sensors

MDPI

Article

A Traceable Vaccine Supply Management System

Yaohong Ai [1], Chin-Ling Chen [2,3,*], Wei Weng [1], Mao-Lun Chiang [4,*], Yong-Yuan Deng [3] and Zi-Yi Lim [5]

[1] School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China
[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] Bachelor Degree Program of Artificial Intelligence, National Taichung University of Science and Technology, Taichung 40401, Taiwan
[5] Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan
* Correspondence: clc@mail.cyut.edu.tw (C.-L.C.); mlchiang@nutc.edu.tw (M.-L.C.)

Abstract: Everyone should be vaccinated, but the eligibility and safety of the vaccine are always overlooked by most people. The outbreak of COVID-19 has led many countries to intensify the development and production of the COVID-19 vaccine. and some countries have even required universal vaccination against this epidemic. However, such popularization of vaccination has also exposed various flaws in vaccine management that existed in the past, and vaccinators have become more concerned about the effectiveness of their vaccinations. In this paper, we propose a blockchain-based traceable vaccine management system. First, the system uses smart contracts to store the records generated during the whole process, from vaccine production to vaccination. Second, the proposed scheme uses the Edwards-curve digital signature algorithm (EdDSA) to guarantee the security and integrity of these data. Third, the system participants can access the corresponding data according to their authority to ensure the transparency of the whole system operation process. Finally, this paper will also conduct a security analysis of the whole system to ensure that the system can resist potential attacks by criminals.

Keywords: blockchain; vaccine safety; EdDSA; supply chain; transparency

Citation: Ai, Y.; Chen, C.-L.; Weng, W.; Chiang, M.-L.; Deng, Y.-Y.; Lim, Z.-Y. A Traceable Vaccine Supply Management System. *Sensors* **2022**, *22*, 9670. https://doi.org/10.3390/s22249670

Academic Editors: He Fang and Shaoshi Yang

Received: 9 November 2022
Accepted: 7 December 2022
Published: 10 December 2022

1. Introduction

1.1. Background

Since ancient times, humans have been constantly battling infectious diseases. Over thousands of years, many infectious diseases have devastated human society. The Black Death that struck Europe in 1347 claimed at least 25 million fresh lives in just four years, which is almost 40% of the total population of Europe [1]. The Spanish flu of 1918 infected about a quarter of the world's population at that time, causing about 50 million deaths [2]. The smallpox virus also caused about 400,000 deaths per year in 18th-century Europe [3]. However, from these tragic experiences, people have also come up with effective ways to combat infectious diseases. For example, the suppression of the Black Death was inseparable from the effective isolation of those infected with the Black Death by the government in the Middle Ages [4]. In addition, in 1796 the English physician Edward Jenner successfully prevented smallpox by implanting the cowpox virus into the human body; this material is also known as a vaccine [3]. Ever since then, vaccination, as an important part of modern disease control, has embodied the protection of susceptible people. The emergence of various vaccines against infectious diseases has also made vaccination the most economical and effective method of infectious disease prevention.

Nowadays, vaccination is closely related to people's lives. Many countries have implemented free vaccination policies to reduce the likelihood of citizens contracting some common infectious diseases. On the other hand, people are generally in agreement with the

act of vaccination. However, the management of vaccine-related records is still relatively backward. There are even some underdeveloped areas in the world that are still using the oldest paper form to record and save various messages about vaccines. This management of such a large number of records in physical form alone will inevitably lead to a series of huge safety risks for vaccine management. Meanwhile, the quality of vaccines in these less-developed regions will always be difficult to guarantee. Therefore, to address these problems, the World Health Organization (WHO) proposed a blueprint for improving global vaccine safety [5], emphasizing the need to safeguard vaccines in low- and middle-income countries (LMICs). In recent years, the rapid development of digital technology, such as the emergence of electronic health records (EHRs), has greatly facilitated the storage of medical data. Therefore, various types of vaccine-related records can be stored digitally in various government-run healthcare institutions. In this case, the safety of vaccines is generally ensured by governmental agencies, such as the lot release agency mentioned in the article [6], which conducts sampling surveys of vaccines, and only those vaccines that pass various quality tests can be sold. However, this way of managing vaccine information through a central institution also has many hidden dangers. For example, the failure of a single node will lead to the paralysis of the whole system [7,8], the increasing amount of various medical data will bring a huge burden to the servers of the central institution [8,9], and the central node is more vulnerable to the attacks of malicious nodes [10]. Supervision by central institutions likewise makes vaccine information highly opaque. If government regulation is not effective, vaccines that do not meet manufacturing standards will be extremely difficult to detect immediately. Once these fraudulent vaccines reach the market, they will cause great harm to the people and will bring a crisis of confidence to the government and the vaccine industry, in turn. In addition, the global outbreak of the COVID-19 pandemic in early 2020 has made several countries require universal COVID-19 vaccination regulations and issue corresponding vaccination certificates to those who have received the vaccine, and only those who hold such certificates are permitted to enter public places [11–15]. While recommending countries use electronic vaccination certificates, WHO formed the Smart Vaccination Certificate consortium [16,17] as a way to monitor the COVID-19 vaccination programs of individual countries and prevent the further spread of the epidemic. The traditional centralized vaccine record management approach is not only difficult to meet the demand for the frequent recall of increasingly large vaccine data but also faces serious security and privacy challenges. Therefore, the technology of vaccine record management needs some revolution. The emergence of blockchain technology will be able to bring innovation to the storage of vaccine records. Blockchain technology was first proposed by Satoshi Nakamoto in 2008 [18]. The essence of Blockchain is a distributed ledger that was initially used as the underlying technology for Bitcoin. Due to its characteristics of decentralization, transparency, security, and anonymity, blockchain technology was soon applied in various fields, among which it has been widely used in the medical field, where information is highly sensitive [19–21]. Blockchain also plays a pivotal role in dealing with a similar pandemic, and the European Parliament even listed blockchain technology as one of the top ten technologies that could effectively mitigate the impact of the COVID-19 epidemic [22].

In summary, a blockchain-based vaccine information management system is proposed in this paper. Our research goals are as follows:

1. The records generated from the manufacture, procurement, distribution, and vaccination up to the diagnosis of vaccine side effects and the identity information of each player will be permanently stored in the blockchain system. All information is guaranteed with integrity and security. In addition, each user in the system can trace the corresponding records based on his or her identity, ensuring the transparency of the whole system.
2. The use of Burrows-Abadi-Needham (BAN) logic ensures that two unfamiliar nodes confirm each other's identity, ensuring the authenticity of each other's identity and the trustworthiness of the information exchanged between nodes.

3. The system using blockchain can resist potential risks, such as replay attacks and man-in-the-middle attacks.

1.2. Related Works

Since Mettle [23] first proposed the application of blockchain in healthcare, numerous scholars have researched the application of blockchain technology in healthcare and have agreed that blockchain technology can make great contributions to improving the quality of healthcare services and enhancing the security of healthcare data. Gorden et al. [9] argued that all kinds of interoperability within the healthcare system would change from traditional institution-driven to patient-driven. They focus on the significant contribution that blockchain technology can make to this patient-driven model of healthcare interactions. Because of the revolution that blockchain can bring to the healthcare field, many articles have proposed corresponding blockchain-based healthcare systems. Azaria et al. [24] proposed a system called MedRec, which uses a permissioned blockchain to manage electronic medical records (EMRs). MedRec protects the security and integrity of EMRs while making them traceable by specific roles through authorization, and the ledgers generated during the operation of the system will be audited in case of disputes. However, the system is only a prototype without a working model, and the local storage of huge amounts of medical data is not reasonable. However, the scheme has no working model, and the local storage of massive amounts of medical data is not reasonable. Zhang et al. [20] developed a blockchain-based app named FHIRChain. The app normalizes and stores clinical data in Fast Healthcare Interoperability Resources (FHIR) standards. To solve the problem of clinical data silos, FHIRChain uses public keys to represent the identity of app users, thus ensuring that these standardized clinical data can be securely shared among authenticated medical personnel. Moreover, the article diagrams the components of the app and the user registration and authorization processes. However, the scheme does not perform a security analysis. Kumar et al. [25] designed a medical data-sharing system using Hyperledger Fabric. Each transaction in the system will be protected by an identity-based broadcast group signcryption scheme (IDBGSC). After passing the Practical Byzantine Fault Tolerance (PBFT) consensus protocol, the transaction information will be written into the ledger, which ensures that the security and integrity of these highly private medical data are guaranteed. The proposed system is then evaluated for security and performance to demonstrate its practical value. However, when the data in the system are disputed, it is not possible to quickly locate the actual signed individual users in the group. Kumar et al. [26] used federated chains and IPFS technology to manage patient information in a distributed manner. The system can easily store huge and cumbersome medical information under the chain, while the chain order stores the content that addresses the hashes of the files. This not only improves the system throughput but also makes the authorization of private information more convenient. However, the article has fewer parts for information security analysis and does not mention how to prevent some common attacks.

However, in contrast to the high enthusiasm of people studying how to use blockchain technology to secure healthcare data, researchers are less likely to think about how the technology can be used to secure vaccine-related information [27,28]. Sigwart et al. [27] suggested the possibility of applying blockchain technology to the vaccine supply chain, but a specific architectural part is missing. Yong et al. [28] proposed a system based on blockchain technology and machine learning to secure vaccines. Each vaccine has an exclusive Radio Frequency Identification (RFID) to prevent the possibility of vaccine information being falsified. The vaccine-related information recorded in the system can also be traced by consumers and regulators, solving problems such as vaccine expiration or vaccine information forgery. However, the article lacks an analysis of potential attacks.

Nevertheless, since the worldwide outbreak of the COVID-19 epidemic in 2020, many countries have had considerable requirements for vaccination and the checking of vaccination certificates against the pandemic [11–17]. These works on how to use blockchain to ensure the security of the vaccine and its related information also launched a fierce discus-

sion [29–33]. Ricci et al. [29] referred to the feasibility of applying blockchain technology in the transport of the COVID-19 vaccine and proof of COVID-19 vaccination. Deka et al. [30] proposed a method to maintain individual vaccination records and proof of immunization by introducing blockchain technology. Then, the vaccination records and proofs stored in the system are managed through IPFS. However, the content lacks a detailed description of the entire vaccine data management process. Antal et al. [31] used smart contracts to monitor vaccine distribution and vaccination. Every vaccine has its corresponding batch number, and temperature changes are constantly monitored by temperature sensors during the storage and transportation of the vaccine. Vaccine recipients can also trace the lot identification of the vaccine they received and access vaccine-related information after vaccination. Chauhan et al. [32] used smart contracts to implement four aspects of the system: registration of each role, monitoring of the vaccine distribution process, tamper-proofing of vaccine information, and vaccination feedback. The registration of each role will hash the private key of the role with its address and generate a unique QR code based on this hash value. Users can access the system based on their QR codes. However, both of the above articles are missing part of the security analysis. Chen et al. [33] proposed a blockchain-based vaccine record storage system. Although the system has a more complete system architecture and security analysis, the system has a complete framework and security analysis. The security of the elliptic curve digital signature algorithm (ECDSA) is not strongly guaranteed, and the proposed method involves the vaccination phase.

The remaining sections of this article are organized as follows. Section 2 briefly introduces the techniques mentioned in the article. Section 3 describes the details of the system design in detail. Section 4 provides a security analysis of the proposed scheme. Section 5 then analyzes the performance of the system. Finally, Section 6 concludes the paper to some extent.

2. Preliminary

2.1. Smart Contract

The concept of a smart contract was first introduced by Nick Szoba [34] and is essentially a computer program or transaction protocol that can be executed spontaneously. A smart contract can reduce the involvement of third-party intermediaries. As long as the participants reach an agreement with each other, the smart contract can be executed spontaneously according to the protocol. With the emergence of Ethereum [35], smart contracts have often been used in blockchains. Because of the feature that smart contracts can be executed spontaneously as long as the conditions are met, they reduce the possible omissions caused by manual operations in the system and also improve the operational efficiency of the blockchain system. In addition, because smart contracts do not require the participation of third-party trust institutions, the security and privacy of data are further ensured. On the other hand, smart contracts do not require the participation of third-party trust institutions, and the security and privacy of data are further ensured.

