sustainability

Article

A Traceable Online Insurance Claims System Based on Blockchain and Smart Contract Technology

Chin-Ling Chen [1,2,3], Yong-Yuan Deng [3,*], Woei-Jiunn Tsaur [4,*], Chun-Ta Li [5,*], Cheng-Chi Lee [6,7,*] and Chih-Ming Wu [8]

[1] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361005, China; clc@mail.cyut.edu.tw
[2] School of Information Engineering, Changchun Sci-Tech University, Changchun 130600, China
[3] Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan
[4] Computer Center, National Taipei University, New Taipei City 237303, Taiwan
[5] Department of Information Management, Tainan University of Technology, Tainan City 71002, Taiwan
[6] Department of Library and Information Science, Research and Development Center for Physical Education, Health, and Information Technology, Fu Jen Catholic University, New Taipei City 242062, Taiwan
[7] Department of Computer Science and Information Engineering, Asia University, Wufeng Shiang, Taichung 41349, Taiwan
[8] School of Civil Engineering and Architecture, Xiamen University of Technology, Xiamen 361024, China; chihmingwu@xmut.edu.cn
* Correspondence: allendeng@cyut.edu.tw (Y.-Y.D.); wjtsaur@mail.ntpu.edu.tw (W.-J.T.); th0040@mail.tut.edu.tw (C.-T.L.); cclee@blue.lins.fju.edu.tw (C.-C.L.)

Abstract: In the current medical insurance claims process, there are problems of low efficiency and complex services. When a patient applies for medical insurance claims, he/she must go to the hospital to apply for a diagnosis certificate and receipt and then send the relevant application documents to the insurance company. The patient will not receive compensation until the company completes the verification with the patient's hospital. However, we can improve the current dilemma through blockchain technology. Blockchain technology can effectively open up the information channels of the insurance industry and medical institutions, promote industry integration, and enhance the ability of insurance companies to obtain information. In this research, we used blockchain and smart contract technology to make the following contributions to the development of Internet insurance. First, blockchain and smart contract technology can effectively solve the problem of online underwriting. Second, it is conducive to improving supervision. Third, it is conducive to solving risk control problems. Fourth, it is conducive to effective anti-money laundering. The proposed scheme fulfills the following security requirements: mutual authentication of identities, non-repudiation between each of two roles, and other major blockchain-based security requirements. In the event of a dispute, we also proposed an arbitration mechanism to divide responsibilities.

Keywords: blockchain; electronic medical record (EMR); insurance claim; mutual authentication; non-repudiation; smart contract

check for updates

Citation: Chen, C.-L.; Deng, Y.-Y.; Tsaur, W.-J.; Li, C.-T.; Lee, C.-C.; Wu, C.-M. A Traceable Online Insurance Claims System Based on Blockchain and Smart Contract Technology. *Sustainability* **2021**, *13*, 9386. https://doi.org/10.3390/su13169386

Academic Editor: Marko Hölbl

Received: 30 July 2021
Accepted: 19 August 2021
Published: 21 August 2021

1. Introduction

1.1. Background

According to the calculations of the International Association of Insurance Supervisors (IAIS) [1], about ~20–30% of insurance claims are suspected of fraud. A small number of policyholders' lack of integrity and weak legal awareness often lead to insurance fraud. The policyholder's information cannot be shared. Even if some policyholders are blacklisted by an insurance company, it cannot prevent other insurance companies from dealing with that dishonest insurance company. The policyholder signs the contract, which is caused by the asymmetry of the information of the insurance parties and the lack of information transparency. On the other hand, the current insurance industry has complicated underwriting procedures and a high cost of underwriting time. The insurance market has poor supervision and poor supervision efficiency. Thus, it is impossible to effectively review the risks of insurance applicants and prevent moral hazards. It is also unable to effectively solve the current problem of money laundering by criminals using Internet insurance.

In the field of medical insurance, insurers, medical institutions, and insurance providers form a triangular relationship. In every interaction, there are problems of low efficiency and complex services. For insurance service providers, insurance costs are high, especially management costs. A significant amount of energy is spent on contract signing and management, maintenance of the database, payment, and collection of funds, claim inspection, data review, etc. Take life insurance as an example. Nowadays, when a patient applies for medical insurance claims services, the patient must go to the hospital to apply for a paper diagnosis certificate and receipt and then send the relevant application documents to the insurance company through the insurance business personnel or in person, waiting thereafter for the insurance. The patient will not receive compensation until the company completes the verification with the patient's hospital. According to research, the current medical insurance claims procedures generally suffer from complicated procedures and endless compensation time [2,3].

Moreover, if the family's insurance history is not clear, the applicant must apply to each insurance company to check the insurance contents one by one to sort out all the insurance rights and interests. For policyholders, most policyholders and their families are full of unknowns and fears when facing medical bills and third-party reimbursement processes. The complexity of insurance reimbursement makes the process lengthy, and policyholders also have many questions that are difficult to resolve. Secondly, for medical institutions, a considerable amount of time is spent each year in the insurance reimbursement process, collation of medical records, insurance service providers and government audits, etc. Claim payment and adjudication is a very complicated process, involving significant management costs and manual processes to verify whether all stakeholders meet and comply with the agreed conditions in the contract [4].

Blockchain, as a technology for collectively maintaining a reliable database through decentralization and trustlessness, has developed rapidly in recent years and is gradually being used in various fields. The characteristics of blockchain technology, such as decentralization, trustlessness, non-tampering, transparency, and traceability, are suitable for promoting the government's structural adjustment, governance, and service transparency, thereby promoting the construction of intelligence and trust. At the same time, the application of blockchain to government work coincides with the concept of various "Internet +" applications proposed by governments of various countries.

Blockchain is a decentralized information storage system based on decentralized storage and P2P networks, comprehensively using cryptography and consensus algorithms to achieve decentralization [5–13]. The advantages of blockchain technology are the decentralization, anonymity, immutability, and traceability of stored information. As an industry with operating risks, insurance relies on data to survive. It requires a large amount of complete and true information in the design of insurance, underwriting, and claim settlement. Blockchain technology can effectively unblock the information channels within the insurance industry and related industries, promote industry integration, and enhance the ability of insurance companies to obtain information. With the maturity of blockchain technology, it is possible to construct a "blockchain + insurance ecological chain".

An electronic medical record (EMR) is a technology of electronic information to store medical record data for users to query and browse [14–21]. The purpose of implementing electronic medical records is roughly as follows. First, electronic medical records are used for management and quality control: to keep records of the modification and deletion of medical records, to facilitate the control of medical procedures, and to generate quality indicators and abnormal events records in real-time. Second, electronic medical records are used for sharing information: to facilitate the sharing of medical record data by all units in the hospital and to maintain the uniformity of medical record data. Third, electronic medical records reduce the cost to maintain medical records and regular operations: they reduce the cost of space and time spent on paper medical records. The medical record data are provided with an information system function so that medical staff can immediately obtain medical records without the need to read paper medical records. Fourth, electronic medical records are used for study and research: structured medical records are used instead of simple scans and archives. As we knew, automatic claims' settlement in the insurance industry has a high degree of relevance to the research issue of EMR.

In the past, Chen et al. [22,23] proposed an EMR-sharing system to reduce the waste of medical resources. However, they did not build a connection with the insurance industry. In 2018, Wang and Song [24] addressed electronic health records (EHRs) by using a consortium blockchain to develop decentralized EHR-sharing methods for insurance companies. Wang and Song's scheme included only conceptual descriptions of the insurance industry; the detailed protocol was not mentioned.

In this study, we used blockchain and smart contract technology to make the following contributions to the development of Internet insurance:

(1) Effectively solve the problem of online underwriting.

When the blockchain technology matures and is used in the financial field, all the past insurance records, credit records, and personal health files of the insured will be recorded on the blockchain one by one, which cannot be changed and cannot be forged. Internet insurance can use blockchain technology to underwrite the health and insurance records of the insured online, to truly realize the program of online underwriting, which effectively reduces the time cost of both offline policyholders and insurers. The underwriting efficiency of internet insurance companies has been greatly improved. At the same time, internet insurance has also realized the mode conversion from offline underwriting to online underwriting. Internet insurance is no longer a method of sales, but a method of operation.

(2) Conducive to improving supervision.

Due to the lack of network supervision, Internet crimes are becoming more and more popular. In the future, blockchain technology can solve this problem from a technical level, especially for the operation and supervision of Internet insurance. One of the characteristics of blockchain technology is time-stamping and non-editable modification. This fundamentally guarantees the authenticity of the data on the chain. Then, the use of blockchain technology in Internet insurance can fundamentally improve the problem of network supervision. On the one hand, due to the timestamp characteristics of the blockchain and the complete record of the data and transaction records from the current block to the genesis block, the legality of the operation of Internet insurance companies can be guaranteed from the perspective of government supervision. On the other hand, it can also effectively improve the current weak supervision in the insurance market and improve the efficiency of supervision.

(3) Conducive to solving risk control problems.

The first aim is to control moral hazard. Blockchain technology can effectively reduce the probability of moral hazard. Since the blockchain builds a complete trustless value network, the credit history, health status, property registration, and use status of the insured person will be formed on the blockchain during the process of Internet insurance from underwriting to claim settlement. Data are recorded with timestamps, and once the insured is in danger, the insurer can effectively review the insured's risk by querying all the records of the insured on the blockchain, thereby effectively preventing moral hazard.

The second aim is to control technical risks. The current computer technology of Internet insurance companies is not mature enough, and problems such as system crashes and hacking attacks are prone to occur. Blockchain technology is a decentralized ledger structure, and decentralized nodes can strongly protect the security of the Internet insurance system. Through collective maintenance, the probability of technical risks is minimized.

The last aim is to control information security risks. Insurers, especially Internet insurance companies' customer information, company internal information, etc., are particularly important. The high security of blockchain technology ensures the confidentiality of Internet insurance companies' information, and the system is not easily prone to malicious tampering and destruction. To a certain extent, it guarantees the normal operation of Internet insurance companies and reduces the cost of system maintenance.

(4) Conducive to effective anti-money laundering.

Internet insurance uses blockchain technology, which can prevent money laundering effectively. Every transaction record of funds has a clear time stamp, and the data loaded on the blockchain must obtain recognition for most of the blocks on the blockchain, and this guarantees the credibility and security of the data to a certain extent. At the same time, transaction records cannot be arbitrarily modified, and the data and records on the blockchain cannot be arbitrarily destroyed; so, every fund can be traced back to the two parties and the time of the transaction, which is a very effective solution to the issue of current criminals using Internet insurance for money laundering. It also provides a more powerful tool for the supervision of the capital chain.

On the whole, blockchain technology provides the following features in the field of insurance applications:

✓ Blockchain smart contracts can improve the efficiency of claims settlement to a certain extent.
✓ The security of the blockchain can solve the privacy and security issues of the policyholders to a certain extent.
✓ Blockchain can solve the trust problem between policyholders and insurance institutions to a certain extent.
✓ Blockchain can solve the problem of traceability in sales links and prevent data tampering to a certain extent.

In response to the above-mentioned characteristics of blockchain, applying it to this research can enable both the insurer and the insured to obtain a sustainable development relationship. For insurers, applying the blockchain mechanism can make their operations and management more efficient and can accurately calculate operational risks. For the insured, choosing an insurance company that provides a blockchain claims mechanism can simplify the procedures for claim settlement, allowing the amount of compensation to be quickly obtained and increasing the insured's trust in the insurance company. In short, this research can allow the overall insurance chain to have a positive development on sustainability.

If a patient is treated in a hospital through blockchain technology, the medical institution will record the blockchain information of the patient's electronic medical record on the chain and cooperate with the insurance company. When the insured is diagnosed with a certain disease in the medical institution, through the smart contract [25], when the preset conditions of the system are met, such as when a medical insurance accident occurs, the automatic performance function of the smart contract will be activated, and the claim payment will be activated. It will be automatically transferred directly to the beneficiary's account to achieve the effect of automatic claim settlement, to accelerate the efficiency of claim settlement, and to save related costs.

Aiming at the problem of inefficient claims settlement, the smart contract of blockchain technology does provide a good solution. The basic idea of smart contracts is to embed contract terms in hardware and software so that the contract cannot be destroyed or the cost of a breach of contract is extremely high. Smart contracts have the advantages of certainty, consistency, termination, verifiability, efficiency, immediacy, and low cost. When the information written into the blockchain meets the claim conditions, Internet insurance will automatically enter the claim settlement program. After the claim payment is automatically credited to the designated account of the insured, the insurance claim contract will be automatically completed. This kind of automated claim settlement using blockchain technology simplifies the payment procedure, reduces many links of manual review, increases the speed of contract execution, and reduces the cost of verification and execution of the contract.

1.2. Research Goals

This research used blockchain and smart contract technology to propose a traceable online insurance claims system to achieve the following research goals [26–38]:

(1) Mutual authentication: the proposed scheme must be able to verify the legitimacy of the identities of the sender and receiver. This study intended to use the BAN logic proof to determine whether the two parties have achieved mutual authentication.

(2) Resist man-in-the-middle attacks: attackers can intercept messages during transmission and send illegal messages as users. The proposed scheme uses the ECDSA system to protect messages. Therefore, when attackers perform a man-in-the-middle attack, they cannot correctly intercept important information.

(3) Verifiable: in the proposed scheme, the sender's information stream will be signed, and the receiver can verify whether the signature is true.

(4) Integrity and unforgery: to ensure the integrity of the message during the communication process, the proposed scheme uses a signature mechanism to ensure that the message is not tampered with.

(5) Traceable: due to equal information, blockchain technology guarantees that all participants have equal rights to know and choose. The health of the insured, the insurance record, and the types of insurance services provided by the insurance company are all auditable. All participants can obtain all the transaction information. Because of the decentralized storage verification feature, all change records will be synchronized on the chain. When any dispute occurs in the future, there will be data to prove to protect the rights of policyholders.

(6) Openness: with the setting of the public key and the private key, except that the private information of the transaction subject is encrypted, everyone can query the blockchain data and develop related applications through the public interface. The system information is open and transparent, reducing information asymmetry and thereby solving the problems of moral hazard and adverse selection between the supply and demand of insurance. With the help of openness, the use of big data and cloud computing can be improved, and the development and pricing of insurance products can be more accurate.

(7) Privacy: behind the alliance chain combined with blockchain technology, only the data that the policyholder agrees to share will be stored on the blockchain. Blockchain technology can not only allow authorized persons to access data through signature private keys, encryption technology, and secure multi-party computing technology but also ensure that the core data and privacy of the blockchain alliance member database are not leaked. To protect the privacy of users, the content of the insurance contract is restricted to access. Only the party can view the personal contract, and the key is in the hands of the party. The contract review, query, modification, and other information will occur and record in the block, and the insurance contract is fully implemented automatically through smart contracts.

(8) Decentralization and information sharing: accounting and storage is decentralized; that is, all nodes have the same rights and obligations, and any error or shutdown of any node will not affect the operation of the entire network.

(9) Non-repudiation: once the data is verified and added to the blockchain, it will be permanently stored, and the inherent time stamp function of the blockchain can record the creation time. Information changes need to control more than 51% of the system nodes, which will be very difficult under an open system.

The organization of the rest of the article is as follows. Section 2 provides the preliminary. Section 3 presents the proposed protocol. In Section 4, we analyze the security issue of our proposed traceable online insurance claims system. In Section 5, we discuss the computation cost and the communication cost and make a comparison with existing medical-related decentralized blockchain surveys. Finally, Section 6 concludes the article.

2. Preliminary

2.1. Blockchain and Smart Contract

A block refers to a data packet that records all transactions in a period. Here, the definition of a transaction is broad, ranging from trading commodities, assets, willingness to deliver results, and progress. Once the parties involved in the transaction agree with the final result, the transaction records generated in the process will be packaged and sealed, and no changes are allowed from then on. All blocks are connected in chronological order and extend endlessly. The blockchain is a mechanism of "national paparazzi" that strengthens the function of supervision. The technology of blockchain advertises the three characteristic goals of "decentralization", "openness", and "transparency". Blockchain technology includes hash algorithm, digital signature, timestamp technology, workload proof mechanism, etc. These technologies can ensure the source, time, and subjects of the data while encrypting the blockchain data. Each node in the blockchain system does not need to master the technical details of these computer information security fields but only needs to understand the specific system operation specifications. At the same time, it is open and transparent. The combination of information security technology and economic management is the greatest innovation in the field of human credit.

