypted data set.

Table 5 The information of data sets

Dataset	Number of observations	Number of features
Low Birth Weight Study	189	9
Heart_UCI	303	14
Umaru Impact Study	575	8
Breast-Cancer-Wisconsin	699	10
NhanesIII	15,649	15

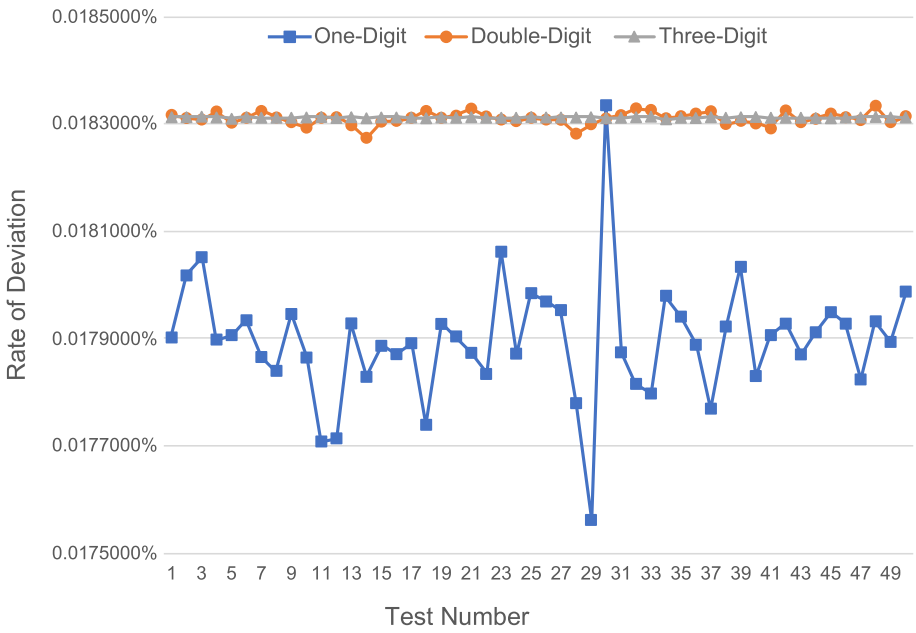

Fig. 6 Variable error rate of different digits

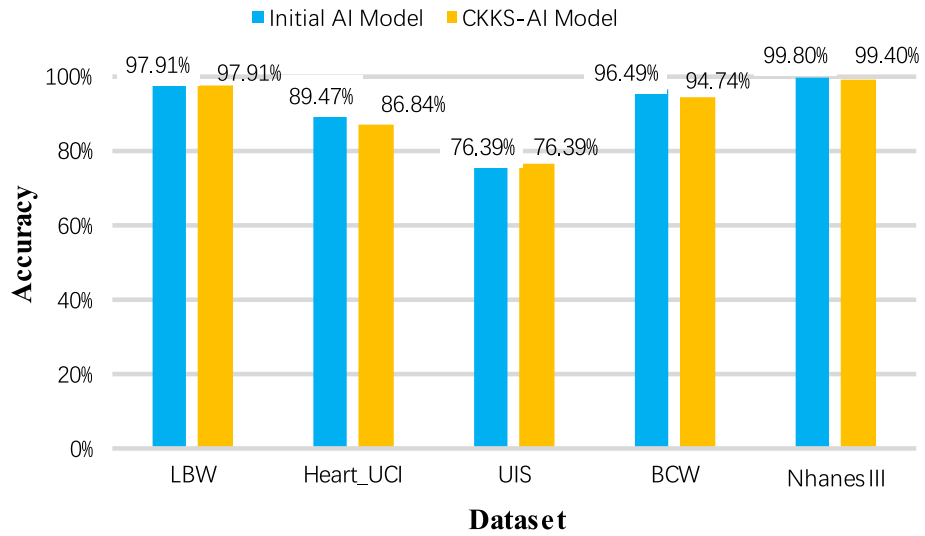

Fig. 7 The accuracy of initial AI model and CKKS-AI model

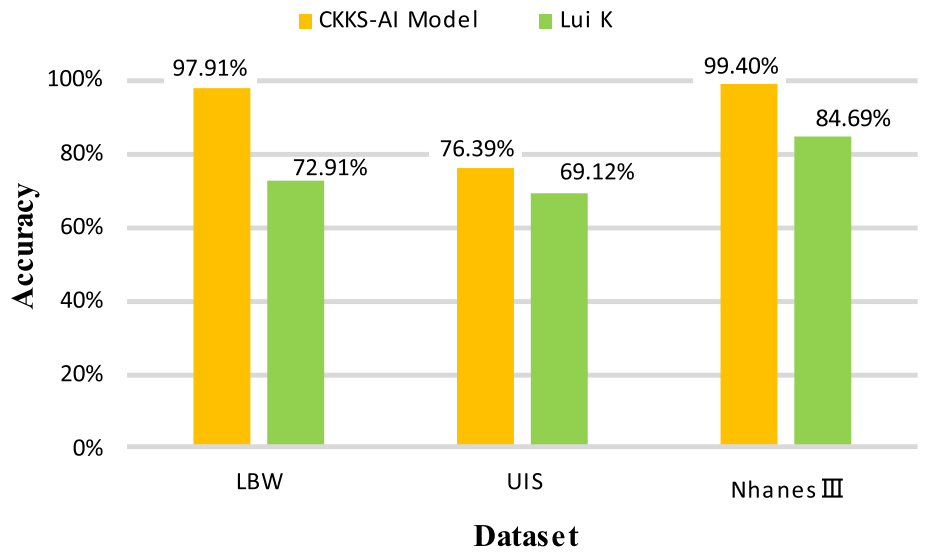

Fig. 8 Comparison of two schemes

Summary and conclusion

Medical and health data are of great value both in the scientific research and medical field, such as clinical auxiliary diagnosis and health management, but they face the risk of privacy disclosure [27]. This paper first introduces the research background and significance of medical and health data privacy preserving, and then illustrates the development of blockchain technology in the privacy preserving of medical data by investigating the related work world widely. Then it introduces blockchain smart contract technology and homomorphic encryption algorithm. On this basis, aiming at the privacy disclosure problems faced by medical data in the disease prediction model, this paper introduces homomorphic encryption technology to propose blockchain-based privacy protection method of medical and health data. Based on this method, we design and implement a privacy preserving framework of smart medical data based on blockchain and homomorphic encryption, named PMHE. In the entire process of data processing, no matter network transmission, model calculation, or data storage, involved data in this procedure is all ciphertext encrypted with public key. On the premise of not affecting the accuracy of calculation, PMHE realizes the security of the data

on the chain, protecting the privacy of the user. In other words, PMHE truly achieves that the data is available but uncollectable.

In summary, the proposed solution can be used in following fields:

- The ciphertext results generated by the disease prediction model can be used in other health big data industries, such as health monitoring, nursing homes and medical institutions. Compared with traditional big data analysis, this greatly protects users' privacy.
- Encrypted messages can be used to exchange relevant data with healthcare providers, and ultimately provide high-quality, low-cost and safe solutions for smart medical products. The disease pattern databases built by data exchanged can be used as health big data, provide reference for disease diagnosis and contribute to social health.

PMHE is an exploration of using blockchain and privacy computing for smart medical application, there are still some problems to be further solved in the future:

- The prototype experiment results of the scheme proposed in this paper are satisfactory, but its efficiency needs to be further verified when it is used in large-scale applications with high real-time requirements.
- Because the AI model in this paper mainly operates on multiplication and addition, it is easy to implement with CKKS. However, the calculation model of many problems is more complex, and the degree of fitting polynomials is too high to exceed the *"level"* limit of CKKS. Therefore, the next step is not only to optimize the model design, but also to optimize the CKKS algorithm itself to improve its universality.

Acknowledgements
Thanks to Dr. Xiang Zhang of University of Nottingham Ningbo China for her help in our work.

Authors' contributions
J.Z., W.W., D.W. and C.M. developed the idea, protocol design, and wrote the original draft. J.Z and D.W. improved the scheme, and provided useful advice, W.W performed the experiment, and X.W improved the AI model. All authors have read and agreed to the published version of the manuscript.

Funding
None

Availability of data and materials
The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 29 November 2021 Accepted: 21 November 2022
Published online: 17 December 2022