2.2. EdDSA

The Edwards-curve digital signature algorithm (EdDSA) was proposed by Bernstein et al. [36] in 2012. EdDSA uses a variant of the Schnorr signature based on twisted Edwards curves [37]. It has high performance across platforms while ensuring high security. Crucially, the random number value of the EdDSA is taken concerning the private key of the node and the content of the message sent, which overcomes the random number quality problem present in the ECDSA and the digital signature algorithm (DSA). Sony Corporation has caused a large number of cracks in Play Station 3 due to the random number quality problem of ECDSA [38].

2.3. BAN Logic

Burrows-Abadi-Needham (BAN) logic is a rule proposed by Burrows et al. [39] in 1990 for defining and analyzing message exchange protocols. This helps the user determine that

the information exchanged is trusted and that the process of exchanging information is without eavesdropping by third parties. To apply BAN logic, it is necessary to first transform the messages in the protocol into formulas in BAN logic and then make reasonable assumptions based on the specific situation.

2.4. Security Requirements

A system is always exposed to many risks, such as attacks by criminals and data leakage, so it is essential to analyze the potential risks. The vaccine management system proposed in this paper faces the following potential risks:

1. Mutual authentication: The exchange of data is necessary for the operation of the system. To guarantee the security and privacy of the exchanged information, both parties need to authenticate the identity of the other node.
2. Integrity: The integrity of the data exchanged throughout the vaccine management system should be ensured to prevent possible data tampering and loss.
3. High-quality random number: The system needs to generate high-quality random numbers to ensure that the digital signature is not easily forged, thereby ensuring the security of the whole system.
4. Non-repudiation: Each node should not deny its actions and send messages.
5. Man-in-the-Middle Attacks: Illegal third-party nodes intercept and obtain the messages being exchanged between two communicating parties in some way.
6. Replay attack: The attacker pretends to be a legitimate message sender and sends a message to the receiving node that it has received. This process can easily cause the disclosure of node identity information.
7. Sybil attack: Sybil attack is an online network security system threat in which an attacker attempts to control a network by creating multiple fake account identities, multiple nodes, or computer coordinates.

3. Method

In this section, some specific details of the system implementation will be covered. The first thing that needs to be discussed is the system architecture of the system. Some notations of the system will also be listed below.

3.1. System Architecture

The study proposes a blockchain-based record storage and sharing system for vaccines from production to vaccination. Figure 1 shows the main architecture of the system, which consists of six actors: the blockchain center, vaccine manufacturer, medical institution, medical personnel, vaccinated person, and arbitration institution. The detailed description is as follows.

1. Blockchain Center (BC): The blockchain center keeps most of the important information during the operation of the whole system. The registration of all nodes and the generation of public and private key pairs are done by this role. The mutual authentication between nodes will also be realized through the blockchain.
2. Vaccine Manufacturer (VM): A vaccine manufacturer is generally a third-party company that is qualified to manufacture vaccines. The vaccine is produced according to the vaccine procurement requirements of the medical institution. The vaccine manufacturer distributes the vaccine to the appropriate medical institutions in agreement with the medical institution. Moreover, it has regulatory responsibility for the transportation process.
3. Medical Institution (MI): Medical institutions purchase vaccines according to the targets given by the government. Upon receipt of the vaccine from the vaccine manufacturer, the medical institution is required to confirm the eligibility of the vaccine and store the vaccine. When the vaccine is about to be used, the medical institution needs to distribute the vaccine to the medical staff responsible for the vaccination.

4. Medical Personnel (MP): Medical personnel must be employed at the appropriate medical institution and have medical vaccination capabilities. After receiving the vaccines to be vaccinated on the day, medical personnel need to inoculate the vaccinated person.
5. Vaccinated Person (VP): Vaccinated persons are the group of people who are currently suitable for vaccination. Before vaccination, medical personnel will determine whether the vaccinated persons are eligible for vaccination by scanning the QR code. If vaccinated persons have some adverse reactions after vaccination, they will be required to submit details of the adverse reactions for further diagnosis.
6. Arbitration Institution (AI): In the case of a medical dispute that is difficult to reconcile, the arbitration institution will make a corresponding decision.

Figure 1 shows the scenario of the proposed scheme, which contains the business processes of user registration, vaccine procurement, vaccine distribution, vaccination, and side effect description. The details are as follows:

Step 1 Each role needs to register through the BC and obtain its corresponding public and private key pairs.

Step 2 The MI submits a vaccine request to a confirmed VM. Upon receiving the request, the VM verifies the identity of the MI. After confirming that the identity is correct, the VM starts to produce the vaccine and uploads the relevant data (vaccine lot number, vaccine manufacturer id, vaccine shelf life, etc.).

Step 3 Once the vaccine is made, the VM uploads the vaccine information to the BC and transports the vaccine to the corresponding MI. The vaccine transportation process requires strict compliance with transportation rules, such as the storage temperature range for each vaccine and the transport time requirements. After receiving the vaccine, the MI needs to verify that the vaccine information is correct. All related records need to be uploaded to the blockchain.

Step 4 MI needs to store vaccines after receiving them. Storage rules include a temperature between 2 °C and 8 °C, storage time cannot be longer than the remaining shelf life of the vaccine, etc. The information generated during storage needs to be uploaded to the BC. Then, the MI distributes the vaccine to be administered to qualified MP. After receiving the vaccine, the MP needs to confirm that the records related to the vaccine are accurate. Finally, the records in the blockchain are updated again.

Step 5 The VP goes to the vaccination site and submits his or her personal information and vaccination status to the MP before the vaccination. Vaccination can be carried out only after the MP has verified the information of the VP and confirmed that the VP is suitable for vaccination. After a vaccination is finished, the MP is required to update the vaccination information of the VP. The VP confirms that the vaccination information is correct. Then, the relevant records in BC will be updated.

Step 6 If a VP experiences side effects after receiving the vaccine, the VP must first provide the MP with his or her personal information, vaccination status, and details of the adverse reaction. After verifying the identity of the VP, the MP determines whether the adverse reaction is a vaccination side effect. If the adverse reaction is confirmed to be caused by the vaccination, the relevant records are uploaded to the BC. Meanwhile, the VP will receive further treatment.

Step 7 In the case of irreconcilable disputes throughout the system process, the arbitration department will obtain information from various parties for adjudication.

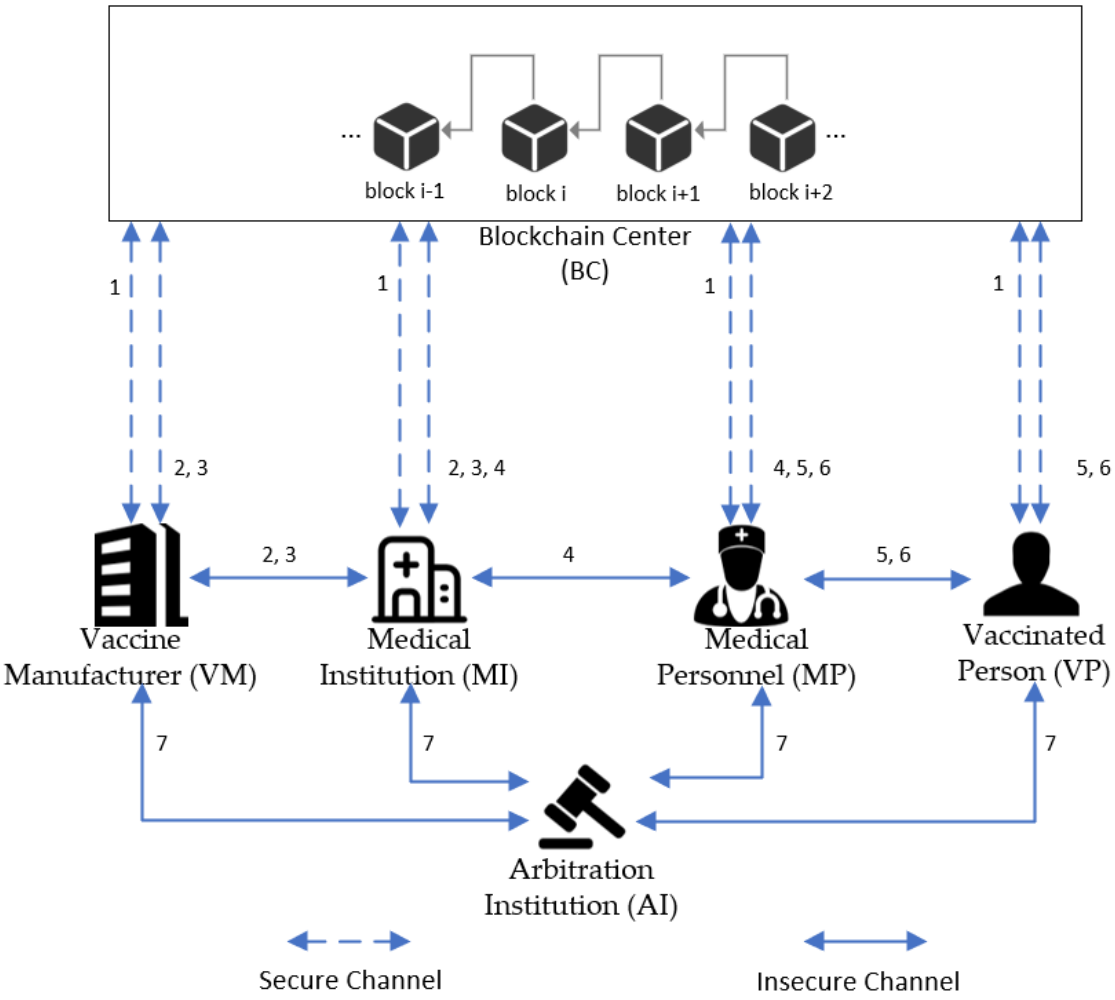

Figure 1. System architecture diagram.

3.2. Notation

The notation of the proposed scheme is shown below:

ID_X	The identity of X
$Cert_X$	The certificate of X
p	A k-bit prime number
$F(p)$	Finite group p
E	The elliptic curve defined on finite group p
G	A generating point based on E
b	An integer b with $2^{b-1} > p$
h_i	The ith bit of the hash value
n	An integer n with $3 \leq n \leq b$
(d_X, Q_X)	The EdDSA private key and public key of X
(R_X, S_X)	The EdDSA signature of X
(pk_X, sk_X)	The public key and private key of X
M_{X-Y}	The message from X to Y
$Enc_{pk_X}(M)$	Encrypted the message M with the public key of X
$Dec_{sk_X}(M)$	Decrypted the message M with the private key of X
$H()$	One-way hash function
r_X	The random value of X based on E
T_X	Timestamp message of X
$\triangle T$	The threshold for checking the validity of timestamps
$A \stackrel{?}{=} B$	Verify whether A is equal to B

3.3. Initial Phase

In this phase, we deployed a scalable blockchain center network based on the architecture of the Hyperledger Fabric, shown in Figure 2. The National Authority (NA) peer represents a peer controlled by government agencies, such as the Food and Drug Administration (FDA). These peers have the highest permission to use the system. The BCC also includes the certificate authority (CA), which is also generally authorized by the government to provide services to other access clients, such as VM, MI, and MP. The CA will give these access nodes the unique ID value, the public and private key pairs, and the certificate after they complete registration. The CA is also responsible for the renewal and revocation of these messages. Finally, BCC also includes ordering nodes (ON). The ON receives transactions containing signed and endorsed proposal responses from applications via a gateway service, and orders and packages the transactions into blocks. These ordered transactions are sent to peers for validation. When the consensus mechanism is passed, the peer commits the block to its ledger.