The concept of smart contracts was first proposed by Nick Szabo [39,40] in 1996. A smart contract is a computer agreement that spreads, verifies, or executes a contract in an information-based way. Smart contracts allow credible transactions without third-party vendors, which are traceable and irreversible. Smart contracts can support automated claims; it is a transparent and reliable payment mechanism and can be used to enforce contract-specific rules. When an insurance event occurs and the insurance payment conditions are met, the smart contract will automatically execute the code instructions, automatically start the insurance claims program, and realize automatic appropriation and payment. A large number of manual operations in the traditional insurance compensation path have been reduced, and the efficiency of compensation has been substantially improved, which has helped insurance companies reduce a large number of operating expenses.

2.2. ECDSA

The elliptic curve digital signature algorithm (ECDSA) was proposed by Vanstone [41] in 1992, which is a derived type of the digital signature algorithm (DSA) that uses elliptic curve cryptography (ECC). The characteristics of ECC make the ECDSA require a significantly smaller key size with the same level of security, which offers faster computations and less storage space. The security of the elliptic curve digital signature algorithm (ECDSA) is based on the discrete logarithm problem. The proof of the ECDSA refers to ref. [42].

Next, we briefly describe the three phases of ECDSA key generation for verification.

For the key generation phase, we assumed that any participant must apply to our blockchain center for public and private keys and the key generation with ECDSA is as follows:

$$Q_x = d_x G \tag{1}$$

where x is the participant ID, Q_x is the public key, d_x is the private key, and G is a generating point based on the elliptic curve. The public key Q_x and the private key d_x are sent to the participant and store. Q_x will also be stored in the blockchain center.

During the signature phase, there is a message that needs to be sent by participant x to y. Participant x chooses a random number k between 1 and $\mathrm{n} - 1$ and calculates a point on the curve as follows:

$$(x, y) = k \times G \tag{2}$$

Then they calculate:

$$r_x = x \bmod n \tag{3}$$

Next, x signs a message m as follows:

$$sig_x = k^{-1}(h(m) + r_x d_x) \tag{4}$$

and sends (m, r_x, sig_x) to y.

During the verification phase, when y receives (m, r_x, sig_x), then the parameters are calculated as follows:

$$(x\prime, y\prime) = (h(m)sig_x{}^{-1}\text{mod}n)G + (r_x sig_x{}^{-1}\text{mod}n)Q_x \tag{5}$$

Then, the following is verified:

$$x' \overset{?}{=} r_x\text{mod}n \tag{6}$$

to determine if the signature is valid.

2.3. BAN Logic

Burrows–Abadi–Needham (BAN) logic is a set of rules for defining and analyzing information exchange protocols. Burrows et al. [43] proposed BAN logic, a defining logic for the analysis of security protocols, in 1990. BAN logic is a logic of beliefs, and the purpose of BAN logic is to analyze authentication protocols by deriving the beliefs [44]. Therefore, BAN logic can be used for validating mutual authentication with the communication protocols.

2.4. Hyperledger Fabric

Hyperledger Fabric [45], proposed by the Linux Foundation in 2015, is a blockchain-based distributed ledger solution platform, which controls transactions through chain codes. Based on a modular architecture, it can provide a high degree of confidentiality, flexibility, and scalability. The chain code in Hyperledger Fabric encapsulates the business logic used to create and modify the ledger, which can be written in different programming languages (such as Java, Go, and Node.js) [46]. The chain code is created and executed by peers to promote, authenticate, and implement the rules for reading. The business logic of the chain code is defined by mutual agreement between members and is used to read, execute, and update the current state of the ledger. When the condition is triggered, the chain code will perform a specific task, and the transaction execution result will be submitted to the blockchain network and will finally be attached to the copy of the ledger of all peers [47].

3. Method

3.1. System Architecture

The solution proposed by this research was to realize an automatic medical insurance claims service system through blockchain technology. As shown in Figure 1, the environment is used to share information among medical institutions, insurance companies, and patients. The roles in the environment include the blockchain center (BCC), the competent authorities (CA), the medical institution (MI), the insurance company (IC), the bank (BK), the patient (PT), and the arbitration institution (AI). Among them, medical institutions can form a medical alliance chain, which is supervised by the medical authority CA1. Insurance and financial institutions can form a financial alliance chain, which is supervised by the financial authority CA2. Members in the same alliance can share complete information content.

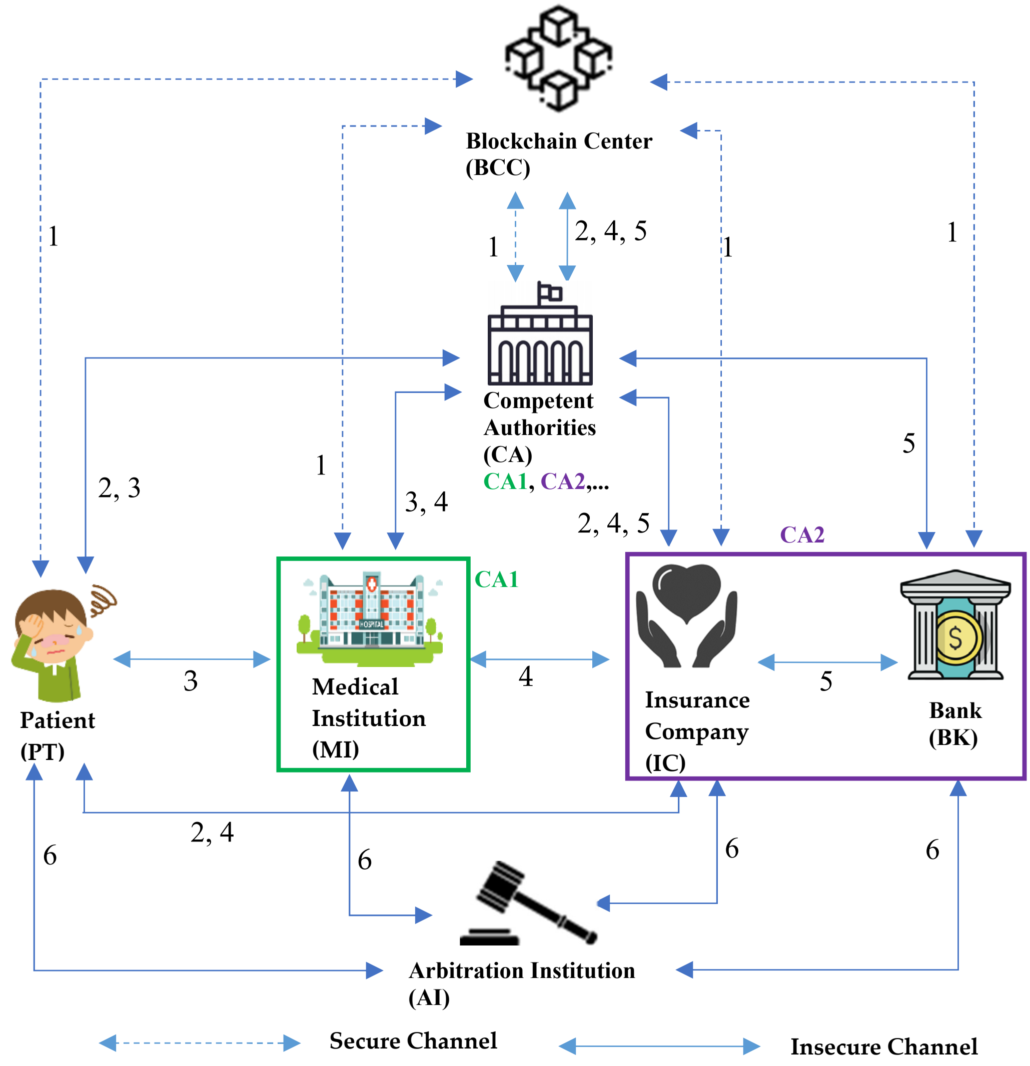


Figure 1. System architecture diagram.

Step 1: All CA, MI, IC, BK, and PT need to be registered with BCC to obtain public and private keys for ECDSA signature and public and private keys for PKI encryption. BCC also stores all patient medical blockchain information. In addition, various CA of different natures will form different alliances with their members, and the information of the alliance members will be shared. For example, CA1 is a medical alliance, and its members are MI, while CA2 is a financial alliance, and its members are IC and BK.

Step 2: The patient PT purchases medical insurance from the insurance company IC. The IC will first verify the identity of the PT and sign an insurance contract with the PT. The PT needs to provide the IC with its BK account, and the record will be transferred to the BCC through the CA. When the PT visits the medical institution MI in the future, if the diagnosis result meets the claimed content specified in the insurance contract, the IC will proceed with the insurance claim.

Step 3: When the patient PT visits a medical institution MI and informs MI that he/she has purchased medical insurance, the MI will first verify the identity of the PT, read the electronic medical record EMR of the PT, and then make a diagnosis, and the records will be transmitted to the BCC through CA.

Step 4: The medical institution MI informs the insurance company IC to carry out insurance claims, and the IC obtains the PT medical-related diagnosis content provided by MI.

If the claims are eligible, the IC will inform the PT of the claim amount and payment time, and the record will be sent to the BCC through the CA.

Step 5: The insurance company IC informs the bank BK to pay the patient PT, and the record is transmitted to the BCC through the CA.

Step 6: In the event of a claims dispute, the patient PT can appeal to the arbitration institution AI. AI will obtain the message content of each party and make reasonable judgments.

3.2. Notation

The notation of the proposed scheme is as follows:

q	A k-bit prime number
$GF(q)$	Finite group q
E	The elliptic curve defined on finite group q
G	A generating point based on the elliptic curve E
ID_x	A name representing identity x
k_x	A random value on the elliptic curve
(r_x, s_x)	Elliptic curve signature value of x
$M_{x\text{-}y}$	A message from x to y
ID_{BC}	An index value of blockchain message
BC_x	Blockchain message of x
TS_x	Timestamp message of x
Enc_x	Encrypted message using the asymmetric key of x
$Cert_x$	The certificate of role x
BK_x	The bank account of role x
EMR_x	The electronic medical record of role x
IC_{PT}	The insurance contract of the patient
IC_{TID}	A transaction number that is changed every round
$h(.)$	Hash function
$A \overset{?}{=} B$	Verify whether A is equal to B

3.3. Smart Contract Initialization

Blockchain technology was applied in the proposed architecture. Some key information is saved and verified through the blockchain, during the authentication and authorization process. The key information in the blockchain is defined in the smart contract. Table 1 below is the blockchain smart contract structure for the proposed scheme.

We developed key information that is stored in the blockchain in the proposed smart contract. The basic fields of id (identification), transaction detail, certificate, and timestamp are presented in each smart contract. The bank account of the patient is added in the pticinf/icbkinf smart contract, while the bank account of the insurance company is added in the bkicinf smart contract. The field insurance contract is provided in the icptinf/ptmiinf/miicinf smart contract. The electronic medical record is offered in the miptinf/miicinf smart contract. At last, the transaction id is shown in the icmiinf/icbkinf/bkicinf smart contract. The blockchain center also issues the public and private key pairs for all roles in the registration phase.

Table 1. Smart contract structure of the proposed scheme.

struct smart contract apcainf/caapinf {
string ap/ca id;
string ap/ca detail;
string ap/ca cert;
string ap/ca tsp;
}
struct smart contract pticinf {
string pt id;
string pt detail;
string pt cert;
string pt bkpt;
string pt tsp;
}
struct smart contract icptinf {
string ic id;
string ic detail;
string ic cert;
string ic icpt;
string ic tsp;
}
struct smart contract ptmiinf {
string pt id;
string pt detail;
string pt cert;
string pt icpt;
string pt tsp;
}
struct smart contract miptinf {
string mi id;
string mi detail;
string mi cert;
string mi emrpt;
string mi tsp;
}
	struct smart contract miicinf {
	string mi id;
	string mi detail;
	string mi cert;
	string mi emrpt;
	string mi icpt;
	string mi tsp;
	}
	struct smart contract icmiinf {
	string ic id;
	string ic detail;
	string ic cert;
	string ic ictid;
	string ic tsp;
	}
	struct smart contract icbkinf {
	string ic id;
	string ic detail;
	string ic cert;
	string ic ictid;
	string ic bkpt;
	string ic tsp;
	}
	struct smart contract bkicinf {
	string bk id;
	string bk detail;
	string bk cert;
	string bk ictid;
	string bk bkic;
	string bk tsp;
	}
	string keypairs;
	string count;

3.4. Registration Phase

The system role X can represent the competent authorities (CA), the medical institution (MI), the insurance company (IC), the bank (BK), and the patient (PT), which registers with the blockchain center (BCC), and obtain a relative public/private key pair and a digital certificate of identity from the blockchain center via a secure channel. Figure 2 shows the flowchart of the registration phase.

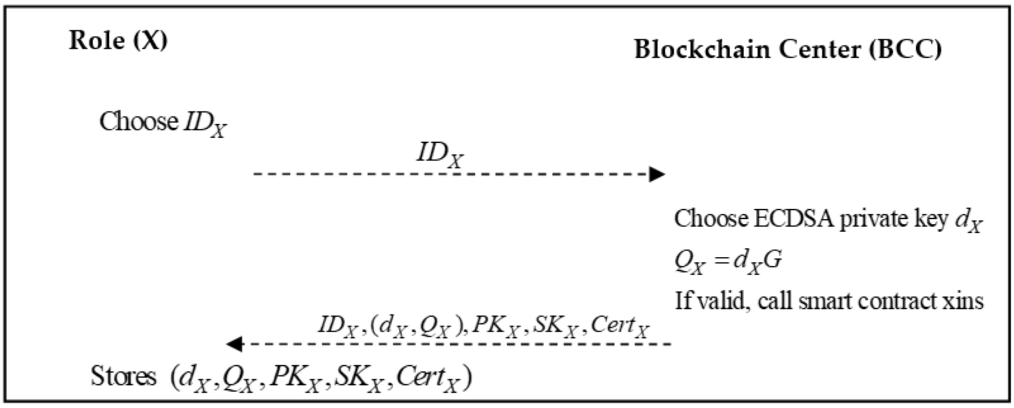

Role (X)
Blockchain Center (BCC)
Choose ID_X
ID_X
Choose ECDSA private key d_X
$Q_X = d_X G$
If valid, call smart contract xins
$ID_X, (d_X, Q_X), PK_X, SK_X, Cert_X$
Stores $(d_X, Q_X, PK_X, SK_X, Cert_X)$

Figure 2. Each role of the system registers with the blockchain center.

Step 1: Role X generates an identity ID_X and sends it to the blockchain center.

Step 2: The blockchain center generates an ECDSA private key d_X based on the role X and calculates:

$$Q_X = d_X G \tag{7}$$

If the identity of the registered role is verified, the smart contract xins will be triggered. Algorithm 1 is presented as follows:

Algorithm 1. Smart contract xins of the proposed scheme.

```
function insert x smart contract xins (
string x_id, string x_detail) {
    count ++;
    x[count].id = id;
    x[count].detail = detail;
}
string x_keypairs;
```

Then, the blockchain center will transmit $ID_X, (d_X, Q_X), PK_X, SK_X, Cert_X$ to role X.

Step 3: The role X stores $(d_X, Q_X, PK_X, SK_X, Cert_X)$.

3.5. ECDSA Authentication Process

In the proposed scheme, system role A and role B must first verify the legitimacy of each other's identities through the ECDSA mechanism at the beginning of the communication. The system role A or role B can represent the competent authorities (CA), the medical institution (MI), the insurance company (IC), the bank (BK), and the patient (PT). We present the flowchart of the ECDSA authentication process in Figure 3. Algorithm 2 shows the ECDSA sign process and Algorithm 3 shows the ECDSA verification process.