References

1. Zhang Y, Qiu M, Tsai C, Hassan M, Alamri A (2017) Health-CPS: healthcare cyber-physical system assisted by cloud and big data. IEEE Syst J 11(1):88–95
2. Zhang X, Zhang J, Hang C, Tang W (2021) Verifiable statistical analysis scheme for encrypted medical data in cloud storage. Comput. Eng 47(6):32–37
3. She W, Chen J, Liu Q, Hu Y, Gu Z, Tian Z, Liu W (2019) New blockchain technology for medical big data security sharing. J Chin Comp Syst 40(7):1449–1454
4. Sovova O (2019) Electronization in health care and privacy protection. J Sustain Dev 9(23):72–80
5. Price W, Cohen I (2019) Privacy in the age of medical big data. Nat Med 25(1):37–43. https://doi.org/10.1038/s41591-018-0272-7
6. Wang S (2020) Research on medical data privacy protection application based on homomorphic encryption. Dissertation, Shenyang Aerospace University.
7. Huang J, Jiang Y, Li Z, Fan L (2018) Application prospect of blockchain in medical industry. J Med Inf 39(2):1–8. https://doi.org/10.3969/j.issn.1673-6036.2018.02.001
8. Niu G (2020) Research and implementation of trusted sharing platform for medical data based on blockchain. Dissertation, Harbin Institute of Technology.
9. Hylock R, Zeng X (2019) A blockchain framework for patient-centered health records and exchange (HealthChain): evaluation and proof-of-concept study. J Med Internet Res 21(8):e13592. https://doi.org/10.2196/13592
10. Wang R, Yu S, Li Y, Tang Y, Zhang F (2019) Medical blockchain of privacy data sharing model based on ring Signature. J Univ Electron Sci Technol China 48(6):886–892
11. Liu J, Liang T, Sun R, Du X, Guizani M (2020) A privacy-preserving medical data sharing scheme based on consortium blockchain. In: GLOBECOM 2020–2020 IEEE Conference and Exhibition on Global Telecommunications. IEEE, Taipei.
12. Zhou Z, Chen Y, Li T, Ren X, Qing X (2021) Medical data security sharing scheme based on consortium blockchain. J Appl 39(1):123–134. https://doi.org/10.3969/j.issn.0255-8297.2021.01.011
13. Crawford J, Gentry C, Halevi S, Platt D, Shoup V (2018) Doing Real Work with FHE: The Case of Logistic Regression. In: Proceedings of the 6th Workshop on Encrypted Computing & Applied Homomorphic Cryptography. https://doi.org/10.1145/3267973.3267974
14. Kim M, Song Y, Wang S, Xia Y, Jiang X (2018) Secure logistic regression based on homomorphic encryption: design and evaluation. JMIR Med Inform 6(2):1–11. https://doi.org/10.2196/medinform.8805
15. Lui K (2020) Research of Privacy Preservation In Logistic Regression Based on Homomorphic Encryption. Dissertation, Dalian University of Technology.
16. Nakamoto S(2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bit-coin.pdf.
17. OuYang L, Wang S, Yuan Y, Ni X, Wang F (2019) Smart contracts: architecture and research progresses. Acta Autom Sin 45(3):445–457. https://doi.org/10.16383/j.aas.c180586
18. Rivest RL, Adleman L, Dertouzos ML (1978) On data banks and privacy homomorphisms. Found Secure Comput 4(11):169–180
19. Rivest RL, Shamir A, Adleman L (1978) A method for obtaining digital signatures and public-key cryptosystems. Assoc Comput Mach 21(2):120–126
20. Gentry C (2009) Fully homomorphic encryption using ideal lattices. In: Proc. 41st ACM Symp. Theory Comput. (STOC), pp. 169–178. https://doi.org/10.1145/1536414.1536440

21. Cheon J. H, Kim A, Kim M, Song Y (2017) Homomorphic encryption for arithmetic of approximate numbers. In: International Conference on the Theory and Application of Cryptology and Information Security, pp. 409–437.
22. Liu Y, Xia Q, Li Z, Xia H, Zhang X, Gao J (2020) Research on secure data sharing system based on blockchain. Big Data Res 6(5):92–105
23. Zhao C, Zhao S, Zhao M, Chen Z, Gao C, Li H, Tan Y (2019) Secure multi-party computation: theory, practice and applications, "Inf. Sci 476:357–372
24. Zheng S, Liu X, Zhou T, Yang X (2021) Optimized CKKS scheme based on learning with errors problem. J Comp Appl 41(6):1723–1728. https://doi.org/10.11772/j.issn.1001-9081.2020091447
25. Anindita L, Isnandar S, Sugiyanto (2020) Comparison of heart disease classification with logistic regression algorithm and random forest algorithm. In: international conference on science and applied science. https://doi.org/10.1063/5.0030579
26. Aono Y, Hayashi T, Wang L, Phong L (2016) Scalable and Secure Logistic Regression via Homomorphic Encryption. In: Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy. https://doi.org/10.1145/2857705.2857731
27. Chen J (2020) Research on privacy protection method of medical and health data based on blockchain. Dissertation, Zhengzhou university.
28. Xu H, Yang X, Zhang X (2021) Protection of face feature information based on fully homomorphic encryption in cloud computing environment. Guizhou Daxue Xuebao (Ziran Kexueban) 38(3):83–91. https://doi.org/10.15958/j.cnki.gdxbzrb.2021.03.12
29. Lattigo v2.2.0. http://github.com/ldsec/lattigo.
30. Low Birth Weight Study Data Set [OL]. 2017. https://rdrr.io/rforge/LogisticDx/man/lbw.html.
31. https://archive.ics.uci.edu/ml/datasets/Heart+Disease.
32. Umaru Impact Study Data Set [OL]. 2017. https://rdrr.io/rforge/LogisticDx/man/uis.html.
33. https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin.
34. Nhanes III Data Set [OL]. 2017. https://rdrr.io/rforge/LogisticDx/man/nhanes3.html.

Publisher's Note