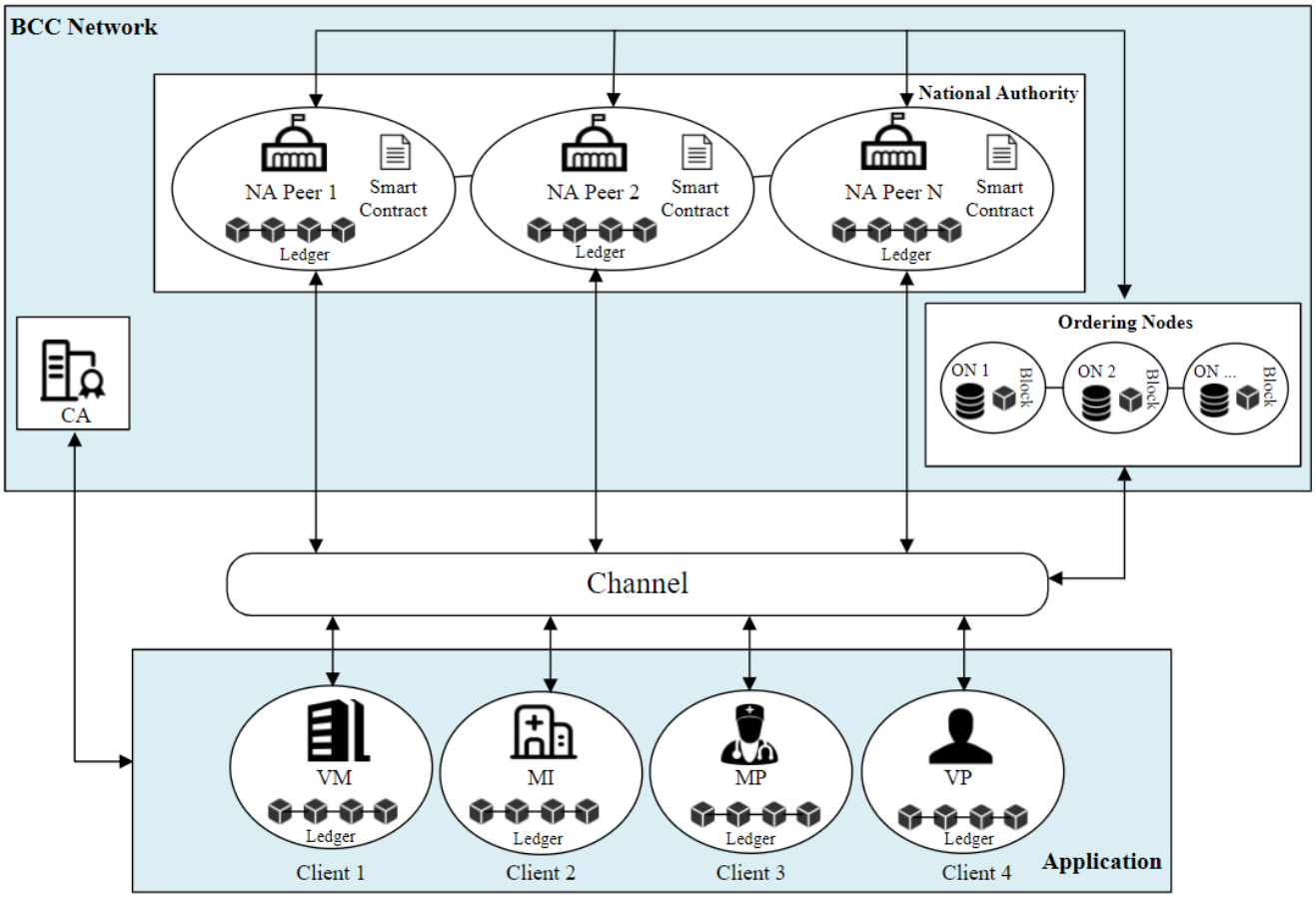

Figure 2. The architecture of the hyperledger fabric.

Moreover, the key information of each role designed within the system will be defined in the smart contract to ensure that the system can authenticate spontaneously and operate properly afterward. Figure 3 shows the smart contract framework associated with the system.

```
structure smart contract of apbcinfo{
    string ab id;
    string ab cert;
    string ab detail;
    string ab timestamp;
}
structure smart contract of seinfo{
    string se id;
    string se detail;
    string se cert;
    string se seDescribe;
    string se vpDetail;
    string se vLotId;
    string se vCert;
    string se timestamp;
}
```

```
structure smart contract of vmmiinfo{
    string vm id;
    string vm detail;
    string vm cert;
    string vm vLotId;
    string vm vDescribe;
    string vm vWhId;
    string vm vStoreWhDetail;
    string vm timestamp;
}
structure smart contract of mimpinfo{
    string mm id;
    string mm detail;
    string mm cert;
    string mm vLotId;
    string mm vStoreMiDetail;
    string mm timestamp;
}
```

```
structure smart contract of vtransinfo{
    string vt id;
    string vt detail;
    string vt cert;
    string vt vLotId;
    string vt vTrDetail;
    string vt timestamp;
}
structure smart contract of mpvpinfo{
    string mv id;
    string mv detail;
    string mv cert;
    string mv vLotId;
    string mv vpDetail;
    string mv vCert;
    string mv timestamp;
}
```

Figure 3. Smart contract structure of the scheme.

3.4. Registration Phase

In this phase, the access parties (AP) in the system are registered through the blockchain center. After receiving the registration request, the blockchain center issues the roles with the corresponding public–private key pairs and a digital certificate that can prove their identity. Figure 4 displays the process of the registration phase.

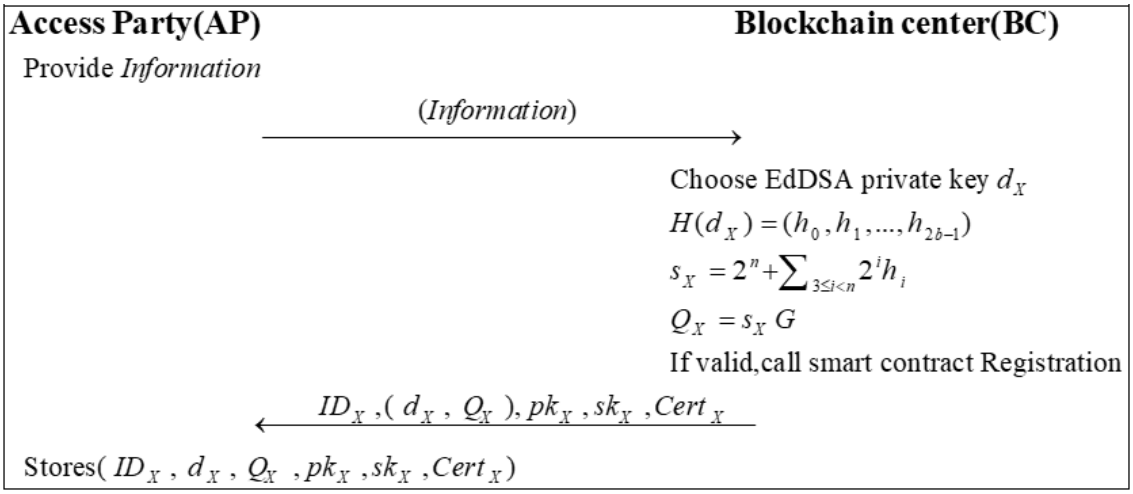

Figure 4. The process of the registration phase.

Step 1: Each AP sends basic information about itself (e.g., role ID) to the BC.
Step 2: The BC uses the EdDSA algorithm to generate a private key d_X, then calculates s_X and the corresponding public key Q_X by the following:

$$H(d_X) = (h_0, h_1, \ldots, h_{2b-1}) \tag{1}$$

$$s_X = 2^n + \sum_{3 \le i < n} 2^i h_i \tag{2}$$

$$Q_X = s_X G \tag{3}$$

If the identity of the AP is valid, the registered smart contract will be woken up. The algorithm for registration is shown in Algorithm 1. Then, the BC will send ID_X, d_X, Q_X, sk_X, pk_X, $Cert_X$ to AP.

Algorithm 1: The smart contract of registration.

```
Var APInfo[] APs;
fuction Registration(String x_id, String x_detail, Roles x_roleType){
    APInfo ap = new APInfo();
    AP.ID = x_id;
    AP.detail = x_detail;
    AP.roleType = x_roleType;
    return x_keypairs;
}
```

Step 3: The AP stores the $(ID_X, d_X, Q_X, pk_X, sk_X, Cert_X,)$ for later signature and verification.

3.5. EdDSA Authentication Phase

Identity authentication is required before any two nodes communicate with each other. This phase is mainly in the form of mutual authentication of identity with the other node by using the EdDSA digital signature, and only two legitimate nodes can pass information between them. Role A and role B can represent the vaccine manufacturer (VM), the medical institution (MI), the medical person (MP), and the vaccinated person (VP). The process of the EdDSA Authentication Phase is shown in Figure 5. Algorithm 2 shows the signature process of EdDSA, and Algorithm 3 shows the verification process of EdDSA.

Step 1: Role A calculates a random number r_{A-B} by encrypting the sent message M_{A-B} with the high b bits of the hash of the private key:

$$r_{A-B} = H(\, h_b, h_{b+1}, \ldots, h_{2b-1}, M_{A-B}) \tag{4}$$

Role A calls Algorithm 2 with $(r_{A-B}, s_A, M_{A-B}, Q_A)$ to sign the message and obtains the signature (R_{A-B}, S_{A-B}). Subsequently, role A uses the public key of role B pk_B to encrypt the message $(\, ID_A, M_{A-B}, T_{A-B})$ to generate Enc_{A-B}:

$$Enc_{A-B} = E_{pk_B}(\, ID_A, M_{A-B}, T_{A-B}) \tag{5}$$

Role A sends $ID_A, Enc_{A-B}, (R_{A-B}, S_{A-B})$ to role B.

Algorithm 2: The process of the EdDSA signature between role A and role B.

```
fuction Sign(String r, String s, String M, String Q){
    R = r × G;
    k = H(R, M, Q);
    S = (r + k × s);
    return(R, S);
}
```

Step 2: After receiving the message, role B first decrypts the message using its private key sk_B:

$$(\, ID_A, M_{A-B}, T_{A-B}) = D_{sk_B}(Enc_{A-B}) \tag{6}$$

Then, role B checks the validity of the timestamp:

$$T_{Now} - T_{A-B} \leq \Delta T \tag{7}$$

Next, role B calls Algorithm 3 to verify the signature based on the public information and the messages it received.

If the signature is valid, role B calculates a random number r_{B-A}:

$$r_{B-A} = H(h_b, h_{b+1}, \ldots, h_{2b-1}, M_{B-A}) \tag{8}$$

Role B calls Algorithm 2 with $(r_{B-A}, s_B, M_{B-A}, Q_B)$ to sign the message and generates the signature (R_{B-A}, S_{B-A}). Later, role B encrypts the message (ID_B, M_{B-A}, T_{B-A}) by using the public key of role A pk_A to generate Enc_{B-A}:

$$Enc_{B-A} = E_{pk_A}(ID_B, M_{B-A}, T_{B-A}) \tag{9}$$

Role B sends $ID_B, Enc_{B-A}, (R_{B-A}, S_{B-A})$ to role A.

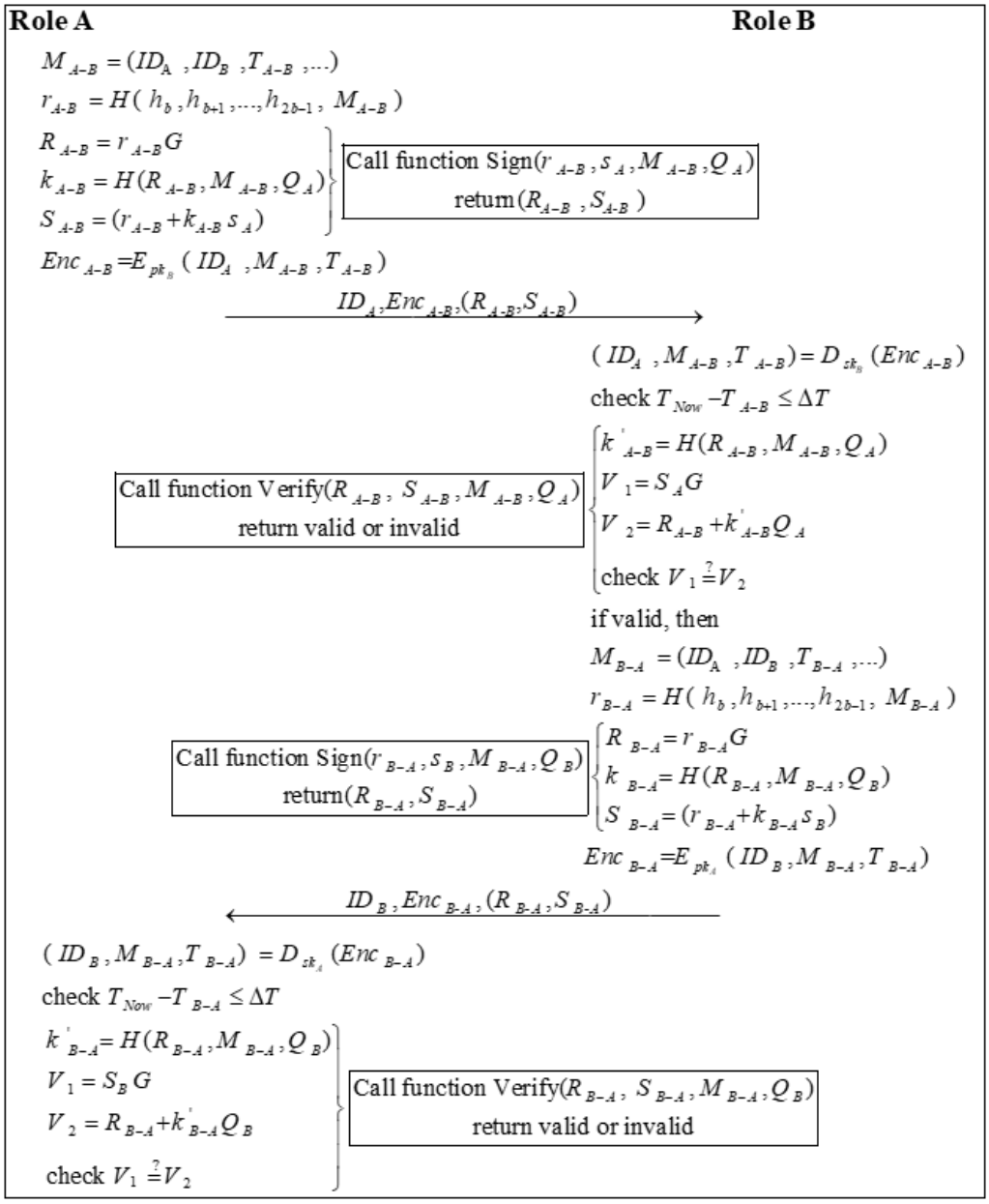

Figure 5. The process of the authentication phase.