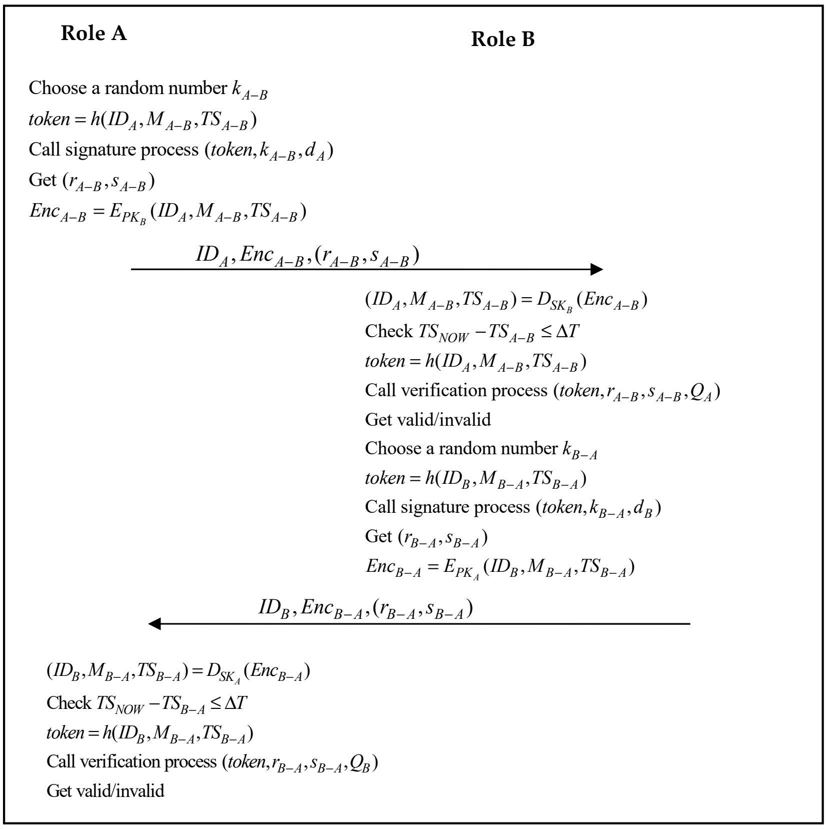

Role A
Role B
Choose a random number k_{A-B}
$token = h(ID_A, M_{A-B}, TS_{A-B})$
Call signature process $(token, k_{A-B}, d_A)$
Get (r_{A-B}, s_{A-B})
$Enc_{A-B} = E_{PK_B}(ID_A, M_{A-B}, TS_{A-B})$
$ID_A, Enc_{A-B}, (r_{A-B}, s_{A-B})$
$(ID_A, M_{A-B}, TS_{A-B}) = D_{SK_B}(Enc_{A-B})$
Check $TS_{NOW} - TS_{A-B} \le \Delta T$
$token = h(ID_A, M_{A-B}, TS_{A-B})$
Call verification process $(token, r_{A-B}, s_{A-B}, Q_A)$
Get valid/invalid
Choose a random number k_{B-A}
$token = h(ID_B, M_{B-A}, TS_{B-A})$
Call signature process $(token, k_{B-A}, d_B)$
Get (r_{B-A}, s_{B-A})
$Enc_{B-A} = E_{PK_A}(ID_B, M_{B-A}, TS_{B-A})$
$ID_B, Enc_{B-A}, (r_{B-A}, s_{B-A})$
$(ID_B, M_{B-A}, TS_{B-A}) = D_{SK_A}(Enc_{B-A})$
Check $TS_{NOW} - TS_{B-A} \le \Delta T$
$token = h(ID_B, M_{B-A}, TS_{B-A})$
Call verification process $(token, r_{B-A}, s_{B-A}, Q_B)$
Get valid/invalid

Figure 3. ECDSA authentication process.

Step 1: Role A generates a random value k_{A-B}, calculates

$$token = h(ID_A, M_{A-B}, TS_{A-B}) \tag{8}$$

calls signature process as Algorithm 2 by $(token, k_{A-B}, d_A)$, obtains (r_{A-B}, s_{A-B}), calculates

$$Enc_{A-B} = E_{PK_B}(ID_A, M_{A-B}, TS_{A-B}) \tag{9}$$

and sends $ID_A, Enc_{A-B}, (r_{A-B}, s_{A-B})$ to role B.

Algorithm 2. ECDSA signature process between role A and role B.

```
string token, $k_{A-B}$, $d_A$;
function signature process (token, $k_{A-B}$, $d_A$)
    $z_{A-B}$ = token;
    ($x_{A-B}$, $y_{A-B}$) = $k_{A-B}G$;
    $r_{A-B} = x_{A-B}$ mod n;
    $s_{A-B} = {k_{A-B}}^{-1}(z_{A-B} + r_{A-B}d_A)$ mod n;
return ($r_{A-B}$, $s_{A-B}$);
```

Step 2: Role B first calculates

$$(ID_A, M_{A-B}, TS_{A-B}) = D_{SK_B}(Enc_{A-B}) \tag{10}$$

uses $TS_{NOW} - TS_{A-B} \leq \Delta T$ to confirm whether the timestamp is valid, and then verifies the correctness of the ECDSA signature. Next, role B calculates

$$token = h(ID_A, M_{A-B}, TS_{A-B}) \tag{11}$$

calls verification process as Algorithm 3 by $(token, r_{A-B}, s_{A-B}, Q_A)$, and obtains valid/invalid.

Algorithm 3. ECDSA verification process between role A and role B.

```
string token, $r_{A-B}$, $s_{A-B}$, $Q_A$;
function verification process ($r_{A-B}$, $s_{A-B}$, $Q_A$)
    ${z_{A-B}}'$ = token;
    $u_{A-B1} = {z_{A-B}}'{s_{A-B}}^{-1}$ mod n;
    $u_{A-B2} = r_{A-B}{s_{A-B}}^{-1}$ mod n;
    (${x_{A-B}}'$, ${y_{A-B}}'$) = $u_{A-B1}G + u_{A-B2}Q_A$;
    check if ${x_{A-B}}' = r_{A-B}$ mod n;
    return valid/invalid;
```

Role B generates a random value k_{B-A} and calculates

$$token = h(ID_B, M_{B-A}, TS_{B-A}) \tag{12}$$

calls signature process by $(token, k_{B-A}, d_B)$, obtains (r_{B-A}, s_{B-A}), calculates

$$Enc_{B-A} = E_{PK_A}(ID_B, M_{B-A}, TS_{B-A}) \tag{13}$$

and sends $ID_B, Enc_{B-A}, (r_{B-A}, s_{B-A})$ to role A.

Step 3: Role A first calculates

$$(ID_B, M_{B-A}, TS_{B-A}) = D_{SK_A}(Enc_{B-A}) \tag{14}$$

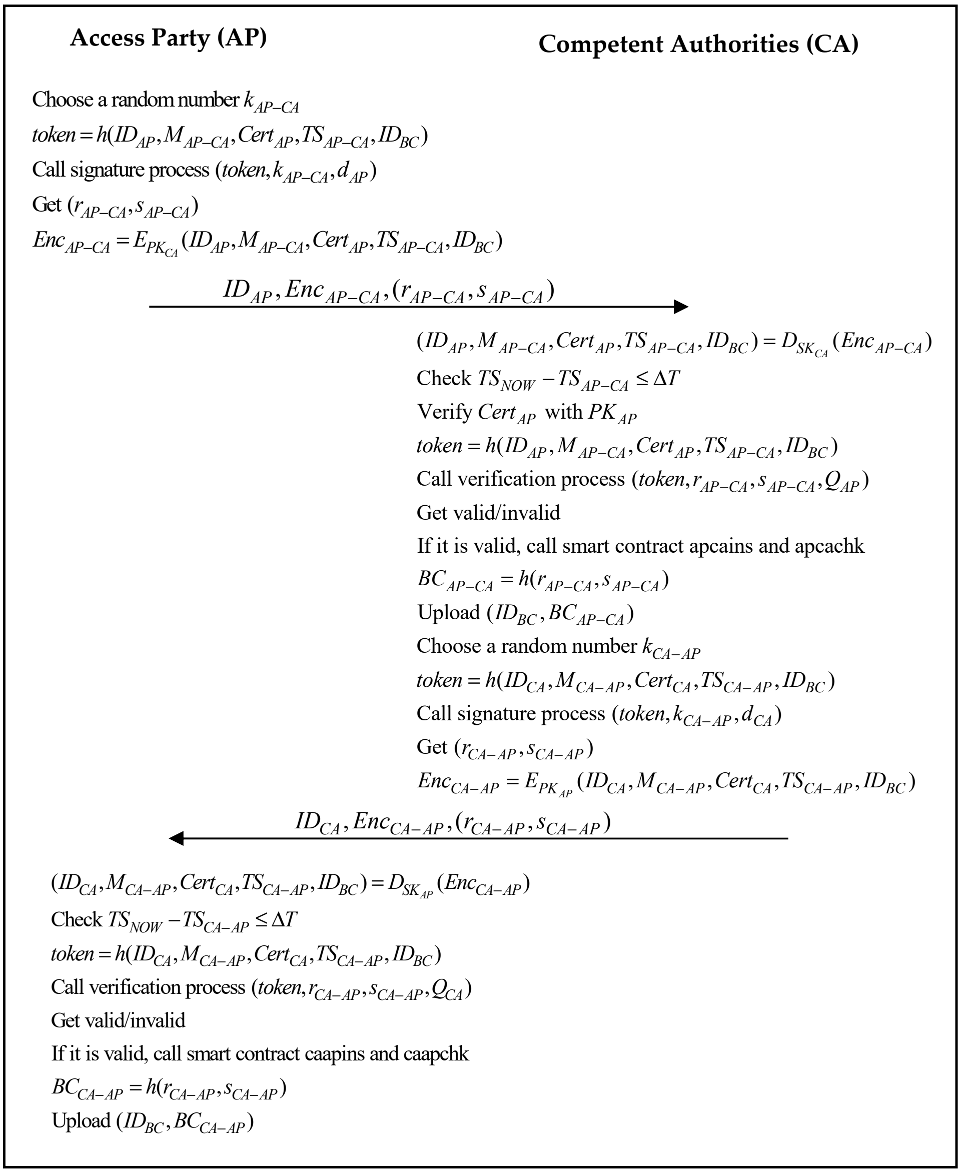
uses $TS_{NOW} - TS_{B-A} \leq \Delta T$ to confirm whether the timestamp is valid, and then verifies the correctness of the ECDSA signature. Next, role A calculates

$$token = h(ID_B, M_{B-A}, TS_{B-A}) \tag{15}$$

calls verification process by $(token, r_{B-A}, s_{B-A}, Q_B)$, and obtains valid/invalid.

3.6. CA Communication Process

In our proposed method, the blockchain architecture of the hyper ledger is used, which increases the role of CA, allows for more flexible access control, and reduces the burden of BCC. After verifying communication between each role, the data will be sent to their respective CA, and then each CA will transmit the blockchain data to the BCC. For example, MI and IC have their own CA, which can ensure the data sharing of each CA member, as well as cross-CA access control, taking into account privacy and efficiency. The access party (AP) can represent the medical institution (MI), the insurance company (IC), the bank (BK), and the patient (PT). We present the flowchart of the CA communication process in Figure 4.



Access Party (AP)

Competent Authorities (CA)

Choose a random number k_{AP-CA}

$token = h(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC})$

Call signature process $(token, k_{AP-CA}, d_{AP})$

Get (r_{AP-CA}, s_{AP-CA})

$Enc_{AP-CA} = E_{PK_{CA}}(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC})$

$ID_{AP}, Enc_{AP-CA}, (r_{AP-CA}, s_{AP-CA})$

$(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC}) = D_{SK_{CA}}(Enc_{AP-CA})$

Check $TS_{NOW} - TS_{AP-CA} \leq \Delta T$

Verify $Cert_{AP}$ with PK_{AP}

$token = h(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC})$

Call verification process $(token, r_{AP-CA}, s_{AP-CA}, Q_{AP})$

Get valid/invalid

If it is valid, call smart contract apcains and apcachk

$BC_{AP-CA} = h(r_{AP-CA}, s_{AP-CA})$

Upload (ID_{BC}, BC_{AP-CA})

Choose a random number k_{CA-AP}

$token = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC})$

Call signature process $(token, k_{CA-AP}, d_{CA})$

Get (r_{CA-AP}, s_{CA-AP})

$Enc_{CA-AP} = E_{PK_{AP}}(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC})$

$ID_{CA}, Enc_{CA-AP}, (r_{CA-AP}, s_{CA-AP})$

$(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC}) = D_{SK_{AP}}(Enc_{CA-AP})$

Check $TS_{NOW} - TS_{CA-AP} \leq \Delta T$

$token = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC})$

Call verification process $(token, r_{CA-AP}, s_{CA-AP}, Q_{CA})$

Get valid/invalid

If it is valid, call smart contract caapins and caapchk

$BC_{CA-AP} = h(r_{CA-AP}, s_{CA-AP})$

Upload (ID_{BC}, BC_{CA-AP})

Figure 4. CA communication process.

Step 1: AP generates a random value k_{AP-CA}, calculates

$$token = h(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC}) \tag{16}$$

calls signature process by $(token, k_{AP-CA}, d_{AP})$, obtains (r_{AP-CA}, s_{AP-CA}), calculates

$$Enc_{AP-CA} = E_{PK_{CA}}(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC}) \tag{17}$$

and sends $ID_{AP}, Enc_{AP-CA}, (r_{AP-CA}, s_{AP-CA})$ to CA.

Step 2: CA first calculates

$$(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC}) = D_{SK_{CA}}(Enc_{AP-CA}) \tag{18}$$

uses $TS_{NOW} - TS_{AP-CA} \leq \Delta T$ to confirm whether the timestamp is valid, verifies $Cert_{AP}$ with PK_{AP}, and then verifies the correctness of the ECDSA signature. Next, CA calculates

$$token = h(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC}) \tag{19}$$

calls the verification process by $(token, r_{AP-CA}, s_{AP-CA}, Q_{AP})$, and obtains valid/invalid. If the verification is passed, CA will obtain the message from AP and trigger the smart contracts apcains and apcachk. Algorithm 4 is as follows:

Algorithm 4. Smart contract apcains and apcachk of the proposed scheme.

```
function insert smart contract apcains (string ap_id, string ap_detail, string ap_cert, string ap_tsp)
{
    count ++;
    ap[count].id = id;
    ap[count].detail = detail;
    ap[count].cert = cert;
    ap[count].tsp = tsp;
}
sign string ap_key (ap_id, ap_detail, ap_cert, ap_tsp);
verify string ap_key (ap_id, ap_detail, ap_cert, ap_tsp);
function check smart contract apcachk (string ap_id, string ap_detail, string ap_cert, string
ap_tsp) {
    return ap_id.exist;
    return ap_detail.exist;
    return ap_cert.exist;
    return ap_tsp.exist;
}
```

CA calculates

$$BC_{AP-CA} = h(r_{AP-CA}, s_{AP-CA}) \tag{20}$$

(ID_{BC}, BC_{AP-CA}) will also be uploaded to the blockchain center. Then, CA generates a random value k_{CA-AP}, calculates

$$token = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC}) \tag{21}$$

calls signature process by $(token, k_{CA-AP}, d_{CA})$, obtains (r_{CA-AP}, s_{CA-AP}), calculates

$$Enc_{CA-AP} = E_{PK_{AP}}(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC}) \tag{22}$$

and sends $ID_{CA}, Enc_{CA-AP}, (r_{CA-AP}, s_{CA-AP})$ to AP.