Step 3: Firstly, role A decrypts the message Enc_{B-A} using its private key sk_A:

$$(ID_B, M_{B-A}, T_{B-A}) = D_{sk_A}(Enc_{B-A}) \tag{10}$$

Next, role A checks the validity of the timestamp:

$$T_{Now} - T_{B-A} \leq \Delta T \tag{11}$$

Finally, role A calls Algorithm 3 to verify the signature based on the public information and the received messages.

Algorithm 3: The process of EdDSA verification between role A and role B.

```
fuction Verify(String R, String S, String M, String Q){
  k = H(R, M, Q);
  V1 = S × G;
  V2 = R + k × QA;
  if V1 == V2{
     return "valid" ;
  }else{
     return"invalid";
}
```

3.6. Vaccine Purchasing Phase

In the vaccine purchasing phase, the MI first issues vaccine purchase requests to the VM. The VM produces the vaccine according to the vaccine needs of the MI after confirming the identity of the MI. When the vaccine is made, the VM needs to submit the vaccine-related information to the BC. Figure 6 describes the process of the vaccine purchasing phase.

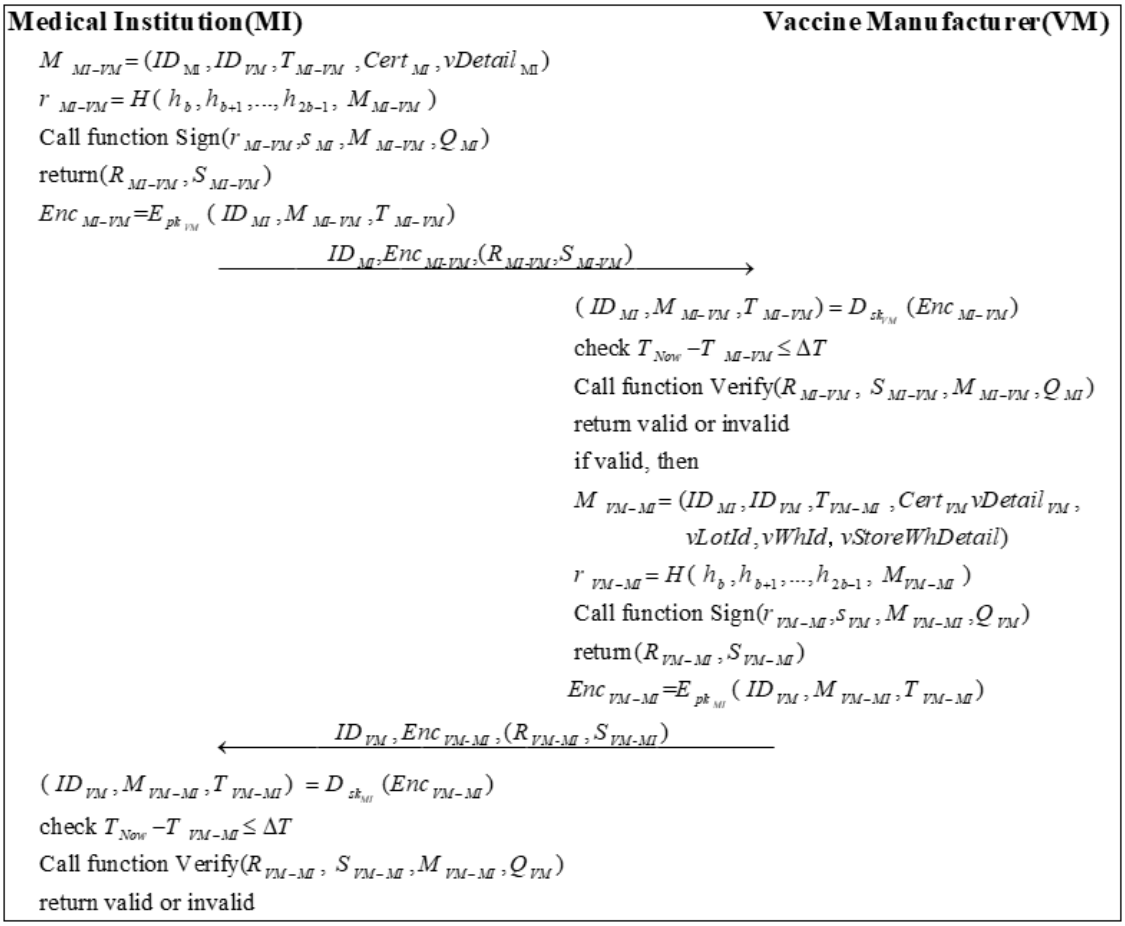

Figure 6. The process of vaccine purchase.

Step 1: MI sends a message M_{MI-VM} to VM. M_{MI-VM} needs to include the required vaccine details of the MI's $vDetail_{\mathrm{MI}}$ beside the primary information. Then, the MI calculates a random number r_{MI-VM} by encrypting M_{MI-VM} with the high b bits of the hash of the private key:

$$r_{MI-VM} = H(\, h_b, h_{b+1}, \ldots, h_{2b-1}, M_{MI-VM}) \tag{12}$$

MI calls Algorithm 2 with $(r_{MI-VM}, s_{MI}, M_{MI-VM}, Q_{MI})$ to sign the message and obtains the signature (R_{MI-VM}, S_{MI-VM}).

$$(R_{MI-VM}, S_{MI-VM}) = \mathrm{Sign}(r_{MI-VM}, s_{MI}, M_{MI-VM}, Q_{MI}) \tag{13}$$

Subsequently, MI uses the public key of the VM pk_{VM} to encrypt the message $(\, ID_{MI}, M_{MI-VM}, T_{MI-VM})$ to generate Enc_{MI-VM}:

$$Enc_{MI-VM} = E_{pk_{VM}}(\, ID_{MI}, M_{MI-VM}, T_{MI-VM}) \tag{14}$$

MI sends $ID_{MI}, Enc_{MI-VM}, (R_{MI-VM}, S_{MI-VM})$ to VM.

Step 2: After receiving the message, VM first decrypts the message Enc_{MI-VM} using its private key sk_{VM}:

$$(\, ID_{MI}, M_{MI-VM}, T_{MI-VM}) = D_{sk_{VM}}(Enc_{MI-VM}) \tag{15}$$

Then, VM checks the validity of the timestamp:

$$T_{Now} - T_{MI-VM} \leq \Delta T \tag{16}$$

Next, VM calls Algorithm 3 to verify the signature based on the public information and the messages it received.

$$\mathrm{Verify}(R_{MI-VM}, S_{MI-VM}, M_{MI-VM}, Q_{MI}) \tag{17}$$

If the signature is valid, VM sends a message M_{VM-MI} to MI. Besides the basic information, M_{VM-MI} needs to include the required vaccine details of the VM's $vDetail_{VM}$, the vaccine lot number $vLotId$, the vaccine storage warehouse number $vWhId$, and the vaccine storage warehouse details $vStoreWhDetail$. Then, VM calculates a random number r_{VM-MI}:

$$r_{VM-MI} = H(\, h_b, h_{b+1}, \ldots, h_{2b-1}, M_{VM-MI}) \tag{18}$$

VM calls Algorithm 2 with $(r_{VM-MI}, s_{VM}, M_{VM-MI}, Q_{VM})$ to sign the message and generates the signature (R_{VM-MI}, S_{VM-MI}).

$$(R_{VM-MI}, S_{VM-MI}) = \mathrm{Sign}(r_{VM-MI}, s_{VM}, M_{VM-MI}, Q_{VM}) \tag{19}$$

Later, VM encrypts the message $(\, ID_{VM}, M_{VM-MI}, T_{VM-MI})$ by using the public key of MI pk_{MI} to generate Enc_{VM-MI}:

$$Enc_{VM-MI} = E_{pk_{MI}}(\, ID_{VM}, M_{VM-MI}, T_{VM-MI}) \tag{20}$$

VM sends $ID_{VM}, Enc_{VM-MI}, (R_{VM-MI}, S_{VM-MI})$ to MI.

Step 3: First, MI decrypts the message Enc_{VM-MI} using its private key sk_{MI}:

$$(\, ID_{VM}, M_{VM-MI}, T_{VM-MI}) = D_{sk_{MI}}(Enc_{VM-MI}) \tag{21}$$

Next, MI checks the validity of the timestamp:

$$T_{Now} - T_{VM-MI} \leq \Delta T \tag{22}$$

Finally, MI calls Algorithm 3 to verify the signature based on the public information and the messages it receives.

$$\text{Verify}(R_{VM-MI}, S_{VM-MI}, M_{VM-MI}, Q_{VM}) \tag{23}$$

3.7. Vaccine Transport Phase

In the previous phase, the VM completed the vaccine according to the requirements of the MI. Once the MI confirms that the vaccine information is correct, the system enters the vaccine transport phase. The information generated by the vaccine transport process will be updated in the BC. Finally, the MI sends a message to the VM to accept the vaccine after checking that the vaccine transport is in order. Figure 7 shows the process of the vaccine transport phase.

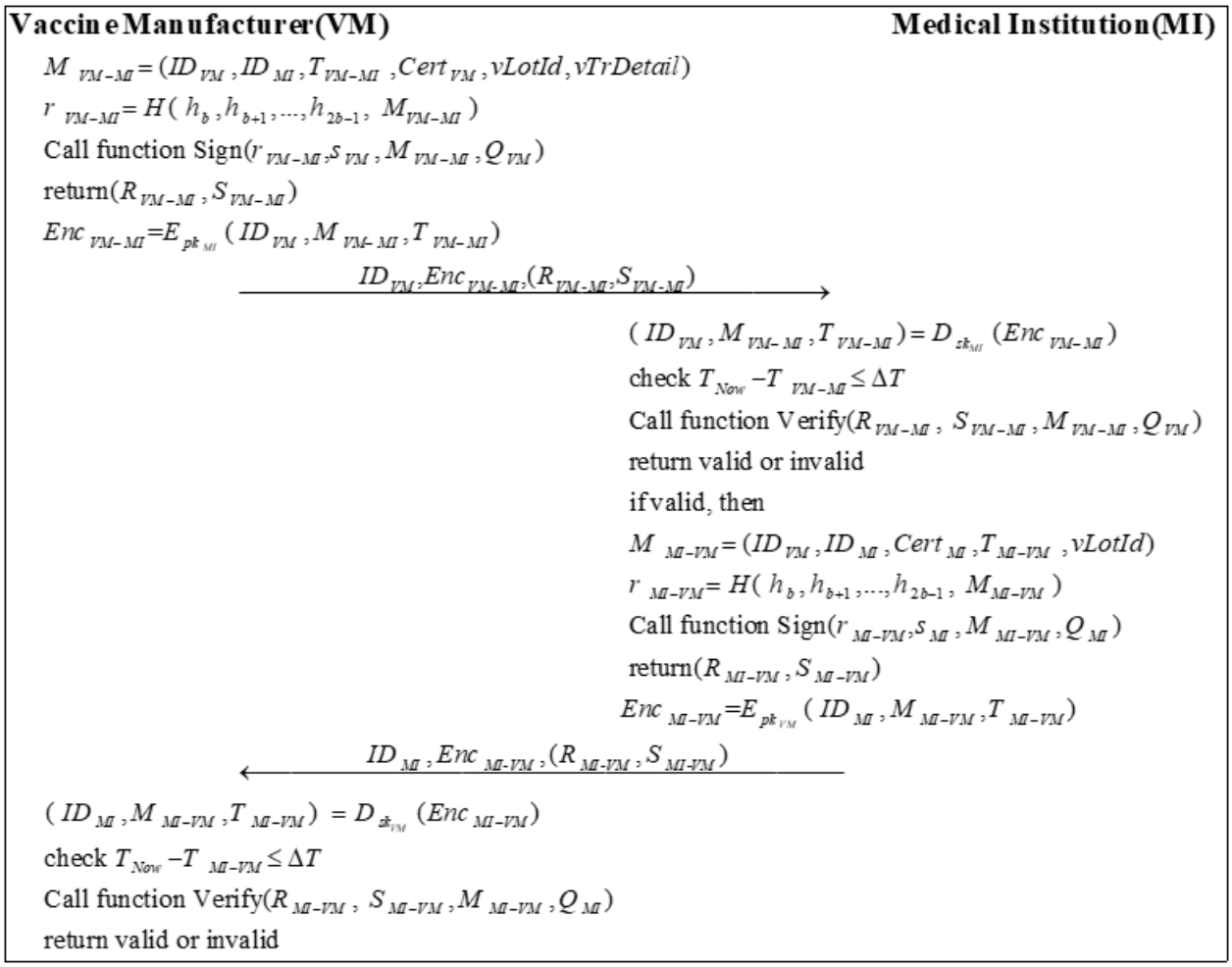

Figure 7. The process of vaccine transport.