Step 3: AP first calculates

$$(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC}) = D_{SK_{AP}}(Enc_{CA-AP}) \tag{23}$$

uses $TS_{NOW} - TS_{CA-AP} \leq \Delta T$ to confirm whether the timestamp is valid and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC}) \tag{24}$$

calls verification process by $(token, r_{CA-AP}, s_{CA-AP}, Q_{CA})$, and obtains valid/invalid. If the verification is passed, CA will obtain the message from AP, and the smart contracts caapins and caapchk will be sent. Algorithm 5 is as follows:

Algorithm 5. Smart contract caapins and caapchk of the proposed scheme.

```
function insert smart contract caapins (string ca_id, string ca_detail, string ca_cert, string ca_tsp) {
    count ++;
    ca[count].id = id;
    ca[count].detail = detail;
    ca[count].cert = cert;
    ca[count].tsp = tsp;
}
sign string ca_key (ca_id, ca_detail, ca_cert, ca_tsp);
verify string ca_key (ca_id, ca_detail, ca_cert, ca_tsp);
function check smart contract caapchk (string ca_id, string ca_detail, string ca_cert, string ca_tsp) {
    return ca_id.exist;
    return ca_detail.exist;
    return ca_cert.exist;
    return ca_tsp.exist;
}
```

AP calculates

$$BC_{CA-AP} = h(r_{CA-AP}, s_{CA-AP}) \tag{25}$$

(ID_{BC}, BC_{CA-AP}) will also be uploaded to the blockchain center.

3.7. Insurance Purchasing Phase

The patient PT purchases medical insurance from the insurance company IC. The IC will first verify the identity of the PT and sign an insurance contract with the PT. The PT needs to provide its bank account with the IC, and the record will be transferred to the BCC through the financial alliance chain. When the PT visits the medical institution MI in the future, if the diagnosis result meets the claimed content specified in the insurance contract, the IC will proceed with the insurance claim. We present the flowchart of the insurance purchasing phase in Figure 5.

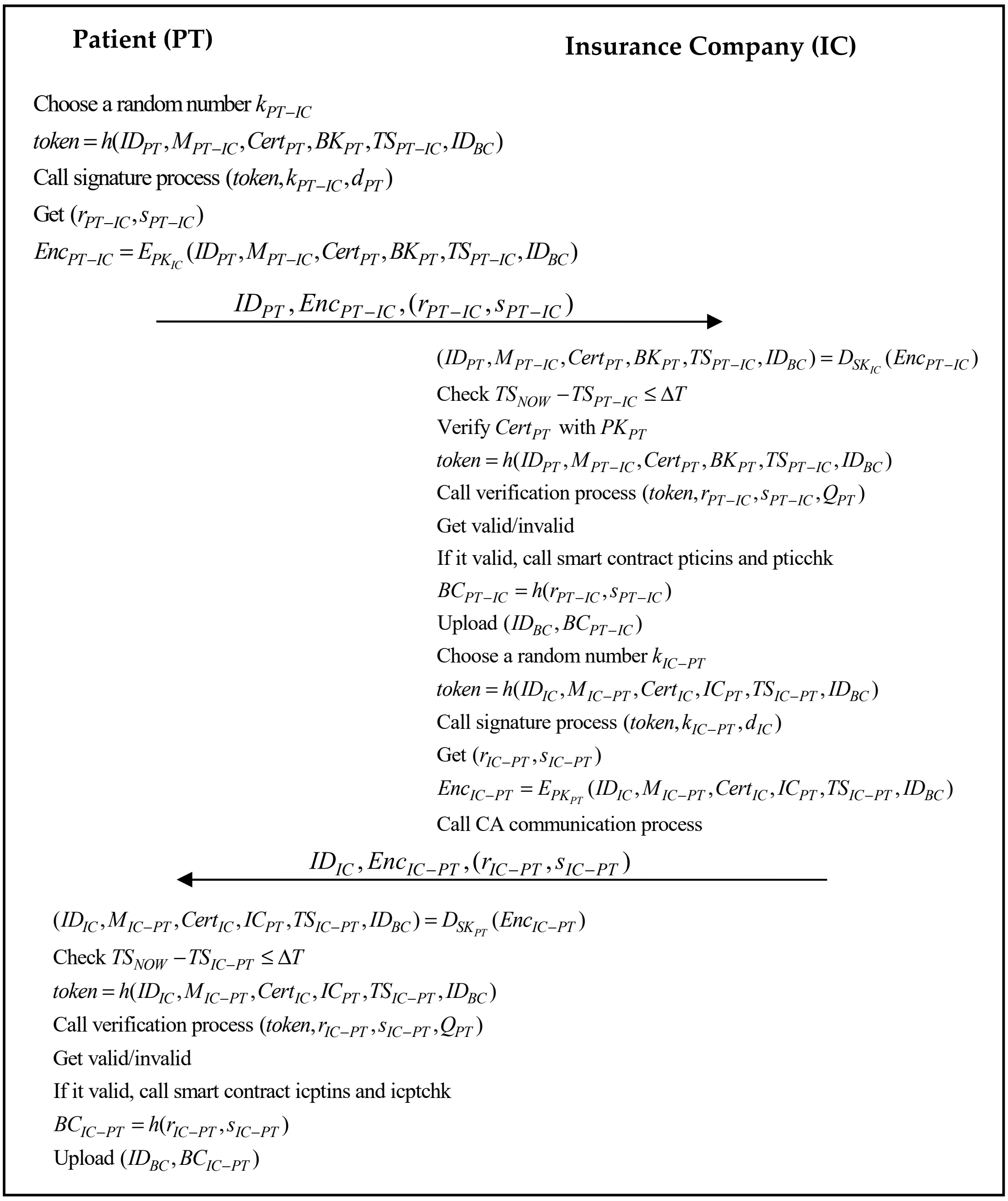
Step 1: PT generates a random value k_{PT-IC}, calculates

$$token = h(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC}) \tag{26}$$

calls signature process by $(token, k_{PT-IC}, d_{PT})$, obtains (r_{PT-IC}, s_{PT-IC}), calculates

$$Enc_{PT-IC} = E_{PK_{IC}}(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC}) \tag{27}$$

and sends $ID_{PT}, Enc_{PT-IC}, (r_{PT-IC}, s_{PT-IC})$ to IC.



Patient (PT) **Insurance Company (IC)**

Choose a random number k_{PT-IC}

$token = h(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC})$

Call signature process $(token, k_{PT-IC}, d_{PT})$

Get (r_{PT-IC}, s_{PT-IC})

$Enc_{PT-IC} = E_{PK_{IC}}(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC})$

$\xrightarrow{ID_{PT}, Enc_{PT-IC}, (r_{PT-IC}, s_{PT-IC})}$

$(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC}) = D_{SK_{IC}}(Enc_{PT-IC})$

Check $TS_{NOW} - TS_{PT-IC} \le \Delta T$

Verify $Cert_{PT}$ with PK_{PT}

$token = h(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC})$

Call verification process $(token, r_{PT-IC}, s_{PT-IC}, Q_{PT})$

Get valid/invalid

If it valid, call smart contract pticins and pticchk

$BC_{PT-IC} = h(r_{PT-IC}, s_{PT-IC})$

Upload (ID_{BC}, BC_{PT-IC})

Choose a random number k_{IC-PT}

$token = h(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC})$

Call signature process $(token, k_{IC-PT}, d_{IC})$

Get (r_{IC-PT}, s_{IC-PT})

$Enc_{IC-PT} = E_{PK_{PT}}(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC})$

Call CA communication process

$\xleftarrow{ID_{IC}, Enc_{IC-PT}, (r_{IC-PT}, s_{IC-PT})}$

$(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC}) = D_{SK_{PT}}(Enc_{IC-PT})$

Check $TS_{NOW} - TS_{IC-PT} \le \Delta T$

$token = h(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC})$

Call verification process $(token, r_{IC-PT}, s_{IC-PT}, Q_{PT})$

Get valid/invalid

If it valid, call smart contract icptins and icptchk

$BC_{IC-PT} = h(r_{IC-PT}, s_{IC-PT})$

Upload (ID_{BC}, BC_{IC-PT})

Figure 5. Insurance-purchasing phase.

Step 2: IC first calculates

$$(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC}) = D_{SK_{IC}}(Enc_{PT-IC}) \tag{28}$$

uses $TS_{NOW} - TS_{PT-IC} \le \Delta T$ to confirm whether the timestamp is valid, verifies $Cert_{PT}$ with PK_{PT}, and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC}) \tag{29}$$

calls the verification process by $(token, r_{PT-IC}, s_{PT-IC}, Q_{PT})$, and obtains valid/invalid. If the verification is passed, IC will obtain the message from PT and trigger the smart contracts pticins and pticchk. Algorithm 6 is as follows:

Algorithm 6. Smart contract pticins and pticchk of the proposed scheme.

```
function insert smart contract pticins (string pt_id, string pt_detail, string pt_cert, string pt_bkpt,
string pt_tsp) {
    count ++;
    pt[count].id = id;
    pt[count].detail = detail;
    pt[count].cert = cert;
    pt[count].bkpt = bkpt;
    pt[count].tsp = tsp;
}
sign string pt_key (pt_id, pt_detail, pt_cert, pt_bkpt, pt_tsp);
verify string pt_key (pt_id, pt_detail, pt_cert, pt_bkpt, pt_tsp);
function check smart contract pticchk (string pt_id, string pt_detail, string pt_cert, string pt_bkpt,
string pt_tsp) {
    return pt_id.exist;
    return pt_detail.exist;
    return pt_cert.exist;
    return pt_bkpt.exist;
    return pt_tsp.exist;
}
```

IC calculates

$$BC_{PT-IC} = h(r_{PT-IC}, s_{PT-IC}) \tag{30}$$

(ID_{BC}, BC_{PT-IC}) will also be uploaded to the blockchain center. Then, IC generates a random value k_{IC-PT} and calculates

$$token = h(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC}) \tag{31}$$

calls signature process by $(token, k_{IC-PT}, d_{IC})$, obtains (r_{IC-PT}, s_{IC-PT}), calculates

$$Enc_{IC-PT} = E_{PK_{PT}}(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC}) \tag{32}$$

calls CA communication process, and sends $ID_{IC}, Enc_{IC-PT}, (r_{IC-PT}, s_{IC-PT})$ to PT.

Step 3: PT first calculates

$$(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC}) = D_{SK_{PT}}(Enc_{IC-PT}) \tag{33}$$

uses $TS_{NOW} - TS_{IC-PT} \leq \Delta T$ to confirm whether the timestamp is valid and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC}) \tag{34}$$

calls verification process by $(token, r_{IC-PT}, s_{IC-PT}, Q_{PT})$, and obtains valid/invalid. If the verification is passed, PT will obtain the message from IC, and the smart contracts icptins and icptchk will be sent. Algorithm 7 is as follows:

Algorithm 7. Smart contract icptins and icptchk of the proposed scheme.

```
function insert smart contract icptins (string ic_id, string ic_detail, string ic_cert, string ic_icpt,
string ic_tsp) {
    count ++;
    ic[count].id = id;
    ic[count].detail = detail;
    ic[count].cert = cert;
    ic[count].icpt = icpt;
    ic[count].tsp = tsp;
}
sign string ic_key (ic_id, ic_detail, ic_cert, ic_icpt, ic_tsp);
verify string ic_key (ic_id, ic_detail, ic_cert, ic_icpt, ic_tsp);
function check smart contract icptchk (string ic_id, string ic_detail, string ic_cert, string ic_icpt,
string ic_tsp) {
    return ic_id.exist;
    return ic_detail.exist;
    return ic_cert.exist;
    return ic_icpt.exist;
    return ic_tsp.exist;
}
```

PT calculates

$$BC_{IC-PT} = h(r_{IC-PT}, s_{IC-PT}) \tag{35}$$

(ID_{BC}, BC_{IC-PT}) will also be uploaded to the blockchain center.

3.8. Patient Treatment Phase

When the patient PT visits a medical institution MI to see a doctor and informs the MI that he/she has purchased medical insurance, the MI will first verify the identity of the PT, read the electronic medical record of the PT, and then make a diagnosis; the record will be transmitted to the BCC through the CA. We present the flowchart of the patient treatment phase in Figure 6.

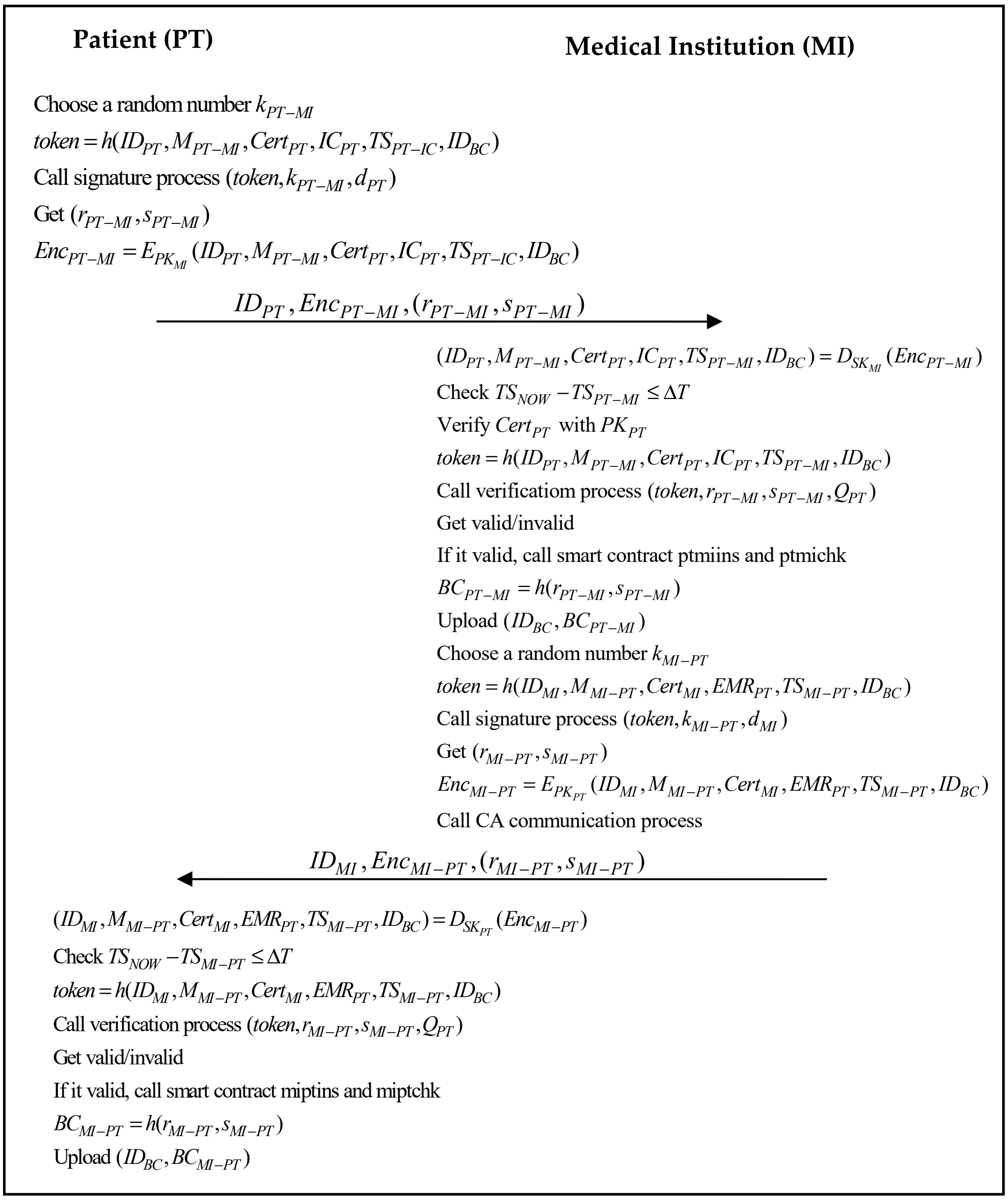
Step 1: PT generates a random value k_{PT-MI}, calculates

$$token = h(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-IC}, ID_{BC}) \tag{36}$$

calls signature process by $(token, k_{PT-MI}, d_{PT})$, obtains (r_{PT-MI}, s_{PT-MI}), calculates

$$Enc_{PT-MI} = E_{PK_{MI}}(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-IC}, ID_{BC}) \tag{37}$$

and sends $ID_{PT}, Enc_{PT-MI}, (r_{PT-MI}, s_{PT-MI})$ to MI.



Patient (PT) **Medical Institution (MI)**

Choose a random number k_{PT-MI}

$token = h(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-IC}, ID_{BC})$

Call signature process $(token, k_{PT-MI}, d_{PT})$

Get (r_{PT-MI}, s_{PT-MI})

$Enc_{PT-MI} = E_{PK_{MI}}(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-IC}, ID_{BC})$

$$\xrightarrow{ID_{PT}, Enc_{PT-MI}, (r_{PT-MI}, s_{PT-MI})}$$

$(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-MI}, ID_{BC}) = D_{SK_{MI}}(Enc_{PT-MI})$

Check $TS_{NOW} - TS_{PT-MI} \leq \Delta T$

Verify $Cert_{PT}$ with PK_{PT}

$token = h(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-MI}, ID_{BC})$

Call verificatiom process $(token, r_{PT-MI}, s_{PT-MI}, Q_{PT})$

Get valid/invalid

If it valid, call smart contract ptmiins and ptmichk

$BC_{PT-MI} = h(r_{PT-MI}, s_{PT-MI})$

Upload (ID_{BC}, BC_{PT-MI})

Choose a random number k_{MI-PT}

$token = h(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC})$

Call signature process $(token, k_{MI-PT}, d_{MI})$

Get (r_{MI-PT}, s_{MI-PT})

$Enc_{MI-PT} = E_{PK_{PT}}(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC})$

Call CA communication process

$$\xleftarrow{ID_{MI}, Enc_{MI-PT}, (r_{MI-PT}, s_{MI-PT})}$$

$(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC}) = D_{SK_{PT}}(Enc_{MI-PT})$

Check $TS_{NOW} - TS_{MI-PT} \leq \Delta T$

$token = h(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC})$

Call verification process $(token, r_{MI-PT}, s_{MI-PT}, Q_{PT})$

Get valid/invalid

If it valid, call smart contract miptins and miptchk

$BC_{MI-PT} = h(r_{MI-PT}, s_{MI-PT})$

Upload (ID_{BC}, BC_{MI-PT})

Figure 6. Patient treatment phase.

Step 2: MI first calculates

$$(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-MI}, ID_{BC}) = D_{SK_{MI}}(Enc_{PT-MI}) \tag{38}$$

uses $TS_{NOW} - TS_{PT-MI} \leq \Delta T$ to confirm whether the timestamp is valid, verifies $Cert_{PT}$ with PK_{PT} and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-MI}, ID_{BC}) \tag{39}$$

calls verification process by $(token, r_{PT-MI}, s_{PT-MI}, Q_{PT})$, and obtains valid/invalid. If the verification is passed, the MI will receive the message from PT and trigger the smart contracts ptmiins and ptmichk. Algorithm 8 is as follows:

Algorithm 8. Smart contract ptmiins and ptmichk of the proposed scheme.

```
function insert smart contract ptmiins (string pt_id, string pt_detail, string pt_cert, string pt_icpt,
string pt_tsp) {
    count ++;
    pt[count].id = id;
    pt[count].detail = detail;
    pt[count].cert = cert;
    pt[count].icpt = icpt;
    pt[count].tsp = tsp;
}
sign string pt_key (pt_id, pt_detail, pt_cert, pt_icpt, pt_tsp);
verify string pt_key (pt_id, pt_detail, pt_cert, pt_icpt, pt_tsp);
function check smart contract ptmichk (string pt_id, string pt_detail, string pt_cert, string pt_icpt,
string pt_tsp) {
    return pt_id.exist;
    return pt_detail.exist;
    return pt_cert.exist;
    return pt_icpt.exist;
    return pt_tsp.exist;
}
```

MI calculates

$$BC_{PT-MI} = h(r_{PT-MI}, s_{PT-MI}) \tag{40}$$

(ID_{BC}, BC_{PT-MI}) will also be uploaded to the blockchain center. Then MI generates a random value k_{MI-PT} and calculates

$$token = h(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC}) \tag{41}$$

calls signature process by $(token, k_{MI-PT}, d_{MI})$, obtains (r_{MI-PT}, s_{MI-PT}), calculates

$$Enc_{MI-PT} = E_{PK_{PT}}(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC}) \tag{42}$$

calls CA communication process, and sends $ID_{MI}, Enc_{MI-PT}, (r_{MI-PT}, s_{MI-PT})$ to PT.

Step 3: PT first calculates

$$(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC}) = D_{SK_{PT}}(Enc_{MI-PT}) \tag{43}$$

uses $TS_{NOW} - TS_{MI-PT} \leq \Delta T$ to confirm whether the timestamp is valid and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC}) \tag{44}$$

calls verification process by $(token, r_{MI-PT}, s_{MI-PT}, Q_{PT})$, and obtains valid/invalid. If the verification is passed, PT will receive the message from MI, and the smart contracts miptins and miptchk will be sent. Algorithm 9 is as follows:

Algorithm 9. Smart contract miptins and miptchk of the proposed scheme.

```
function insert smart contract miptins (string mi_id, string mi_detail, string mi_cert, string
mi_emrpt, string mi_tsp) {
    count ++;
    mi[count].id = id;
    mi[count].detail = detail;
    mi[count].cert = cert;
    mi[count].emrpt = emrpt;
    mi[count].tsp = tsp;
}
sign string mi_key (mi_id, mi_detail, mi_cert, mi_emrpt, mi_tsp);
verify string mi_key (mi_id, mi_detail, mi_cert, mi_emrpt, mi_tsp);
function check smart contract miptchk (string mi_id, string mi_detail, string mi_cert, string
mi_emrpt, string mi_tsp) {
    return mi_id.exist;
    return mi_detail.exist;
    return mi_cert.exist;
    return mi_emrpt.exist;
    eturn mi_tsp.exist;
}
```

PT calculates

$$BC_{MI-PT} = h(r_{MI-PT}, s_{MI-PT}), \tag{45}$$

(ID_{BC}, BC_{MI-PT}) will also be uploaded to the blockchain center.

3.9. Automatic Claims Phase

After the patient PT sees a doctor, the medical institution MI informs the insurance company IC to carry out insurance claims, and the IC obtains the diagnosis content related to the PT medical treatment provided by MI. If the claim is eligible, the IC will inform the PT of the claim amount and payment time, and the record will be sent to the BCC through the CA. We present the flowchart of the automatic claims phase in Figure 7.

Step 1: MI generates a random value k_{MI-IC}, calculates

$$token = h(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC}), \tag{46}$$

calls signature process by $(token, k_{MI-IC}, d_{MI})$, obtains (r_{MI-IC}, s_{MI-IC}), calculates

$$Enc_{MI-IC} = E_{PK_{IC}}(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC}), \tag{47}$$

and sends $ID_{MI}, Enc_{MI-IC}, (r_{MI-IC}, s_{MI-IC})$ to IC.

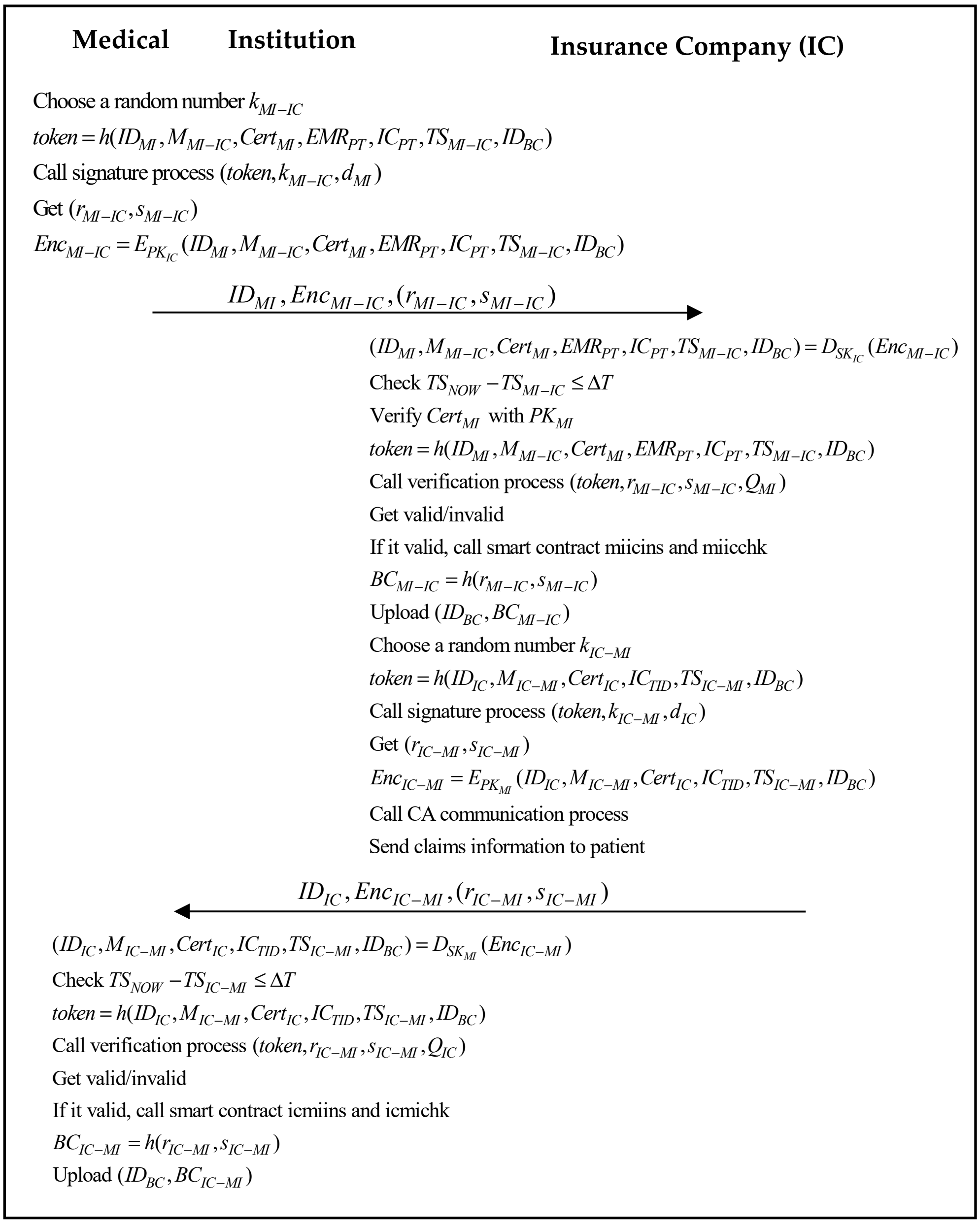

Medical Institution
Insurance Company (IC)
Choose a random number k_{MI-IC}
$token = h(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC})$
Call signature process $(token, k_{MI-IC}, d_{MI})$
Get (r_{MI-IC}, s_{MI-IC})
$Enc_{MI-IC} = E_{PK_{IC}}(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC})$
$ID_{MI}, Enc_{MI-IC}, (r_{MI-IC}, s_{MI-IC})$
$(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC}) = D_{SK_{IC}}(Enc_{MI-IC})$
Check $TS_{NOW} - TS_{MI-IC} \leq \Delta T$
Verify $Cert_{MI}$ with PK_{MI}
$token = h(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC})$
Call verification process $(token, r_{MI-IC}, s_{MI-IC}, Q_{MI})$
Get valid/invalid
If it valid, call smart contract miicins and miicchk
$BC_{MI-IC} = h(r_{MI-IC}, s_{MI-IC})$
Upload (ID_{BC}, BC_{MI-IC})
Choose a random number k_{IC-MI}
$token = h(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC})$
Call signature process $(token, k_{IC-MI}, d_{IC})$
Get (r_{IC-MI}, s_{IC-MI})
$Enc_{IC-MI} = E_{PK_{MI}}(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC})$
Call CA communication process
Send claims information to patient
$ID_{IC}, Enc_{IC-MI}, (r_{IC-MI}, s_{IC-MI})$
$(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC}) = D_{SK_{MI}}(Enc_{IC-MI})$
Check $TS_{NOW} - TS_{IC-MI} \leq \Delta T$
$token = h(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC})$
Call verification process $(token, r_{IC-MI}, s_{IC-MI}, Q_{IC})$
Get valid/invalid
If it valid, call smart contract icmiins and icmichk
$BC_{IC-MI} = h(r_{IC-MI}, s_{IC-MI})$
Upload (ID_{BC}, BC_{IC-MI})

Figure 7. Automatic claims phase.

Step 2: IC first calculates

$$(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC}) = D_{SK_{IC}}(Enc_{MI-IC}), \quad (48)$$

uses $TS_{NOW} - TS_{MI-IC} \leq \Delta T$ to confirm whether the timestamp is valid, verifies $Cert_{MI}$ with PK_{MI}, and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC}), \quad (49)$$

Calls the verification process by $(token, r_{MI-IC}, s_{MI-IC}, Q_{MI})$, and obtains valid/invalid. If the verification is passed, IC will receive the message from MI and trigger the smart contracts miicins and miicchk. Algorithm 10 is as follows:

Algorithm 10. Smart contract miicins and miicchk of the proposed scheme.

```
function insert smart contract miicins (string mi_id, string mi_detail, string mi_cert, string
mi_emrpt, string mi_icpt, string mi_tsp) {
    count ++;
    mi[count].id = id;
    mi[count].detail = detail;
    mi[count].cert = cert;
    mi[count].emrpt = emrpt;
    mi[count].icpt = icpt;
    mi[count].tsp = tsp;
}
sign string mi_key (mi_id, mi_detail, mi_cert, mi_emrpt, mi_icpt, mi_tsp);
verify string mi_key (mi_id, mi_detail, mi_cert, mi_emrpt, mi_icpt, mi_tsp);
function check smart contract miicchk (string mi_id, string mi_detail, string mi_cert, string
mi_emrpt, string mi_icpt, string mi_tsp) {
    return mi_id.exist;
    return mi_detail.exist;
    return mi_cert.exist;
    return mi_emrpt.exist;
    return mi_icpt.exist;
    return mi_tsp.exist;
}
```

IC calculates

$$BC_{MI-IC} = h(r_{MI-IC}, s_{MI-IC}), \tag{50}$$

(ID_{BC}, BC_{MI-IC}) will also be uploaded to the blockchain center. Then, IC generates a random value k_{IC-MI} and calculates

$$token = h(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC}), \tag{51}$$

calls signature process by $(token, k_{IC-MI}, d_{IC})$, obtains (r_{IC-MI}, s_{IC-MI}), calculates

$$Enc_{IC-MI} = E_{PK_{MI}}(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC}), \tag{52}$$

calls the CA communication process, and sends $ID_{IC}, Enc_{IC-MI}, (r_{IC-MI}, s_{IC-MI})$ to MI.

Step 3: MI first calculates

$$(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC}) = D_{SK_{MI}}(Enc_{IC-MI}), \tag{53}$$

uses $TS_{NOW} - TS_{IC-MI} \leq \Delta T$ to confirm whether the timestamp is valid and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC}), \tag{54}$$

calls verification process by $(token, r_{IC-MI}, s_{IC-MI}, Q_{IC})$, and obtains valid/invalid. If the verification is passed, MI will receive the message from IC, and the smart contracts icmiins and icmichk will be sent. Algorithm 11 is as follows:

Algorithm 11. Smart contract icmiins and icmichk of the proposed scheme.

```
function insert smart contract icmiins (string ic_id, string ic_detail, string ic_cert, string ic_ictid,
string ic_tsp) {
    count ++;
    ic[count].id = id;
    ic[count].detail = detail;
    ic[count].cert = cert;
    ic[count].ictid = ictid;
    ic[count].tsp = tsp;
}
sign string ic_key (ic_id, ic_detail, ic_cert, ic_ictid, ic_tsp);
verify string ic_key (ic_id, ic_detail, ic_cert, ic_ictid, ic_tsp);
function check smart contract icmichk (string ic_id, string ic_detail, string ic_cert, string ic_ictid,
string ic_tsp) {
    return ic_id.exist;
    return ic_detail.exist;
    return ic_cert.exist;
    return ic_ictid.exist;
    return ic_tsp.exist;
}
```

MI calculates

$$BC_{IC-MI} = h(r_{IC-MI}, s_{IC-MI}), \tag{55}$$

(ID_{BC}, BC_{IC-MI}) will also be uploaded to the blockchain center.

3.10. Bank Payment Phase

After the insurance company IC informs the PT of the claim amount and payment time, the insurance company IC then informs the bank (BK) to pay the patient (PT), and the record is transmitted to the BCC through the CA. We present the flowchart of the bank payment phase in Figure 8.