Step 1: VM sends a message M_{VM-MI} to MI. M_{VM-MI} needs to include the vaccine lot number $vLotId$, the vaccine transport details $vTrDetail$, and the primary information. Then, MI calculates a random number r_{VM-MI} by encrypting M_{VM-MI} with the high b bits of the hash of the private key:

$$r_{VM-MI} = H(h_b, h_{b+1}, \ldots, h_{2b-1}, M_{VM-MI}) \tag{24}$$

VM calls Algorithm 2 with $(r_{MI-VM}, s_{MI}, M_{MI-VM}, Q_{MI})$ to sign the message and obtains the signature (R_{VM-MI}, S_{VM-MI}).

$$(R_{VM-MI}, S_{VM-MI}) = \text{Sign}(r_{VM-MI}, s_{VM}, M_{VM-MI}, Q_{VM}) \tag{25}$$

Subsequently, VM uses the public key of MI pk_{MI} to encrypt the message $(ID_{VM}, M_{VM-MI}, T_{VM-MI})$ to generate Enc_{VM-MI}:

$$Enc_{VM-MI} = E_{pk_{MI}}(ID_{VM}, M_{VM-MI}, T_{VM-MI}) \tag{26}$$

VM sends $ID_{MI}, Enc_{MI-VM}, (R_{MI-VM}, S_{MI-VM})$ to MI.

Step 2: After receiving the message, MI first decrypts the message Enc_{VM-MI} using its private key sk_{MI}:

$$(ID_{VM}, M_{VM-MI}, T_{VM-MI}) = D_{sk_{MI}}(Enc_{VM-MI}) \tag{27}$$

Then, MI checks the validity of the timestamp:

$$T_{Now} - T_{VM-MI} \leq \Delta T \tag{28}$$

Next, MI calls Algorithm 3 to verify the signature based on the public information and the messages it receives.

$$\text{Verify}(R_{VM-MI}, S_{VM-MI}, M_{VM-MI}, Q_{VM}) \tag{29}$$

If the signature is valid, MI sends a message M_{MI-VM} to VM. Besides the basic information, M_{MI-VM} should include the vaccine lot number $vLotId$. Then, VM calculates a random number r_{MI-VM}:

$$r_{MI-VM} = H(h_b, h_{b+1}, \ldots, h_{2b-1}, M_{MI-VM}) \tag{30}$$

MI calls Algorithm 2 with $(r_{MI-VM}, s_{MI}, M_{MI-VM}, Q_{MI})$ to sign the message and generates the signature (R_{MI-VM}, S_{MI-VM}).

$$(R_{MI-VM}, S_{MI-VM}) = \text{Sign}(r_{MI-VM}, s_{MI}, M_{MI-VM}, Q_{MI}) \tag{31}$$

Later, MI encrypts the message $(ID_{MI}, M_{MI-VM}, T_{MI-VM})$ by using the public key of the VM pk_{VM} to generate Enc_{MI-VM}:

$$Enc_{MI-VM} = E_{pk_{VM}}(ID_{MI}, M_{MI-VM}, T_{MI-VM}) \tag{32}$$

MI sends $ID_{MI}, Enc_{MI-VM}, (R_{MI-VM}, S_{MI-VM})$ to VM.

Step 3: First, VM decrypts the message Enc_{MI-VM} using its private key sk_{VM}:

$$(ID_{MI}, M_{MI-VM}, T_{MI-VM}) = D_{sk_{VM}}(Enc_{MI-VM}) \tag{33}$$

Next, VM checks the validity of the timestamp:

$$T_{Now} - T_{MI-VM} \leq \Delta T \tag{34}$$

Finally, VM calls Algorithm 3 to verify the signature based on the public information and the messages it received.

$$\text{Verify}(R_{MI-VM}, S_{MI-VM}, M_{MI-VM}, Q_{MI}) \tag{35}$$

3.8. Vaccine Distributing Phase

In this phase, the MI first needs to store the qualified vaccines received and manage them properly and strictly. Next, MI distributes the vaccines to the corresponding MP, based on the local government's vaccination requirements. Figure 8 illustrates the entire process of the vaccine-distributing phase.

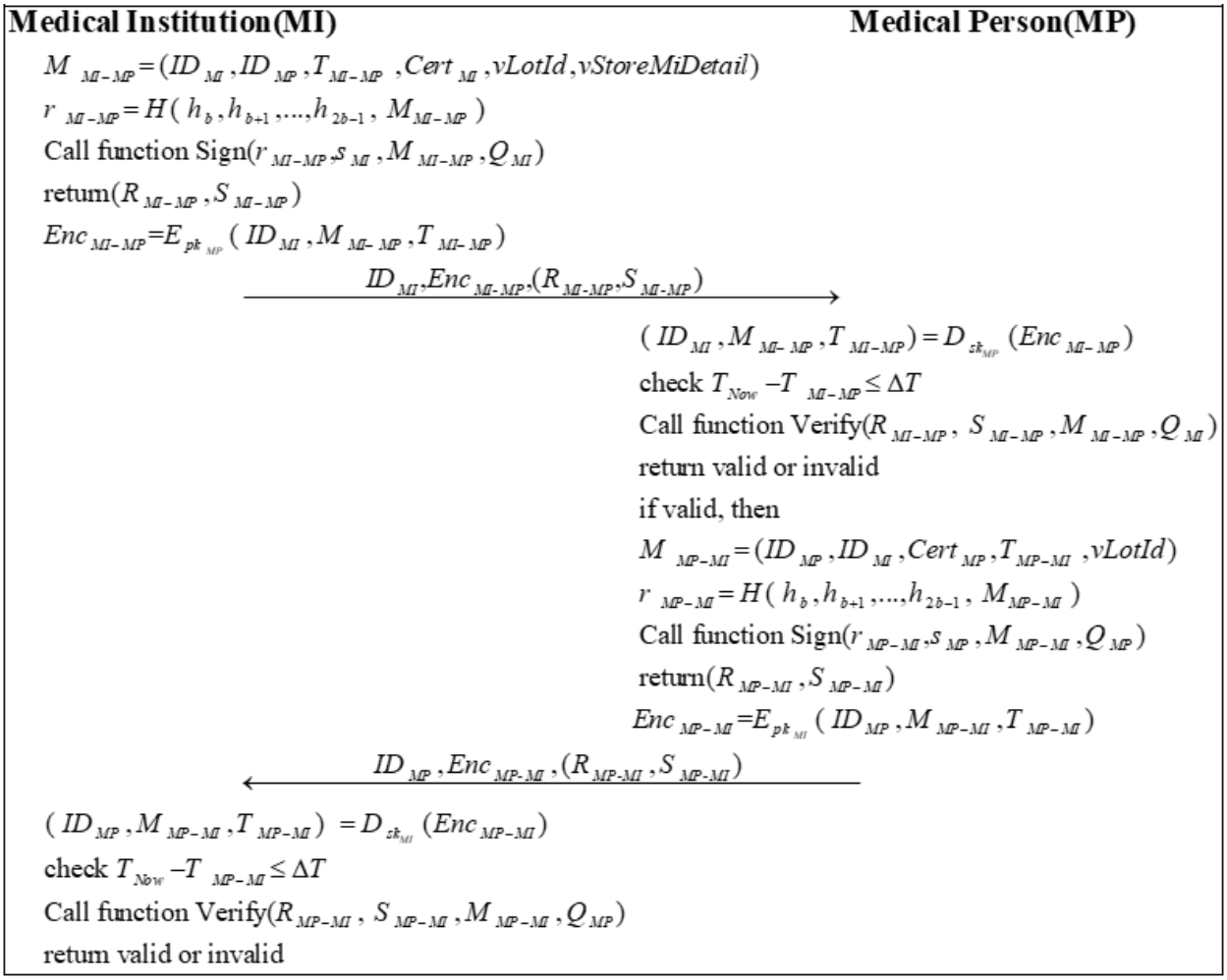

Figure 8. The process of vaccine distribution.

Step 1: MI sends a message M_{MI-MP} to MP. M_{MI-MP} needs to include the vaccine lot number $vLotId$, the vaccine storage details in MI $vStoreMiDetail$, and the primary information. Then, MI calculates a random number r_{MI-MP} by encrypting M_{MI-MP} with the high b bits of the hash of the private key:

$$r_{MI-MP}=H(h_b,h_{b+1},\ldots,h_{2b-1},M_{MI-MP}) \tag{36}$$

MI calls Algorithm 2 with $(r_{MI-MP},s_{MI},M_{MI-MP},Q_{MI})$ to sign the message and obtains the signature (R_{MI-MP},S_{MI-MP}).

$$(R_{MI-MP},S_{MI-MP})=\text{Sign}(r_{MI-MP},s_{MI},M_{MI-MP},Q_{MI}) \tag{37}$$

Subsequently, MI uses the public key of MP pk_{MP} to encrypt the message $(ID_{MI},M_{MI-MP},T_{MI-MP})$ to generate Enc_{MI-MP}:

$$Enc_{MI-MP}=E_{pk_{MP}}(ID_{MI},M_{MI-MP},T_{MI-MP}) \tag{38}$$

MI sends ID_{MI},Enc_{MI-MP},(R_{MI-MP},S_{MI-MP}) to MP.

Step 2: After receiving the message, MP first decrypts the message Enc_{MI-MP} using its private key sk_{MP}:

$$(\ ID_{MI},M_{MI-MP},T_{MI-MP}) = D_{sk_{MP}}(Enc_{MI-MP}) \tag{39}$$

Then, MP checks the validity of the timestamp:

$$T_{Now} - T_{MI-MP} \leq \Delta T \tag{40}$$

Next, MP calls Algorithm 3 to verify the signature based on the public information and the messages it receives.

$$\text{Verify}(R_{MI-MP},\ S_{MI-MP},M_{MI-MP},Q_{MI}) \tag{41}$$

If the signature is valid, MP sends a message M_{MP-MI} to MI. Besides the basic information, M_{MP-MI} should include the vaccine lot number $vLotId$. Then, MP calculates a random number r_{MP-MI}:

$$r_{MP-MI} = H(\ h_b,h_{b+1},\ldots,h_{2b-1},\ M_{MP-MI}) \tag{42}$$

MP calls Algorithm 2 with $(r_{MP-MI},s_{MP},M_{MP-MI},Q_{MP})$ to sign the message and generates the signature (R_{MP-MI},S_{MP-MI}).

$$(R_{MP-MI},S_{MP-MI}) = \text{Sign}(r_{MP-MI},s_{MP},M_{MP-MI},Q_{MP}) \tag{43}$$

Later, MP encrypts the message $(\ ID_{MP},M_{MP-MI},T_{MP-MI})$ by using the public key of MI pk_{MI} to generate Enc_{MP-MI}:

$$Enc_{MP-MI} = E_{pk_{MI}}(\ ID_{MP},M_{MP-MI},T_{MP-MI}) \tag{44}$$

MP sends ID_{MP},Enc_{MP-MI},(R_{MP-MI},S_{MP-MI}) to MI.

Step 3: First, MI decrypts the message Enc_{MP-MI} using its private key sk_{MI}:

$$(\ ID_{MP},M_{MP-MI},T_{MP-MI}) = D_{sk_{MI}}(Enc_{MP-MI}) \tag{45}$$

Next, MI checks the validity of the timestamp:

$$T_{Now} - T_{MP-MI} \leq \Delta T \tag{46}$$

Finally, MI calls Algorithm 3 to verify the signature based on the public information and the messages it receives.

$$\text{Verify}(R_{MP-MI},\ S_{MP-MI},M_{MP-MI},Q_{MP}) \tag{47}$$

3.9. Vaccination Phase

This phase mainly involves vaccination of the VP. The VP must first submit personal information and vaccination status to the MP before vaccination. The vaccination is administered only after the MP confirms that the information is correct and that the VP is medically fit to receive the vaccine. Finally, the MP will also need to update the vaccination certificate of the vaccine recipient. Figure 9 shows the process of vaccination.

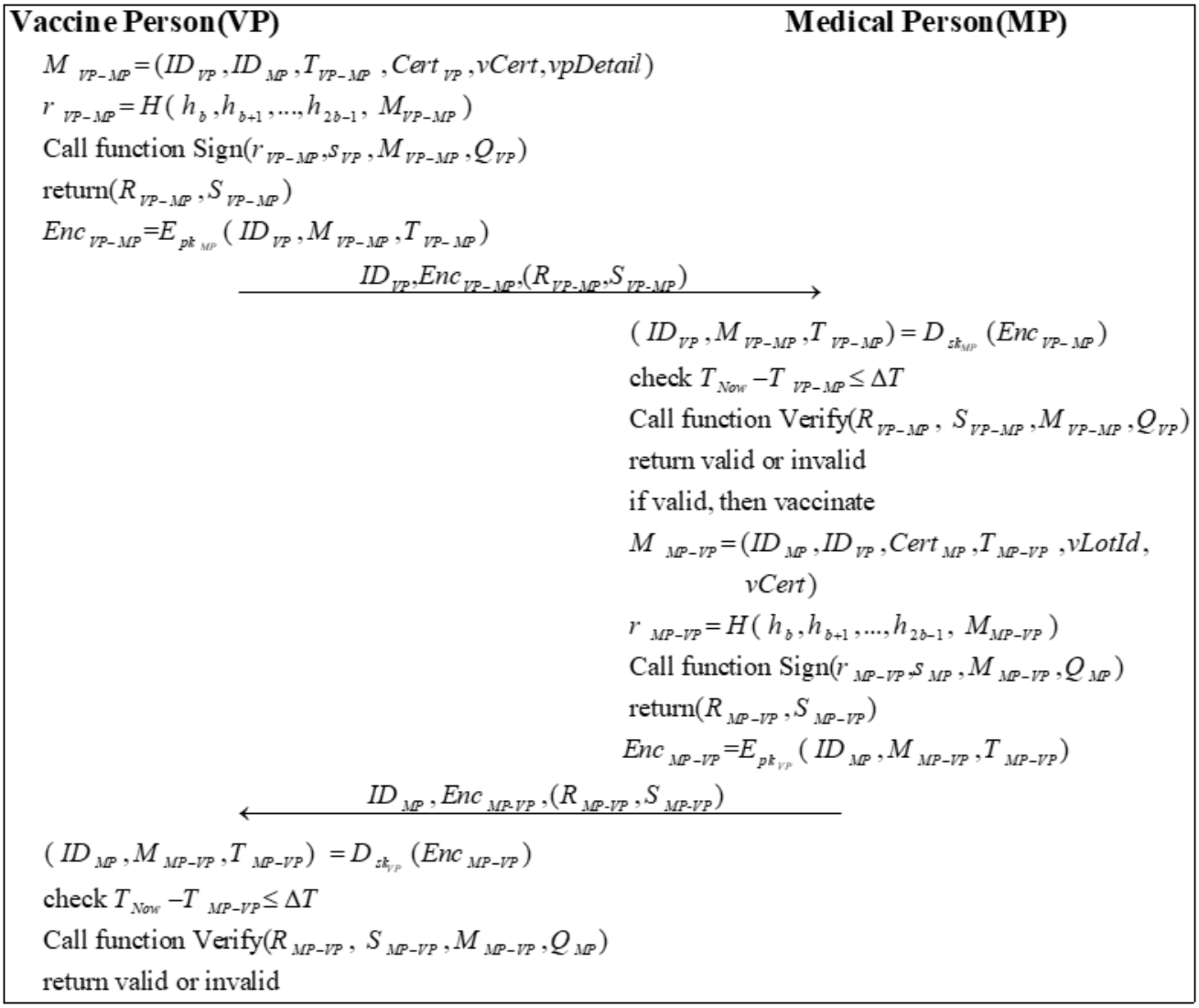

Figure 9. The process of vaccination.

Step 1: VP sends a message M_{VP-MP} to MP. M_{VP-MP} needs to include the vaccination certificate of the VP $vCert$, the VP details $vpDetail$, and the primary information. Then, VP calculates a random number r_{VP-MP} by encrypting M_{VP-MP} with the high b bits of the hash of the private key:

$$r_{VP-MP}=H(h_b,h_{b+1},\ldots,h_{2b-1},M_{VP-MP}) \tag{48}$$

VP calls Algorithm 2 with $(r_{VP-MP},s_{VP},M_{VP-MP},Q_{VP})$ to sign the message and obtains the signature (R_{VP-MP},S_{VP-MP}).

$$(R_{VP-MP},S_{VP-MP})=\text{Sign}(r_{VP-MP},s_{VP},M_{VP-MP},Q_{VP}) \tag{49}$$

Subsequently, VP uses the public key of MP pk_{MP} to encrypt the message $(ID_{VP},M_{VP-MP},T_{VP-MP})$ to generate Enc_{VP-MP}:

$$Enc_{VP-MP}=E_{pk_{MP}}(ID_{VP},M_{VP-MP},T_{VP-MP}) \tag{50}$$

VP sends $ID_{VP},Enc_{VP-MP},(R_{VP-MP},S_{VP-MP})$ to MP.

Step 2: After receiving the message, MP first decrypts the message Enc_{VP-MP} using its private key sk_{MP}:

$$(ID_{VP}, M_{VP-MP}, T_{VP-MP}) = D_{sk_{MP}}(Enc_{VP-MP}) \tag{51}$$

Then, MP checks the validity of the timestamp:

$$T_{Now} - T_{VP-MP} \leq \Delta T \tag{52}$$

Next, MP calls Algorithm 3 to verify the signature based on the public information and the messages it receives.

$$\text{Verify}(R_{VP-MP}, S_{VP-MP}, M_{VP-MP}, Q_{VP}) \tag{53}$$

If the signature is valid, MP vaccinates VP. Afterward, MP sends a message M_{MP-VP} to VP. Besides the basic information, M_{MP-VP} should include the vaccine lot number $vLotId$ and the vaccination certificate of the VP $vCert$. Then, MP calculates a random number r_{MP-VP}:

$$r_{MP-VP} = H(h_b, h_{b+1}, \ldots, h_{2b-1}, M_{MP-VP}) \tag{54}$$

MP calls Algorithm 2 with $(r_{MP-VP}, s_{MP}, M_{MP-VP}, Q_{MP})$ to sign the message and generates the signature (R_{MP-VP}, S_{MP-VP}).

$$(R_{MP-VP}, S_{MP-VP}) = \text{Sign}(r_{MP-VP}, s_{MP}, M_{MP-VP}, Q_{MP}) \tag{55}$$

Later, MP encrypts the message $(ID_{MP}, M_{MP-VP}, T_{MP-VP})$ by using the public key of VP pk_{VP} to generate Enc_{MP-VP}:

$$Enc_{MP-VP} = E_{pk_{VP}}(ID_{MP}, M_{MP-VP}, T_{MP-VP}) \tag{56}$$

MP sends $ID_{MP}, Enc_{MP-VP}, (R_{MP-VP}, S_{MP-VP})$ to VP.

Step 3: First, VP decrypts the message Enc_{MP-VP} using its private key sk_{VP}:

$$(ID_{MP}, M_{MP-VP}, T_{MP-VP}) = D_{sk_{VP}}(Enc_{MP-VP}) \tag{57}$$

Next, VP checks the validity of the timestamp:

$$T_{Now} - T_{MP-VP} \leq \Delta T \tag{58}$$

Finally, VP calls Algorithm 3 to verify the signature based on the public information and the messages it received.

$$\text{Verify}(R_{MP-VP}, S_{MP-VP}, M_{MP-VP}, Q_{MP}) \tag{59}$$

3.10. Side Effect Phase

If a VP has an adverse reaction after receiving the vaccine, the VP is required to upload the appropriate information to the BC, including details of the adverse reaction, personal information, and proof of vaccination. If the adverse reaction is judged to be a side effect of the vaccination, the vaccine lot ID will be returned to the VP, and the appropriate treatment will be provided. Figure 10 shows the process of the side-effect phase.

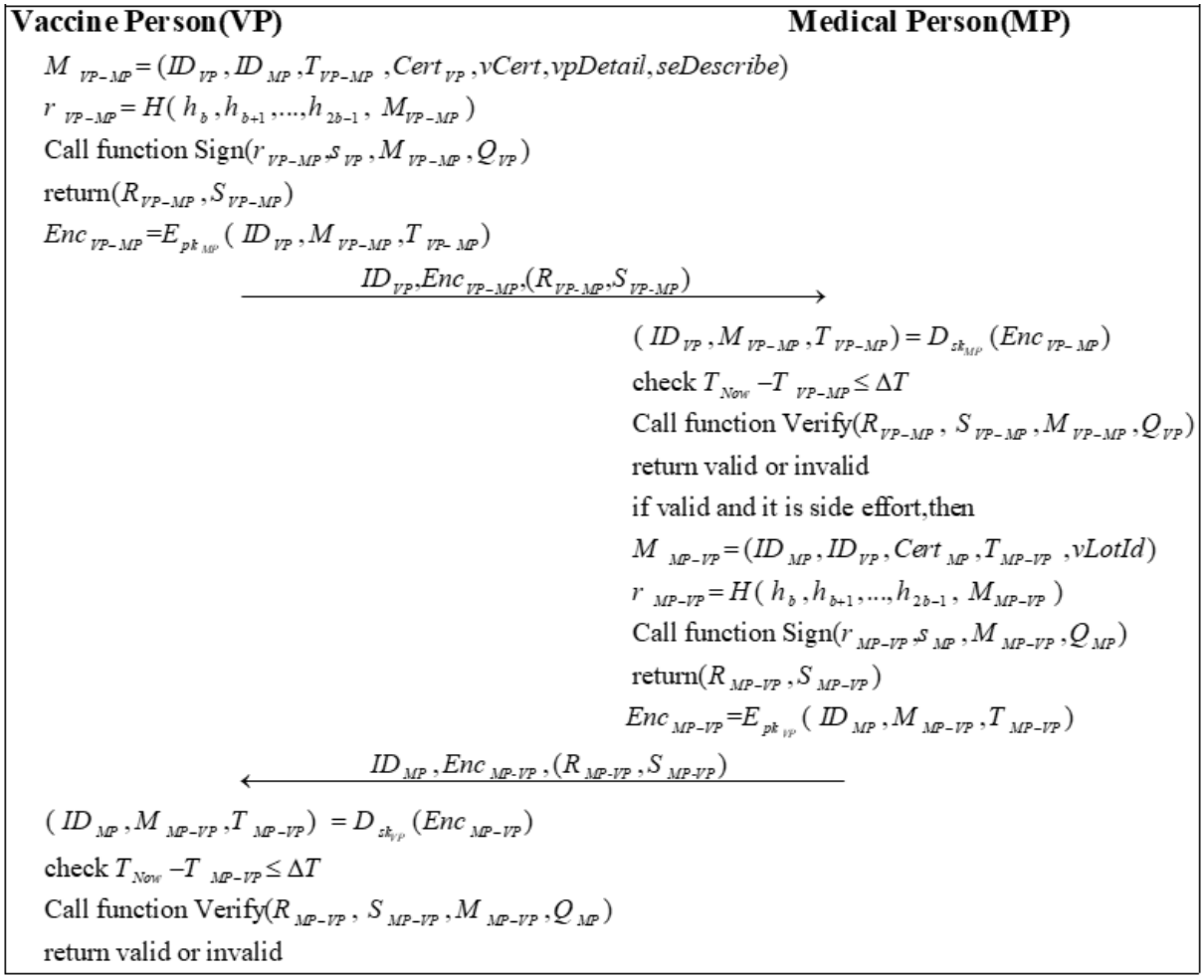

Figure 10. The process of the side effects submitted.