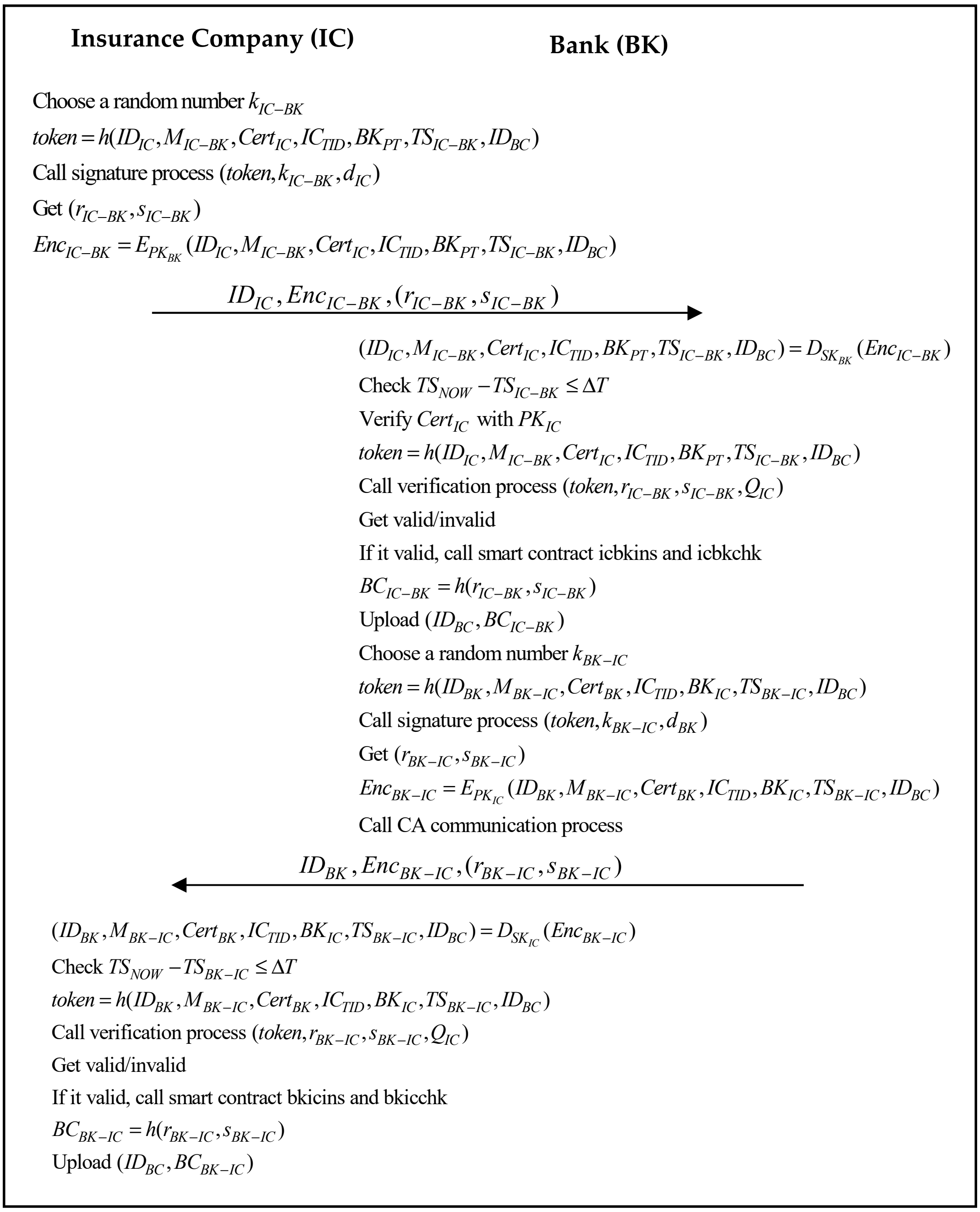
Step 1: IC generates a random value k_{IC-BK}, calculates

$$token = h(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC}) \tag{56}$$

calls the signature process by $(token, k_{IC-BK}, d_{IC})$, obtains (r_{IC-BK}, s_{IC-BK}), calculates

$$Enc_{IC-BK} = E_{PK_{BK}}(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC}) \tag{57}$$

and sends $ID_{IC}, Enc_{IC-BK}, (r_{IC-BK}, s_{IC-BK})$ to the BK.



Insurance Company (IC) **Bank (BK)**

Choose a random number k_{IC-BK}

$token = h(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC})$

Call signature process $(token, k_{IC-BK}, d_{IC})$

Get (r_{IC-BK}, s_{IC-BK})

$Enc_{IC-BK} = E_{PK_{BK}}(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC})$

$$\xrightarrow{ID_{IC}, Enc_{IC-BK}, (r_{IC-BK}, s_{IC-BK})}$$

$(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC}) = D_{SK_{BK}}(Enc_{IC-BK})$

Check $TS_{NOW} - TS_{IC-BK} \le \Delta T$

Verify $Cert_{IC}$ with PK_{IC}

$token = h(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC})$

Call verification process $(token, r_{IC-BK}, s_{IC-BK}, Q_{IC})$

Get valid/invalid

If it valid, call smart contract icbkins and icbkchk

$BC_{IC-BK} = h(r_{IC-BK}, s_{IC-BK})$

Upload (ID_{BC}, BC_{IC-BK})

Choose a random number k_{BK-IC}

$token = h(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC})$

Call signature process $(token, k_{BK-IC}, d_{BK})$

Get (r_{BK-IC}, s_{BK-IC})

$Enc_{BK-IC} = E_{PK_{IC}}(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC})$

Call CA communication process

$$\xleftarrow{ID_{BK}, Enc_{BK-IC}, (r_{BK-IC}, s_{BK-IC})}$$

$(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC}) = D_{SK_{IC}}(Enc_{BK-IC})$

Check $TS_{NOW} - TS_{BK-IC} \le \Delta T$

$token = h(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC})$

Call verification process $(token, r_{BK-IC}, s_{BK-IC}, Q_{IC})$

Get valid/invalid

If it valid, call smart contract bkicins and bkicchk

$BC_{BK-IC} = h(r_{BK-IC}, s_{BK-IC})$

Upload (ID_{BC}, BC_{BK-IC})

Figure 8. Bank payment phase.

Step 2: BK first calculates

$$(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC}) = D_{SK_{BK}}(Enc_{IC-BK}), \tag{58}$$

uses $TS_{NOW} - TS_{IC-BK} \leq \Delta T$ to confirm whether the timestamp is valid, verifies $Cert_{IC}$ with PK_{IC} and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC}), \tag{59}$$

calls the verification process by $(token, r_{IC-BK}, s_{IC-BK}, Q_{IC})$, and obtains valid/invalid. If the verification is passed, the BK will receive the message from the IC and trigger the smart contracts icbkins and icbkchk. Algorithm 12 is as follows:

Algorithm 12. Smart contract icbkins and icbkchk of the proposed scheme.

```
function insert smart contract icbkins (string ic_id, string ic_detail, string ic_cert, string ic_ictid,
string ic_bkpt, string ic_tsp) {
    count ++;
    ic[count].id = id;
    ic[count].detail = detail;
    ic[count].cert = cert;
    ic[count].ictid = ictid;
    ic[count].bkpt = bkpt;
    ic[count].tsp = tsp;
}
sign string ic_key (ic_id, ic_detail, ic_cert, ic_ictid, ic_bkpt, ic_tsp);
verify string ic_key (ic_id, ic_detail, ic_cert, ic_ictid, ic_bkpt, ic_tsp);
function check smart contract icbkchk (string ic_id, string ic_detail, string ic_cert, string ic_ictid,
string ic_bkpt, string ic_tsp) {
    return ic_id.exist;
    return ic_detail.exist;
    return ic_cert.exist;
    return ic_ictid.exist;
    return ic_bkpt.exist;
    return ic_tsp.exist;
}
```

BK calculates

$$BC_{IC-BK} = h(r_{IC-BK}, s_{IC-BK}) \tag{60}$$

(ID_{BC}, BC_{IC-BK}) will also be uploaded to the blockchain center. Then, BK generates a random value k_{BK-IC}, calculates

$$token = h(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC}), \tag{61}$$

calls signature process by $(token, k_{BK-IC}, d_{BK})$, obtains (r_{BK-IC}, s_{BK-IC}), calculates

$$Enc_{BK-IC} = E_{PK_{IC}}(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC}), \tag{62}$$

calls CA communication process, and sends $ID_{BK}, Enc_{BK-IC}, (r_{BK-IC}, s_{BK-IC})$ to the IC.

Step 3: IC first calculates

$$(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC}) = D_{SK_{IC}}(Enc_{BK-IC}), \tag{63}$$

uses Check $TS_{NOW} - TS_{BK-IC} \leq \Delta T$ to confirm whether the timestamp is valid and then verifies the correctness of the ECDSA signature, calculates

$$token = h(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC}), \tag{64}$$

calls the verification process by $(token, r_{BK-IC}, s_{BK-IC}, Q_{IC})$, and obtains valid/invalid. If the verification is passed, the IC will receive the message from the BK, and the smart contracts bkicins and bkicchk will be sent. Algorithm 13 is as follows:

Algorithm 13. Smart contract bkicins and bkicchk of the proposed scheme.

```
function insert smart contract bkicins (string bk_id, string bk_detail, string bk_cert, string
bk_ictid, string bk_bkic, string bk_tsp) {
    count ++;
    bk[count].id = id;
    bk[count].detail = detail;
    bk[count].cert = cert;
    bk[count].ictid = ictid;
    bk[count].bkic = bkic;
    bk[count].tsp = tsp;
}
sign string bk_key (bk_id, bk_detail, bk_cert, bk_ictid, bk_bkic, bk_tsp);
verify string bk_key (bk_id, bk_detail, bk_cert, bk_ictid, bk_bkic, bk_tsp);
function check smart contract bkicchk (string bk_id, string bk_detail, string bk_cert, string
bk_ictid, string bk_bkic, string bk_tsp) {
    return bk_id.exist;
    return bk_detail.exist;
    return bk_cert.exist;
    return bk_ictid.exist;
    return bk_bkic.exist;
    return bk_tsp.exist;
}
```

IC calculates

$$BC_{BK-IC} = h(r_{BK-IC}, s_{BK-IC}), \tag{65}$$

and (ID_{BC}, BC_{BK-IC}) will also be uploaded to the blockchain center.

3.11. Arbitration Mechanism Phase

When there is a dispute between the PT, the IC, the BK, and the MI, we can conduct it through the arbitration mechanism. The PT requests arbitration from the AI, and the AI compares blockchain data through the signature message to confirm the insurance claim. We present the arbitration mechanism verification phase in Figure 9.

Step 1: The AI can download the certificate, signature, and blockchain data of the package through ID_{BC}, and then the AI verifies

$$BC_{PT-MI} \stackrel{?}{=} h(r_{PT-MI}, s_{PT-MI}) \tag{66}$$

and

$$BC_{PT-MI} \stackrel{?}{=} h(r_{PT-MI}, s_{PT-MI}). \tag{67}$$

If the verification fails, then the PT did not go to the MI for medical treatment.

Step 2: If the signature between the PT and the MI, and the blockchain data, are verified, the AI will verify

$$BC_{MI-IC} \stackrel{?}{=} h(r_{MI-IC}, s_{MI-IC}) \tag{68}$$

and

$$BC_{IC-MI} \stackrel{?}{=} h(r_{IC-MI}, s_{IC-MI}) \tag{69}$$

If the verification fails, then the MI did not send the insurance application to the IC.

Step 3: If the signature between the MI and the IC, and the blockchain data, are verified, the AI will verify

$$BC_{IC-BK} \stackrel{?}{=} h(r_{IC-BK}, s_{IC-BK}) \tag{70}$$

and

$$BC_{BK-IC} \overset{?}{=} h(r_{BK-IC}, s_{BK-IC}) \tag{71}$$

If the verification fails, then the BK did not receive the remittance request of the IC.

Step 4: If the signature between IC and BK, and the blockchain data are verified, the blockchain data has been fully verified, and the AI can confirm that the claim has been paid successfully.

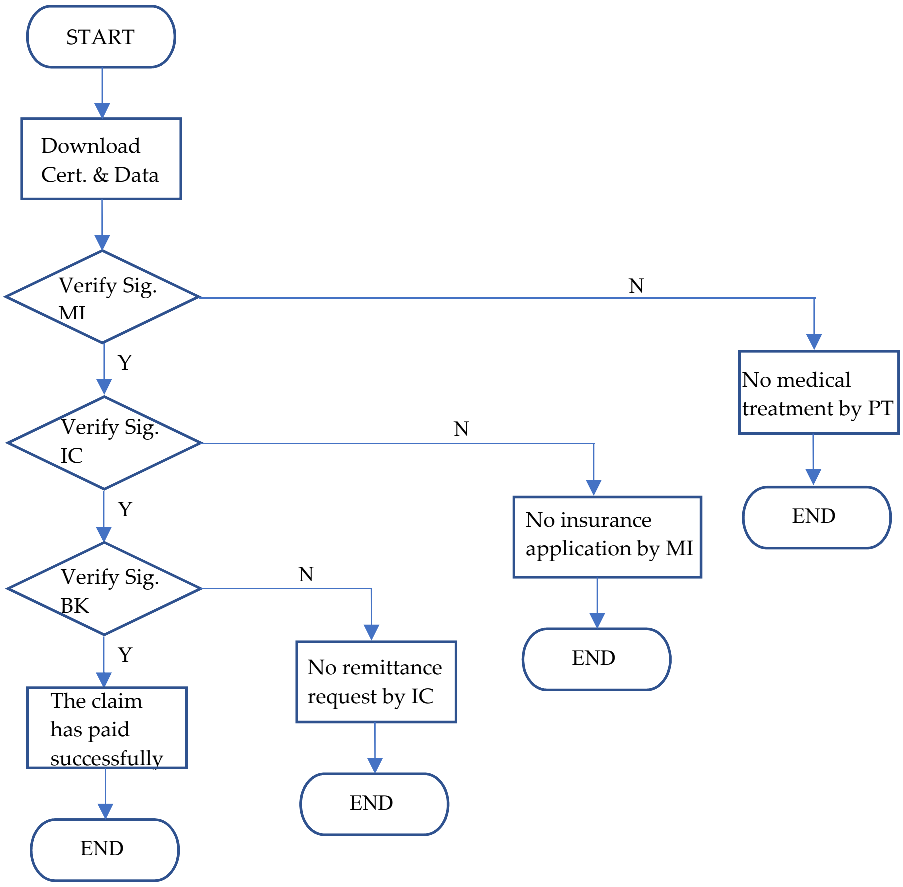

START
Download Cert. & Data
Verify Sig. MI
N
Y
No medical treatment by PT
END
Verify Sig. IC
N
Y
No insurance application by MI
END
Verify Sig. BK
N
Y
No remittance request by IC
END
The claim has paid successfully
END

Figure 9. The arbitration mechanism verification phase.

4. Security Analysis

4.1. Mutual Authentication

In each communication phase of the proposed scheme, the main goal of the scheme is to authenticate the private key of role A and role B. The system role A or role B can represent the competent authorities (CA), the medical institution (MI), the insurance company (IC), the bank (BK), and the patient (PT).