Step 1: VP sends a message M_{VP-MP} to MP. M_{VP-MP} needs to include the vaccination certificate of VP *vCert*, the VP details *vpDetail*, the description of side effects *seDescribe*, and the primary information. Then, VP calculates a random number r_{VP-MP} by encrypting M_{VP-MP} with the high b bits of the hash of the private key:

$$r_{VP-MP}=H(h_b,h_{b+1},\ldots,h_{2b-1},M_{VP-MP}) \tag{60}$$

VP calls Algorithm 2 with $(r_{VP-MP},s_{VP},M_{VP-MP},Q_{VP})$ to sign the message and obtains the signature (R_{VP-MP},S_{VP-MP}).

$$(R_{VP-MP},S_{VP-MP})=\text{Sign}(r_{VP-MP},s_{VP},M_{VP-MP},Q_{VP}) \tag{61}$$

Subsequently, VP uses the public key of MP pk_{MP} to encrypt the message $(ID_{VP},M_{VP-MP},T_{VP-MP})$ to generate Enc_{VP-MP}:

$$Enc_{VP-MP}=E_{pk_{MP}}(ID_{VP},M_{VP-MP},T_{VP-MP}) \tag{62}$$

VP sends $ID_{VP},Enc_{VP-MP},(R_{VP-MP},S_{VP-MP})$ to MP.

Step 2: After receiving the message, MP first decrypts the message Enc_{VP-MP} using its private key sk_{MP}:

$$(ID_{VP},M_{VP-MP},T_{VP-MP})=D_{sk_{MP}}(Enc_{VP-MP}) \tag{63}$$

Then, MP checks the validity of the timestamp:

$$T_{Now} - T_{VP-MP} \leq \Delta T \tag{64}$$

Next, MP calls Algorithm 3 to verify the signature based on the public information and the messages it receives.

$$\text{Verify}(R_{VP-MP}, S_{VP-MP}, M_{VP-MP}, Q_{VP}) \tag{65}$$

If the signature is valid and MP considers the adverse reaction to being a side effect of the vaccination, MP sends a message M_{MP-VP} to VP. In addition to the basic information, M_{MP-VP} should include the vaccine lot number. Then, MP calculates a random number r_{MP-VP}:

$$r_{MP-VP} = H(h_b, h_{b+1}, \ldots, h_{2b-1}, M_{MP-VP}) \tag{66}$$

MP calls Algorithm 2 to sign the message $(r_{MP-VP}, s_{MP}, M_{MP-VP}, Q_{MP})$ and generates the signature (R_{MP-VP}, S_{MP-VP}).

$$(R_{MP-VP}, S_{MP-VP}) = \text{Sign}(r_{MP-VP}, s_{MP}, M_{MP-VP}, Q_{MP}) \tag{67}$$

Later, MP encrypts the message $(ID_{MP}, M_{MP-VP}, T_{MP-VP})$ by using the public key of VP pk_{VP} to generate Enc_{MP-VP}:

$$Enc_{MP-VP} = E_{pk_{VP}}(ID_{MP}, M_{MP-VP}, T_{MP-VP}) \tag{68}$$

MP sends $ID_{MP}, Enc_{MP-VP}, (R_{MP-VP}, S_{MP-VP})$ to VP.

Step 3: First, VP decrypts the message Enc_{MP-VP} using its private key sk_{VP}:

$$(ID_{MP}, M_{MP-VP}, T_{MP-VP}) = D_{sk_{VP}}(Enc_{MP-VP}) \tag{69}$$

Next, VP checks the validity of the timestamp:

$$T_{Now} - T_{MP-VP} \leq \Delta T \tag{70}$$

Finally, VP calls Algorithm 3 to verify the signature based on the public information and the messages it received.

$$\text{Verify}(R_{MP-VP}, S_{MP-VP}, M_{MP-VP}, Q_{MP}) \tag{71}$$

4. Security Analysis

4.1. Mutual Authentication

The proposed scheme uses BAN logic to achieve mutual authentication between role A and role B. The scheme of role A and role B can represent the blockchain center (BC), the vaccine manufacturer (VM), the medical institution (MI), the medical personnel (MP), and the vaccinated person (VP). The notation of BAN logic is shown below.

$P \mid\equiv X$	P believes X
$P \lhd X$	P sees X
$P \mid\sim X$	P said X
$P \mid\Rightarrow X$	P controls X
$\#(X)$	The message X is fresh
$P \overset{K}{\leftrightarrow} Q$	P and Q communicate with a shared key K
$\{X\}_K$	X is encrypted with a key K

The goals of the entire authentication process are as follows:

$$G1: A\left|\equiv A \overset{X_A}{\leftrightarrow} B\right.$$
$$G2: A\left|\equiv B\right|\equiv A \overset{X_A}{\leftrightarrow} B$$
$$G3: B\left|\equiv A \overset{X_B}{\leftrightarrow} B\right.$$
$$G4: B\left|\equiv A\right|\equiv A \overset{X_B}{\leftrightarrow} B$$
$$G5: A|\equiv ID_B$$
$$G6: A|\equiv B|\equiv ID_B$$
$$G7: B|\equiv ID_A$$
$$G8: B|\equiv A|\equiv ID_A$$

Depending on the authentication process, BAN logic generates the following idealized model:

$$M1: Role\ A \rightarrow Role\ B(\{ID_A, ID_B, T_{A-B}\}_{PK_B}, R_A, R_A)$$
$$M2: Role\ B \rightarrow Role\ A(\{ID_A, ID_B, T_{B-A}\}_{PK_A}, R_B, R_B)$$

To analyze the proposed scheme, we make the following assumptions:

$$A1: A|\equiv \#(T_{A-B})$$
$$A2: B|\equiv \#(T_{A-B})$$
$$A3: A|\equiv \#(T_{B-A})$$
$$A4: B|\equiv \#(T_{B-A})$$
$$A5: A\left|\equiv B\right|\Rightarrow B \overset{X_B}{\leftrightarrow} A$$
$$A6: B\left|\equiv A\right|\Rightarrow A \overset{X_A}{\leftrightarrow} B$$
$$A7: A|\equiv B|\Rightarrow ID_B$$
$$A8: B|\equiv A|\Rightarrow ID_A$$

Based on the rules of BAN logic and the assumptions above, the authentication process between the two nodes is shown below:

a. Role B authenticates role A.
The statement $S1$ can be derived from $M1$ the seeing rule:
$S1: B \lhd (\{ID_A, ID_B, T_{A-B}\}_{PK_B}, R_A, R_A)$
The statement $S2$ can be derived from $A2$ and the freshness rule:
$S2: B\left|\equiv \#(\{ID_A, ID_B, T_{A-B}\}_{PK_B}, R_A, R_A)\right.$
The statement $S3$ can be derived from $S1$, $A4$ and the message meaning rule:
$S3: B|\equiv A|\sim (ID_A, ID_B, T_{A-B}, R_A, R_A)$
The statement $S4$ can be derived by $S2$, $S3$ and the nonce verification rule:
$S4: B|\equiv A|\equiv (ID_A, ID_B, T_{A-B}, R_A, R_A)$
The statement $S5$ can be derived from $S4$ and the belief rule:
$S5: B\left|\equiv A\right|\equiv A \overset{X_A}{\leftrightarrow} B$
The statement $S6$ can be derived from $S5$, $A6$ and the jurisdiction rule:
$S6: B\left|\equiv A \overset{X_A}{\leftrightarrow} B\right.$
The statement $S7$ can be derived from $S4$ and the belief rule:
$S7: B|\equiv A|\equiv ID_A$
The statement $S8$ can be derived from $S7$, $A8$ and the belief rule:
$S8: B|\equiv ID_A$

b. Role A authenticates role B.
The statement $S9$ can be derived from $M2$ and the seeing rule:
$S9: A \lhd (\{ID_A, ID_B, T_{B-A}\}_{PK_A}, R_B, R_B)$
The statement $S10$ can be derived from $A1$ and the freshness rule:
$S10: A\left|\equiv \#(\{ID_A, ID_B, T_{B-A}\}_{PK_A}, R_B, R_B)\right.$

The statement $S11$ can be derived from $S9$, $A3$ and the message meaning rule:
$S11 : A|\equiv B|\sim (ID_A, ID_B, T_{B-A}, R_B, R_B)$
The statement $S12$ can be derived by $S10$,$S11$ and the nonce verification rule:
$S12 : A|\equiv B|\equiv (ID_A, ID_B, T_{B-A}, R_B, R_B)$
The statement $S13$ can be derived from $S12$ and the belief rule:
$S13 : A\left|\equiv B\right|\equiv B \overset{X_B}{\leftrightarrow} A$
The statement $S14$ can be derived from $S13$,$A5$ and the jurisdiction rule:
$S14 : A\left|\equiv B \overset{X_B}{\leftrightarrow} A\right.$
The statement $S15$ can be derived from $S12$ and the belief rule:
$S15 : A|\equiv B|\equiv ID_B$
The statement $S16$ can be derived from $S15$,$A7$ and the belief rule:
$S16 : A|\equiv ID_{\mathrm{B}}$

With Statement $S6$, $S8$, $S14$, $S16$, role A and role B can easily verify the identity of each other when passing messages.

4.2. Decentralization and Information Sharing

The essence of blockchain technology is a distributed ledger. In the proposed scheme, all registered nodes jointly maintain the entire vaccine information management system, and any information has to be uploaded to the chain through the consensus mechanism of the system. Meanwhile, the failure of a single node does not cause the whole system to break down. Moreover, the information uploaded to the blockchain requires the sender to use its private key for signature, and the information on the chain can be viewed by other registered nodes. These features not only ensure the safety and reliability of the uploaded information but also ensure the openness and transparency of this information and realize the trust relationship between unfamiliar nodes.

4.3. Traceable

Messages sent in a blockchain system should be accompanied by using Algorithm 2. This message, if proven to be valid, is permanently stored in the blockchain and cannot be tampered with. Therefore, other nodes in the blockchain can trace the message and guarantee the validity of the message by using Algorithm 3. In this way, the traceability of the system is achieved.

4.4. High-Quality Random Number

The security of digital signature algorithms, such as DSA and ECDSA, relies on high-quality random number generators to generate random numbers. Once the quality of the random numbers is not up to par, the information of the system users will also be compromised. The random number generation of the EdDSA algorithm is shown in Equations (4), (8), (12), (18), (24), (36), (42), (48), (54), (60) and (66).

The generation of a random number of EdDSAs relies on the user's private key with the delivered message itself. This random number is naturally of high quality, which pretty much eliminates the problem of information leakage caused by the quality of the random number.

4.5. Integrity and Non-Repudiation

When two nodes communicate, they are very concerned about the integrity of the transmitted message. In our scheme, the EdDSA algorithm is used to generate the signature. The sender generates a specific signature when sending a message based on random numbers, message content, and other parameters. Any tampering with the parameters will change the original signature, and the original message cannot be inferred from the signature string.

Meanwhile, the sender signs the message with its private key when sending it, and the receiver will use the sender's public key to verify the signature when receiving the message. Therefore, the sender cannot deny the message it sent.

The signature in Table 1 describes the data integrity proof for each stage, and the verification describes the non-repudiation proof for each stage.

Table 1. Verification of integrity and non-repudiation in the proposed scheme.