G1 : $A| \equiv A \overset{x_{B-A}}{\leftrightarrow} B$

G2 : $A| \equiv B| \equiv A \overset{x_{B-A}}{\leftrightarrow} B$

G3 : $B| \equiv A \overset{x_{A-B}}{\leftrightarrow} B$

G4 : $B| \equiv A| \equiv A \overset{x_{A-B}}{\leftrightarrow} B$

G5 : $A \equiv ID_B$

G6 : $A| \equiv B| \equiv ID_B$

G7 : $B| \equiv ID_A$

G8 : $B| \equiv A| \equiv ID_A$

According to the proposed scheme, BAN logic is applied to produce an idealized form as follows:

M1 $: (< ID_A, k_{A-B}, TS_{A-B} >_{PK_B}, < H(ID_A, k_{A-B}, TS_{A-B}) >_{x_{A-B}})$
M2 $: (< ID_B, k_{B-A}, TS_{B-A} >_{PK_A}, < H(ID_B, k_{B-A}, TS_{B-A}) >_{x_{B-A}})$

To analyze the proposed scheme, the following assumptions are made:

A1 $: A| \equiv \#(k_{A-B})$
A2 $: B| \equiv \#(k_{A-B})$
A3 $: A| \equiv \#(k_{B-A})$
A4 $: B| \equiv \#(k_{B-A})$
A5 $: A| \equiv B| \Rightarrow A \overset{x_{B-A}}{\leftrightarrow} B$
A6 $: B| \equiv A| \Rightarrow A \overset{x_{A-B}}{\leftrightarrow} B$
A7 $: A| \equiv B| \Rightarrow ID_B$
A8 $: B| \equiv A| \Rightarrow ID_A$

According to these assumptions and rules of BAN logic, the main proof of each communication phase is as follows:

a. The role B authenticates the role A.

We derive the following statement by *M1* and the *seeing rule*:

$$B \lhd (< ID_A, k_{A-B}, TS_{A-B} >_{PK_B}, < H(ID_A, k_{A-B}, TS_{A-B}) >_{x_{A-B}}) \tag{72}$$

We derive the following statement by *A2* and the *freshness rule*:

$$B| \equiv \#(< ID_A, k_{A-B}, TS_{A-B} >_{PK_B}, < H(ID_A, k_{A-B}, TS_{A-B}) >_{x_{A-B}}) \tag{73}$$

We derive the following statement by Equation (72), *A4*, and the *message meaning rule*:

$$B| \equiv A| \sim (< ID_A, k_{A-B}, TS_{A-B} >_{PK_B}, < H(ID_A, k_{A-B}, TS_{A-B}) >_{x_{A-B}}) \tag{74}$$

We derive the following statement by Equations (73) and (74), and the *nonce verification rule*:

$$B| \equiv A| \equiv (< ID_A, k_{A-B}, TS_{A-B} >_{PK_B}, < H(ID_A, k_{A-B}, TS_{A-B}) >_{x_{A-B}}) \tag{75}$$

We derive the following statement by Equation (75) and the *belief rule*:

$$B| \equiv A| \equiv A \overset{x_{A-B}}{\leftrightarrow} B \tag{76}$$

We derive the following statement by Equation (76), *A6*, and the *jurisdiction rule*:

$$B| \equiv A \overset{x_{A-B}}{\leftrightarrow} B \tag{77}$$

We derive the following statement by Equation (77) and the *belief rule*:

$$B| \equiv A| \equiv ID_A \tag{78}$$

We derive the following statement by Equation (78), *A8*, and the *jurisdiction rule*:

$$B| \equiv ID_A \tag{79}$$

b. The role A authenticates the role B.

We derive the following statement by *M2* and the *seeing rule*:

$$A \lhd (< ID_B, k_{B-A}, TS_{B-A} >_{PK_A}, < H(ID_B, k_{B-A}, TS_{B-A}) >_{x_{B-A}}) \tag{80}$$

We derive the following statement by *A1* and the *freshness rule*:

$$A| \equiv \#(< ID_B, k_{B-A}, TS_{B-A} >_{PK_A}, < H(ID_B, k_{B-A}, TS_{B-A}) >_{x_{B-A}}) \tag{81}$$

We derive the following statement by Equation (80), *A3*, and the *message meaning rule*:

$$A|\equiv B| \sim (< ID_B, k_{B-A}, TS_{B-A} >_{PK_A}, < H(ID_B, k_{B-A}, TS_{B-A}) >_{x_{B-A}}) \tag{82}$$

We derive the following statement by Equations (81) and (82), and the *nonce verification rule*:

$$A|\equiv B|\equiv (< ID_B, k_{B-A}, TS_{B-A} >_{PK_A}, < H(ID_B, k_{B-A}, TS_{B-A}) >_{x_{B-A}}) \tag{83}$$

We derive the following statement by Equation (83) and the *belief rule*:

$$A| \equiv B| \equiv A \overset{x_{B-A}}{\leftrightarrow} B \tag{84}$$

We derive the following statement by Equation (84), *A5*, and the *jurisdiction rule*:

$$A| \equiv A \overset{x_{B-A}}{\leftrightarrow} B \tag{85}$$

We derive the following statement by Equation (85) and the *belief rule*:

$$A| \equiv B| \equiv ID_B \tag{86}$$

We derive the following statement by Equation (86), *A7*, and the *jurisdiction rule*:

$$A| \equiv ID_B \tag{87}$$

By Equations (77), (79), (85) and (87), it can be proved that, in the proposed scheme, the role A and the role B authenticate each other. Moreover, it can also be proved that the proposed scheme can authenticate the private key of the role A and the role B.

In the automatic claims phase of the proposed scheme, for example, the insurance company authenticates the medical institution by $x_{MI-IC}' \overset{?}{=} r_{MI-IC} \bmod n$. If it passes the verification, the insurance company authenticates the legality of the medical institution. The medical institution authenticates the insurance company by $x_{IC-MI}' \overset{?}{=} r_{IC-MI} \bmod n$. If it passes the verification, the medical institution authenticates the legality of the insurance company. The automatic claims phase of the proposed scheme thus guarantees mutual authentication between the medical institution and the insurance company.

4.2. Resists Man-in-the-Middle Attacks

To achieve the goal in this study, we used digital signatures and public-key technology. When a malicious person alters the data, the receiver can verify whether the data has been altered by verifying the signature. The user can further use the public key to encrypt data for transmission. The malicious attacker will not be able to obtain encrypted data because they do not have the correct private key. The data cannot be tampered with because a malicious attacker cannot access it.

Scenario: the attacker intercepts the message from the sender to the accessing party and tampers with the message before sending it to the accessing party.

Analysis: the attacker will fail because our proposed method uses public keys to encrypt data as follows:

$$\begin{aligned}
&token = h(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC})\\
&token = h(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC})\\
&token = h(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC})\\
&token = h(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC})\\
&token = h(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-IC}, ID_{BC})\\
&token = h(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC})\\
&token = h(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC})\\
&token = h(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC})
\end{aligned}$$

$$token = h(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC})$$
$$token = h(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC})$$

The malicious attacker will not be able to obtain encrypted data because they do not have a correct private key. Even if the attacker intercepts the message, the attacker cannot decrypt the message and tamper with the message. In summary, the attacker cannot modify the message and send it to the accessing party.

4.3. Verifiable

Digital certificate verification can be used publicly to verify the identity of each role, and the authorization information was also published. It is based on the openness and transparency of the information on the blockchain, which will truly realize the specialization and high efficiency in the field of medical insurance claims service.

Take the message transmitted by the PT and the IC as an example. When the PT sends a message signed by ECDSA to the IC, the IC first verifies the correctness of the timestamp and signature then generates blockchain data $BC_{PT-IC} = h(r_{PT-IC}, s_{PT-IC})$ and uses ID_{BC} as an index to upload the blockchain data to the competent authorities. That is, after verifying the correctness of the timestamp and signature for each role that receives the message, it also verifies the correctness of the blockchain data generated by the previous role. Therefore, our proposed solution achieves the characteristics of public verification through blockchain technology and ECDSA digital signature.

4.4. Integrity and Unforgery

The timestamp and signature mechanism is used to irreversibly generate a string composed of random numbers and letters for the data placed in each block. This original text cannot be inferred from the string, thus effectively solving the trust problem. After the hash function operation, the messages are described as follows.

$$Enc_{AP-CA} = E_{PK_{CA}}(ID_{AP}, M_{AP-CA}, Cert_{AP}, TS_{AP-CA}, ID_{BC})$$
$$Enc_{CA-AP} = E_{PK_{AP}}(ID_{CA}, M_{CA-AP}, Cert_{CA}, TS_{CA-AP}, ID_{BC})$$
$$Enc_{PT-IC} = E_{PK_{IC}}(ID_{PT}, M_{PT-IC}, Cert_{PT}, BK_{PT}, TS_{PT-IC}, ID_{BC})$$
$$Enc_{IC-PT} = E_{PK_{PT}}(ID_{IC}, M_{IC-PT}, Cert_{IC}, IC_{PT}, TS_{IC-PT}, ID_{BC})$$
$$Enc_{PT-MI} = E_{PK_{MI}}(ID_{PT}, M_{PT-MI}, Cert_{PT}, IC_{PT}, TS_{PT-IC}, ID_{BC})$$
$$Enc_{MI-PT} = E_{PK_{PT}}(ID_{MI}, M_{MI-PT}, Cert_{MI}, EMR_{PT}, TS_{MI-PT}, ID_{BC})$$
$$Enc_{MI-IC} = E_{PK_{IC}}(ID_{MI}, M_{MI-IC}, Cert_{MI}, EMR_{PT}, IC_{PT}, TS_{MI-IC}, ID_{BC})$$
$$Enc_{IC-MI} = E_{PK_{MI}}(ID_{IC}, M_{IC-MI}, Cert_{IC}, IC_{TID}, TS_{IC-MI}, ID_{BC})$$
$$Enc_{IC-BK} = E_{PK_{BK}}(ID_{IC}, M_{IC-BK}, Cert_{IC}, IC_{TID}, BK_{PT}, TS_{IC-BK}, ID_{BC})$$
$$Enc_{BK-IC} = E_{PK_{IC}}(ID_{BK}, M_{BK-IC}, Cert_{BK}, IC_{TID}, BK_{IC}, TS_{BK-IC}, ID_{BC})$$

The hash value cannot be reversed back to the original content, so this agreement achieves the characteristic that the message cannot be tampered with.

4.5. Traceable

After the message is on the chain, the data block containing the transaction information is permanently stored on the blockchain and cannot be tampered with. For example, when we want to verify and trace whether the blockchain data between the MI and the IC is legal in the automatic claims phase, we can compare and verify $BC_{MI-IC} = h(r_{MI-IC}, s_{MI-IC})$ and $BC_{IC-MI} = h(r_{IC-MI}, s_{IC-MI})$. When we want to verify and trace whether the blockchain data between the IC and the BK is legal in the bank payment phase, we can compare and verify $BC_{IC-BK} = h(r_{IC-BK}, s_{IC-BK})$ and $BC_{BK-IC} = h(r_{BK-IC}, s_{BK-IC})$.

4.6. Openness

With the setting of the public key and private key, except that the private information of the transaction subject is encrypted, everyone can query blockchain data and develop related applications through the public interface. The system information is open and transparent, reducing information asymmetry and solving the problems of moral hazard and adverse selection between insurance supply and demand; with the help of openness, the use of big data and cloud computing can be improved, and insurance product development and pricing can be more accurate.

4.7. Privacy

The alliance chain combines blockchain technology, and only the data that the policyholder agrees to share will be stored on the blockchain. Blockchain technology can not only allow authorized persons to access data through signature private keys, encryption technology, and secure multi-party computing technology but also ensure that the core data and privacy of the blockchain alliance member database are not leaked. To protect the privacy of users, the content of the insurance contract is restricted to access. Only the litigant can view the personal contract, and the key is in the hands of the litigant. The contract review, query, modification, and other information will occur and record in the blockchain, and the insurance contract is fully executed automatically through smart contracts.

4.8. Decentralization and Information Sharing

In the proposed scheme, information is processed by each role and signed by the role with a private key. Under the structure of the Hyperledger, alliance members must be registered before they can share information. For members within the same alliance, the circulation of all information is open and transparent. One node cannot deceive other nodes. In this way, the trust relationship between nodes is realized, making it possible to obtain trust between nodes at a low cost. Therefore, the proposed scheme realizes the decentralization and information sharing of alliance members.

4.9. Non-Repudiation

The content of the message sent by each role is signed by the sender with its ECDSA private key. After receiving the message, the receiver will verify the message with the sender's public key. If the message is successfully verified, the sender will not deny the content of the message transmitted. Once the data is verified and added to the blockchain, it will be permanently stored, and the inherent time stamp function of the blockchain can record the creation time. The information is changed in order to control more than 51% of the nodes in the system, which is a significant difficulty. Table 2 is an undeniable description of each role in the proposed scheme.

Table 2. Non-repudiation of the proposed scheme.

Item Phase	Signature	Sender	Receiver	Signature Verification
CA communication process	(r_{AP-CA}, s_{AP-CA})	AP	CA	$x_{AP-CA}{}' \overset{?}{=} r_{AP-CA} \bmod n$
	(r_{CA-AP}, s_{CA-AP})	CA	AP	$x_{CA-AP}{}' \overset{?}{=} r_{CA-AP} \bmod n$
Insurance-purchasing phase	(r_{PT-IC}, s_{PT-IC})	PT	IC	$x_{PT-IC}{}' \overset{?}{=} r_{PT-IC} \bmod n$
	(r_{IC-PT}, s_{IC-PT})	IC	PT	$x_{IC-PT}{}' \overset{?}{=} r_{IC-PT} \bmod n$
Patient treatment phase	(r_{PT-MI}, s_{PT-MI})	PT	MI	$x_{PT-MI}{}' \overset{?}{=} r_{PT-MI} \bmod n$
	(r_{MI-PT}, s_{MI-PT})	MI	PT	$x_{MI-PT}{}' \overset{?}{=} r_{MI-PT} \bmod n$
Automatic claims phase	(r_{MI-IC}, s_{MI-IC})	MI	IC	$x_{MI-IC}{}' \overset{?}{=} r_{MI-IC} \bmod n$
	(r_{IC-MI}, s_{IC-MI})	IC	MI	$x_{IC-MI}{}' \overset{?}{=} r_{IC-MI} \bmod n$
Bank payment phase	(r_{IC-BK}, s_{IC-BK})	IC	BK	$x_{IC-BK}{}' \overset{?}{=} r_{IC-BK} \bmod n$
	(r_{BK-IC}, s_{BK-IC})	BK	IC	$x_{BK-IC}{}' \overset{?}{=} r_{BK-IC} \bmod n$

5. Discussions

5.1. Computation Cost

In Table 3, we analyzed the computational costs of each phase. We used the asymmetrical, comparison, hash function, and multiplication operation as the basis for calculating the cost.

Table 3. Computation costs of the proposed scheme.

Party Phase	PT	MI	IC	BK
Insurance purchasing phase	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$	*N/A*	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$	*N/A*
Patient treatment phase	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$	*N/A*	*N/A*
Automatic claims phase	*N/A*	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$	*N/A*
Bank payment phase	*N/A*	*N/A*	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$	$2T_{asy}$ + $2T_{cmp}$ +$3T_h$ + $7T_{mul}$

Notes: T_{asy}: the time required for an asymmetrical signature/verifying a signature. T_{cmp}: the time required for a comparison operation. T_h: the time required for a one-way hash function. T_{mul}: the time required for a multiplication operation.

5.2. Communication Performance

In Table 4, we analyzed the communication costs at each stage. In a 4G environment, the maximum transmission speed is 100 Mbps. In a 5G environment, the maximum transmission speed is 20 Gbps [48].

Table 4. Communication cost of the proposed scheme.

Party Phase	Message Length	Round	4G (100 Mbps)	5G (20 Gbps)
CA communication process	$2T_{sig}$ + $2T_{asy}$ + $5T_{ohter}$ = 2 * 512 + 2 * 1024 + 2 * 80 = 3232 bits	2	3232/102,400 = 0.032 ms	3232/20,480,000 = 0.16 us
Insurance purchasing phase	$2T_{sig}$ + $2T_{asy}$ + $5T_{ohter}$ = 2 * 512 + 2 * 1024 + 2 * 80 = 3232 bits	2	3232/102,400 = 0.032 ms	3232/20,480,000 = 0.16 us
Patient treatment phase	$2T_{sig}$ + $2T_{asy}$ + $5T_{ohter}$ = 2 * 512 + 2 * 1024 + 2 * 80 = 3232 bits	2	3232/102,400 = 0.032 ms	3232/20,480,000 = 0.16 us
Automatic claims phase	$2T_{sig}$ + $2T_{asy}$ + $5T_{ohter}$ = 2 * 512 + 2 * 1024 + 2 * 80 = 3232 bits	2	3232/102,400 = 0.032 ms	3232/20,480,000 = 0.16 us
Bank payment phase	$2T_{sig}$ + $2T_{asy}$ + $5T_{ohter}$ = 2 * 512 + 2 * 1024 + 2 * 80 = 3232 bits	2	3232/102,400 = 0.032 ms	3232/20,480,000 = 0.16 us

Notes: T_{sig}: the time required to transmit an ECDSA signature (512 bits). T_{asy}: the time required to transmit an asymmetric message (1024 bits). T_{ohter}: the time required to transmit information (80 bits).

5.3. Characteristic Comparison

In this section, we compare our proposed scheme and existing medical-related decentralized blockchain surveys in Table 5.

Table 5. Comparison of the proposed and existing medical-related decentralized blockchain surveys.

Authors	Year	Objective	1	2	3	4	5	6
Esposito et al. [3]	2018	Proposed a conceptual blockchain-based EMR/EHR/PHR ecosystem.	Y	NA	NA	Y	NA	Y
Xia et al. [7]	2018	Proposed a blockchain-based data sharing for electronic medical records in cloud environments.	Y	NA	Y	NA	Y	Y
Chen et al. [10]	2021	Proposed a blockchain-based preserving and sharing system for medical data privacy.	Y	NA	Y	Y	NA	Y
Johari et al. [11]	2021	Proposed a BLOSOM: BLOckchain technology for Security Of Medical records.	Y	NA	Y	NA	Y	Y
Our scheme	2021	Proposed a traceable online insurance claims system based on blockchain and smart contract technology.	Y	Y	Y	Y	Y	Y

Notes: 1: blockchain-focused, 2: automatic medical insurance claims service system, 3: detail protocol, 4: defend against attacks, 5: traceable, 6: security analysis. NA: not available, Y: yes.

6. Conclusions

Blockchain technology has changed people's traditional thinking, and many new business models have been derived from it. At this stage, many insurance companies have realized the important role of blockchain technology. In the future, "blockchain + insurance" will be used in a higher level and wider field in the insurance industry. Through this research, the following goals have been achieved:

(1) Realize automatic claims settlement through smart contracts.

From the occurrence of an insurance incident to the payment of indemnities, all information and data will be automatically generated through smart contracts, eliminating the need for investigation, damage assessment, and assessment. For example, if a vehicle has an insurance accident, the insurance information can be self-collected and uploaded to the insurance company, and the insurance company receives instructions to automatically pay the indemnity, which is faster than the current insurance claims and which saves labor costs and improves customer service experience.

(2) Realize customer identity security verification through information sharing.

At present, the insurance industry often has the problem of employees or agents imposing on customers to receive surrender or survival benefits for customers. The root cause is that insurance institutions do not control the identification of customers in place. If a customer is given a blockchain identity, the customer's identity information is no longer determined by the citizen ID but needs to be verified by all parties involved, which will largely eliminate the risks of various judicial cases in the industry.

(3) Establish an industry blacklist through data deposit

Due to the low threshold of the insurance industry, there are a large number of agents who violate the principle of good faith in the industry, and there are also many clients who have violated laws and regulations. However, due to the lack of a blacklist platform in the industry, the identification of practitioners and customers cannot form effective feedback. The establishment of blockchain data storage technology to establish an industry blacklist, and the establishment of an open and transparent blacklist database will be effective in combating insurance fraud.

(4) Improve the mutual insurance mechanism through traceability technology.

The main reason that restricts the development of mutual insurance is that participants cannot grasp the flow of each fund. The information traceability technology of the blockchain can ensure that participants clearly understand the expenditure and whereabouts of each fund so that participants can fully trust the mutual insurance organization. In an environment of full trust, mutual insurance organizations will achieve long-term development.

(5) Combating false claims through the subject information on the chain.

In property insurance contracts, insurers often lose control of the true conditions of the insurance subject after the contract is established. For example, cargo transportation insurance, artwork insurance, guarantee insurance, and other types of insurance are prone to large amounts of compensation due to lack of control. Through blockchain technology to link the underlying assets to the chain, the entire process of tracking and management of the underlying assets of insurance is realized, to protect the true contractual benefits and prevent the risks of repeated insurance, out of control of the target, and false claims.

We applied electronic medical records (EMR) in this system, which can have significant futuristic expansion. The biggest breakthrough of EMR is that they can be read by multiple people at the same time, and because the information is quite complete, they can also be used for research purposes to improve the quality of medical care. It is worth mentioning that every citizen must be responsible for his own health. Therefore, the function of EMR is not only to record medical records but a kind of "personal health record", including dietary records and planning, exercise planning, blood sugar, blood pressure, and other information. It can even store data produced by non-medical institutions, which means to integrate and classify data from different sources. In this way, it will be more convenient for people to manage their own health.

Moreover, the proposed scheme needs an environment with public key infrastructure (PKI), which is the necessary infrastructure. In addition, the medical institutions in the environment must form an alliance and reach the premise that the members of the alliance share the electronic medical records (EMR) of patients. Insurers and banks in this environment must also form an alliance, and members of the financial alliance must share customer credit information. Therefore, to actually apply this system to each country, the country must first have public key infrastructure (PKI), a medical alliance that can share patient electronic medical records (EMR), and a financial alliance that can share customers' credit information.

On the whole, blockchain technology has shown many promising applications in the insurance industry. This will play a greater role in the ideological collision and technological integration between blockchain technology and the insurance industry and will play a greater role in "serving the real economy and preventing financial risks".

Author Contributions: Conceptualization, C.-L.C. and Y.-Y.D.; methodology, C.-L.C. and Y.-Y.D.; validation, W.-J.T., C.-T.L. and C.-C.L.; investigation, C.-M.W.; data analysis, C.-L.C.; writing—original draft preparation, Y.-Y.D.; writing—review and editing, W.-J.T., C.-T.L., C.-C.L. and C.-M.W.; supervision, C.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C., under Contract MOST 110-2218-E-305-001-MBK and Contract MOST 110-2410-H-324-004-MY2, and the National Natural Science Foundation of China (No. 51808474).

Institutional Review Board Statement: This study was based only on basic theoretical research. It did not involve any human participants.

Informed Consent Statement: This study was based only on basic theoretical research. It did not involve any human participants.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. International Association of Insurance Supervisors (IAIS). Available online: https://www.iaisweb.org/home (accessed on 20 August 2021).
2. Raikwar, M.; Mazumdar, S.; Ruj, S.; Gupta, S.S.; Chattopadhyay, A.; Lam, K.Y. A Blockchain Framework for Insurance Processes. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018. [CrossRef]

3. Limin, H.; Jianmin, Y. Application Research of Blockchain in the Field of Medical Insurance. In Proceedings of the 2019 3rd International Conference on Economics, Management Engineering and Education Technology (ICEMEET 2019), Suzhou, China, 18–19 May 2019. [CrossRef]
4. Zhang, X. Design and Implementation of Medical Insurance System Based on Blockchain Smart Contract Technology. Master's Thesis, Huazhong University of Science & Technology, Wuhan, China, May 2019.
5. Esposito, C.; De Santis, A.; Tortora, G.; Chang, H.; Choo, K.K.R. Blockchain: A panacea for healthcare cloud-based data security and privacy? *IEEE Cloud Comput.* **2018**, *5*, 31–37. [CrossRef]
6. Novo, O. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
7. Wang, J.; Li, M.; He, Y.; Li, H.; Xiao, K.; Wang, C. A blockchain based privacy-preserving incentive mechanism in crowdsensing applications. *IEEE Access* **2018**, *6*, 17545–17556. [CrossRef]
8. Dorri, A.; Steger, M.; Kanhere, S.S.; Jurdak, R. Blockchain: A distributed solution to automotive security and privacy. *IEEE Commun. Mag.* **2017**, *55*, 119–125. [CrossRef]
9. Xia, Q.; Sifah, E.; Smahi, A.; Amofa, S.; Zhang, X. BBDS: Blockchain-Based data sharing for electronic medical records in cloud environments. *Information* **2017**, *8*, 44. [CrossRef]
10. Xu, J.; Xue, K.; Li, S.; Tian, H.; Hong, J.; Hong, P.; Yu, N. Healthchain: A blockchain-based privacy preserving scheme for large-scale health data. *IEEE Internet Things J.* **2019**, *6*, 8770–8781. [CrossRef]
11. Liu, X.; Wang, Z.; Jin, C.; Li, F.; Li, G. A Blockchain-based medical data sharing and protection scheme. *IEEE Access* **2019**, *7*, 118943–118953. [CrossRef]
12. Chen, Z.; Xu, W.; Wang, B.; Yu, H. A blockchain-based preserving and sharing system for medical data privacy. *Future Gener. Comput. Syst.* **2021**, *124*, 338–350. [CrossRef]
13. Johari, R.; Kumar, V.; Gupta, K.; Vidyarthi, D.P. BLOSOM: BLOckchain technology for Security of Medical records. *ICT Express* **2021**, in press. [CrossRef]
14. Chiuchisan, I.; Dimian, M. Internet of Things for e-Health: An approach to medical application. In Proceedings of the IEEE International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), Prague, Czech Republic, 29–30 October 2015; pp. 1–5.
15. Moosavi, S.R.; Gia, T.N.; Nigussie, E.; Rahmani, A.M.; Virtanen, S.; Tenhunen, H.; Isoaho, J. End-to-end security scheme for mobility enabled healthcare Internet of Things. *Future Gener. Comput. Syst.* **2016**, *64*, 108–124. [CrossRef]
16. Azeez, N.A.; Vyver, C.V.D. Security and privacy issues in e-health cloud-based system: A comprehensive content analysis. *Egypt. Inform. J.* **2019**, *20*, 97–108. [CrossRef]
17. Li, C.T.; Shih, D.H.; Wang, C.C. Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems. *Comput. Methods Programs Biomed.* **2018**, *157*, 191–203. [CrossRef]
18. Iribarren, S.J.; Brown, W.; Giguere, R.; Stone, P.; Schnall, R.; Staggers, N.; Carballo-Diéguez, A. Scoping review and evaluation of SMS/text messaging platforms for mHealth projects or clinical interventions. *Int. J. Med. Inform.* **2017**, *101*, 28–40. [CrossRef]
19. Khemissa, H.; Tandjaoui, D. A lightweight authentication scheme for e-health applications in the context of Internet of Things. In Proceedings of the 9th International Conference on Next Generation Mobile Applications, Services and Technologies, Paris, France, 29–30 June 2015; pp. 90–95.
20. Yang, Y.; Ma, M. Conjunctive keyword search with designated tester and timing enabled proxy re-encryption function for e-health clouds. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 746–759. [CrossRef]
21. Shakhovska, N.; Fedushko, S.; Greguš ml, M.; Melnykova, N.; Shvorob, I.; Syerov, Y. Big Data analysis in development of personalized medical system. *Proc. Comput. Sci.* **2019**, *160*, 229–234. [CrossRef]
22. Chen, C.C.; Deng, Y.Y.; Weng, W.; Sun, H.; Zhou, M. A Blockchain-Based Secure Inter-Hospital EMR Sharing System. *Appl. Sci.* **2020**, *10*, 4958. [CrossRef]
23. Chen, C.C.; Huang, P.T.; Deng, Y.Y.; Chen, H.C.; Wang, Y.C. A Secure Electronic Medical Record Authorization System for Smart device application in cloud computing environments. *Hum. Centr. Comput. Inf. Sci.* **2020**, *10*, 21. [CrossRef]
24. Wang, H.; Song, Y. Secure cloud-based EHR system using attribute-based cryptosystem and blockchain. *J. Med. Syst.* **2018**, *42*, 152. [CrossRef]
25. Buterin, V. A next-generation smart contract and decentralized application platform. *Ethereum White Paper* **2014**, *3*, 36.
26. Roy, S.; Das, A.K.; Chatterjee, S.; Kumar, N.; Chattopadhyay, S.; Rodrigues, J.J. Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. *IEEE Trans. Ind. Inform.* **2018**, *15*, 457–468. [CrossRef]
27. Wazid, M.; Das, A.K.; Kumari, S.; Li, X.; Wu, F. Provably secure biometric-based user authentication and key agreement scheme in cloud computing. *Secur. Commun. Netw.* **2016**, *9*, 4103–4119. [CrossRef]
28. Sureshkumar, V.; Amin, R.; Vijaykumar, V.R.; Sekar, S.R. Robust secure communication protocol for smart healthcare system with FPGA implementation. *Future Gener. Comput. Syst.* **2019**, *100*, 938–951. [CrossRef]
29. Roy, S.; Chatterjee, S.; Das, A.K.; Chattopadhyay, S.; Kumari, S.; Jo, M. Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things. *IEEE Internet Things J.* **2017**, *5*, 2884–2895. [CrossRef]

30. Banerjee, S.; Odelu, V.; Das, A.K.; Srinivas, J.; Kumar, N.; Chattopadhyay, S.; Choo, K.K.R. A provably secure and lightweight anonymous user authenticated session key exchange scheme for the Internet of Things deployment. *IEEE Internet Things J.* **2019**, *6*, 8739–8752. [CrossRef]
31. Shuai, M.; Yu, N.; Wang, H.; Xiong, L. Anonymous authentication scheme for smart home environment with provable security. *Comput. Secur.* **2019**, *86*, 132–146. [CrossRef]
32. Abbas, A.; Khan, S. A review on the state-of-the-art privacy preserving approaches in e-health clouds. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 1431–1441. [CrossRef] [PubMed]
33. Yang, J.; Li, J.; Niu, Y. A hybrid solution for privacy preserving medical data sharing in the cloud environment. *Future Gener. Comput. Syst.* **2015**, *43–44*, 74–86. [CrossRef]
34. Soni, P.; Pal, A.K.; Islam, S.H. An improved three-factor authentication scheme for patient monitoring using WSN in remote health-care system. *Comput. Methods Programs Biomed.* **2019**, *182*, 105054. [CrossRef]
35. Masdari, M.; Ahmadzadeh, S. A survey and taxonomy of the authentication schemes in Telecare Medicine Information Systems. *J. Netw. Comput. Appl.* **2017**, *87*, 1–19. [CrossRef]
36. Amin, R.; Islam, S.H.; Biswas, G.P.; Khan, M.K.; Kumar, N. A robust and anonymous patient monitoring system using wireless medical sensor networks. *Future Gener. Comput. Syst.* **2018**, *80*, 483–495. [CrossRef]
37. Chen, L.; Lee, W.K.; Chang, C.C.; Choo, K.K.R.; Zhang, N. Blockchain based searchable encryption for electronic health record sharing. *Future Gener. Comput. Syst.* **2019**, *95*, 420–429. [CrossRef]
38. Tanwar, S.; Parekh, K.; Evans, R. Blockchain-based electronic healthcare record system for healthcare 4.0 applications. *J. Inf. Secur. Appl.* **2020**, *50*, 102407. [CrossRef]
39. Szabo, N. Smart contracts: Building blocks for digital markets. *EXTROPY J. Transhum. Thought* **1996**, *18*, 16.
40. Szabo, N. The Idea of Smart Contracts. 1997. Available online: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_idea.html (accessed on 20 August 2021).
41. Vanstone, S. Responses to NIST's proposal. *Commun. ACM* **1992**, *35*, 50–52.
42. Johnson, D.; Menezes, A.; Vanstone, S. The Elliptic Curve Digital Signature Algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
43. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
44. Sierra, J.M.; Hernández, J.C.; Alcaide, A.; Torres, J. *Validating the Use of BAN LOGIC*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 851–858.
45. Hyperledger Fabric Docs. Available online: https://hyperledger-fabric.readthedocs.io/_/downloads/en/release-2.2/pdf/ (accessed on 20 August 2021).
46. Foschini, L.; Gavagna, A.; Martuscelli, G.; Montanari, R. Hyperledger Fabric Blockchain: Chaincode Performance Analysis. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6. [CrossRef]
47. Uddin, M. Blockchain Medledger: Hyperledger fabric enabled drug traceability system for counterfeit drugs in pharmaceutical industry. *Int. J. Pharm.* **2021**, *597*, 120235. [CrossRef]
48. Marcus, M.J. 5G and IMT for 2020 and beyond. *IEEE Wirel. Commun.* **2015**, *22*, 2–3. [CrossRef]