Phase	Sender	Receiver	Signature	Verification
Authentication Phase	A	B	$\mathrm{Sign}(r_{A-B}, s_A, M_{A-B}, Q_A)$	$\mathrm{Verify}(R_{A-B}, S_{A-B}, M_{A-B}, Q_A)$
	B	A	$\mathrm{Sign}(r_{B-A}, s_B, M_{B-A}, Q_B)$	$\mathrm{Verify}(R_{B-A}, S_{B-A}, M_{B-A}, Q_B)$
Vaccine Purchasing Phase	MI	VM	$\mathrm{Sign}(r_{MI-VM}, s_{MI}, M_{MI-VM}, Q_{MI})$	$\mathrm{Verify}(R_{MI-VM}, S_{MI-VM}, M_{MI-VM}, Q_{MI})$
	VM	MI	$\mathrm{Sign}(r_{VM-MI}, s_{VM}, M_{VM-MI}, Q_{VM})$	$\mathrm{Verify}(R_{VM-MI}, S_{VM-MI}, M_{VM-MI}, Q_{VM})$
Vaccine transport Phase	VM	MI	$\mathrm{Sign}(r_{VM-MI}, s_{VM}, M_{VM-MI}, Q_{VM})$	$\mathrm{Verify}(R_{MI-MP}, S_{MI-MP}, M_{MI-MP}, Q_{MI})$
	MI	VM	$\mathrm{Sign}(r_{MI-VM}, s_{MI}, M_{MI-VM}, Q_{MI})$	$\mathrm{Verify}(R_{MP-MI}, S_{MP-MI}, M_{MP-MI}, Q_{MP})$
Vaccine Distributing Phase	MI	MP	$\mathrm{Sign}(r_{MI-MP}, s_{MI}, M_{MI-MP}, Q_{MI})$	$\mathrm{Verify}(R_{MI-MP}, S_{MI-MP}, M_{MI-MP}, Q_{MI})$
	MP	MI	$\mathrm{Sign}(r_{MP-MI}, s_{MP}, M_{MP-MI}, Q_{MP})$	$\mathrm{Verify}(R_{MP-MI}, S_{MP-MI}, M_{MP-MI}, Q_{MP})$
Vaccination Phase	VP	MP	$\mathrm{Sign}(r_{VP-MP}, s_{VP}, M_{VP-MP}, Q_{VP})$	$\mathrm{Verify}(R_{VP-MP}, S_{VP-MP}, M_{VP-MP}, Q_{VP})$
	MP	VP	$\mathrm{Sign}(r_{MP-VP}, s_{MP}, M_{MP-VP}, Q_{MP})$	$\mathrm{Verify}(R_{MP-VP}, S_{MP-VP}, M_{MP-VP}, Q_{MP})$
Side Effect Phase	VP	MP	$\mathrm{Sign}(r_{VP-MP}, s_{VP}, M_{VP-MP}, Q_{VP})$	$\mathrm{Verify}(R_{VP-MP}, S_{VP-MP}, M_{VP-MP}, Q_{VP})$
	MP	VP	$\mathrm{Sign}(r_{MP-VP}, s_{MP}, M_{MP-VP}, Q_{MP})$	$\mathrm{Verify}(R_{MP-VP}, S_{MP-VP}, M_{MP-VP}, Q_{MP})$

4.6. Man-in-the-Middle Attacks

To avoid this potential risk, the proposed scheme requires the sender to encrypt the message with the public key of the receiver before sending it to the receiver. Therefore, only the receiver can decrypt the message with its private key and obtain the message content. Thus, it solves the problem of man-in-the-middle attacks that may exist in the system. The encryption process for the message is shown in Equations (5), (6), (9), (10), (14), (15), (20), (21), (26), (27), (32), (33), (38), (39), (44), (45), (50), (51), (56), (57), (62), (63), (68) and (69).

Scenario: The sender sends a message to the receiver. Before the message is delivered, the malicious attacker eavesdrops and modifies the message.

Analysis: The sender encrypts the message with the receiver's public key when sending the message. The malicious attacker does not have the receiver's private key, so the attacker cannot decrypt the exact contents of the message.

4.7. Replay Attack

To avoid this potential risk, the proposed scheme requires a timestamp to be attached to the message when it is passed between users. Both the timestamp and the message content are encrypted by the sender using the public key of the receiver, so the timestamp can only be decrypted and obtained by the receiver using his private key. If a malicious attacker sends the same message to the receiver later, the system will compare the difference between the current time and the message timestamp. Then, if the difference is greater than the threshold, the message is illegal. Therefore, the problem of replay attacks is eliminated. The specific process is shown as follows:

$$Enc_{A-B} = E_{pk_B}(ID_A, M_{A-B}, T_{A-B}) \tag{72}$$

$$(ID_A, M_{A-B}, T_{A-B}) = D_{sk_B}(Enc_{A-B}) \tag{73}$$

$$\text{check } T_{Now} - T_{A-B} \leq \Delta T \tag{74}$$

Scenario: The malicious attacker sends an identical message to the receiver after listening to the message sent by the sender.

Analysis: The receiver decrypts the message with its private key, obtains the corresponding timestamp, and compares the difference between the current time and the timestamp with the threshold. If the difference is greater than the queue value, the system determines that it is a replay attack and rejects the message.

4.8. Sybil Attack

To avoid this potential risk, blockchain can use the consensus mechanism to increase the entry barrier of nodes. At this point, the high cost makes the Sybil attack unrealistic because the malicious attacker must occupy more than half of the nodes of the whole system. In addition, the proposed scheme requires each user to obtain the corresponding ID number and EdDSA public and private key pairs in the registration phase, and all users entering the system must pass identity validation. The parameters related to the user's identity are generated by the blockchain center using Equations (1)–(3). Then, the user stores these parameters.

$$\text{Stores}(\ ID_X,\ d_X,\ Q_X, pk_X, sk_X, Cert_X) \tag{75}$$

Scenario: The malicious nodes attempt to forge vast numbers of fake identities to access the blockchain system.

Analysis: Every ID number is generated by the blockchain center with the corresponding and unique public and private key pairs and certificates. The attacker has no chance of obtaining the complete parameters, and its operations in the system are considered invalid. Thus, the Sybil attack is hardly successful.

5. Discussion

5.1. Computation Cost

In this section, we analyzed the performance of the system. Table 2 presents the performance analysis of each phase. We used asymmetrical encryption/decryption, hash functions, addition, subtraction, multiplication, and division operations as the basis for calculating the costs.

Table 2. The computation cost of each phase.

Phase	1st Party	2nd Party
Authentication Phase	Role A: $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$	Role B: $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$
Vaccine Purchasing Phase	MI : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$	VM : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$
Vaccine Transport Phase	VM : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$	MI : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$
Vaccine Distributing Phase	MI : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$	MP : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$
Vaccination Phase	VP : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$	MP : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$
Side Effect Phase	VP : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$	MP : $2T_{asy} + 2T_h + 2T_{add} + T_{sub} + 4T_{mul} + 2T_{com}$

Notes: T_{asy}: asymmetrical encryption/decryption; T_h: a hash operation; T_{add}: an additional operation; T_{sub}: a subtraction operation. T_{mul}: a multiplication operation; T_{com}: the time required for a comparison operation.

5.2. Comparison

In this section, we compare the proposed scheme with previous articles dealing with vaccine record security in Table 3.

Table 3. Comparison of the proposed and other vaccine-related articles.

Authors	Year	Objective	1	2	3	4	5	6
Sigwart et al. [27]	2019	Proposed the feasibility of blockchain application in the vaccine supply chain.	Y	N	Y	N	N	N
Yong et al. [28]	2019	Proposed to use of blockchain and machine learning to ensure vaccine safety.	Y	Y	Y	N	N	N
Deka et al. [30]	2020	Proposed to use of blockchain and IPFS to maintain personal vaccination records.	Y	Y	Y	N	N	N
Antal et al. [31]	2021	Proposed to use of smart contracts to monitor COVID-19 vaccine supply management.	Y	Y	N	N	N	Y
Chauhan et al. [32]	2021	Proposed to use of blockchain to ensure transparency and anti-counterfeiting of the COVID-19 vaccine.	Y	Y	N	N	N	Y
Chen et al. [33]	2022	Proposed a traceable blockchain-based vaccination record storage and share system.	Y	Y	Y	Y	Y	N
Our scheme	2022	Propose a blockchain-based traceable vaccine system.	Y	Y	Y	Y	Y	Y

Notes: 1: based on blockchain, 2: proposed system architecture, 3: used in multiple vaccines, 4: Mutual authentication, 5: security analysis, 6: Involved in the vaccine supply chain. Y: yes, N: no.

5.3. Performance Analysis

In this section, we perform a performance analysis of the proposed scheme using a caliper, a blockchain performance testing framework that allows its users to test the system on different blockchain platforms and obtain the corresponding performance test results. All tests were run in the following environment: Intel(R) Core (TM) i7-7700HQ CPU @ 2.80GHz, 8GB RAM. We use Fabric 2.0.0 and Go 1.17.5. The operating system is Ubuntu 18.04.5 LTS.

Due to the different communication protocols of the schemes in each article, it is hard to find suitable papers to compare their performances. Therefore, we only base our scheme on analyzing its performance. The performance of blockchain systems is usually evaluated in terms of throughput and latency. The throughput refers to the speed at which transactions are added to the ledger and represents the performance level of the system, which is expressed in transactions per second (TPS) at the testing time. Latency is an indicator of the time spent between the application initiating a transaction time and the received time. This is the first thing that users care about when using a blockchain system.

Figure 11 shows the relationship between throughput and send rate. Ten sets of data were selected for the test, and the difference between the send rates of each set was 50 tps. With a fixed system block size, we can find that the throughput of read transactions is approximately linearly related to the send rate, ranging from a low of 50.1 tps to a high of 447.3 tps. The throughput of write transactions is positively correlated with the send rate, which grows gently from a low of 46.8 tps to a high of 91.2 tps. However, we can notice that the relative increase in throughput slows down after the send rate gradually increases, which could be approaching the threshold. Figure 12 shows the relationship between latency and the send rate. We can see that with fixed block size, latency is positively correlated with send rate, ranging from a minimum of 0.05 s to a maximum of 1.41 s for reading transactions, and from a minimum of 0.16 s to a maximum of 12.74 s for write transactions. Similarly, after the send rate reaches 400 tps, the latency growth rate becomes flatter, indicating that the system may have reached the threshold. Thus, the proposed system has adequate performance to read and write vaccine-related data. Meanwhile, users can access vaccine-related data in a short enough time and can modify these data according to their rights.

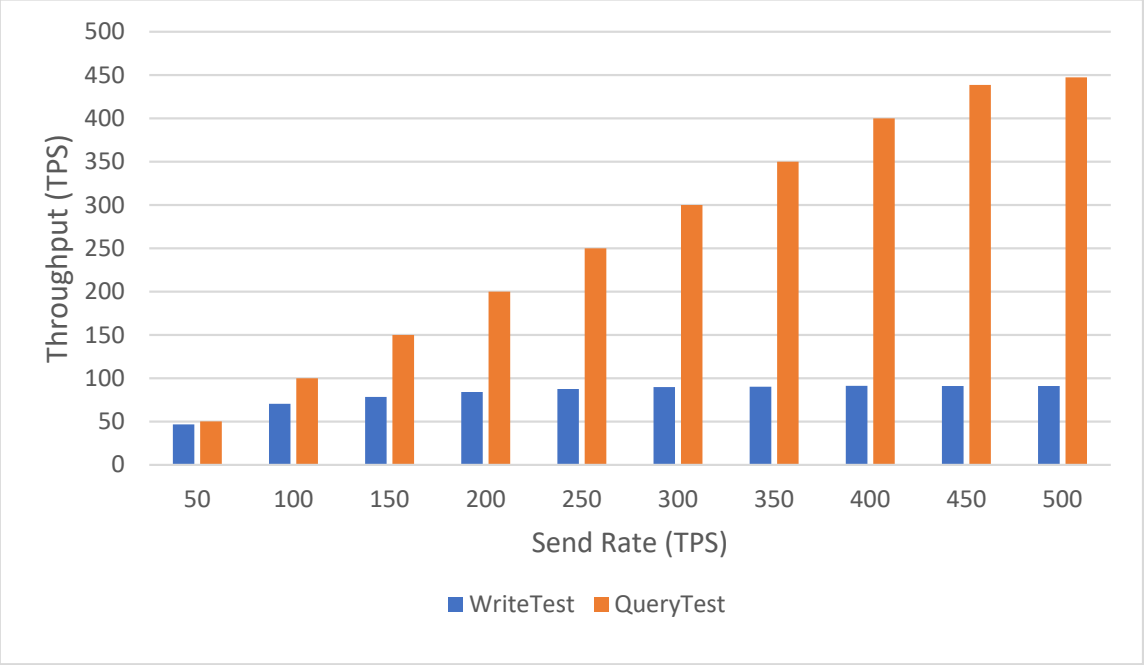

Figure 11. Throughput of the system at various send rates.

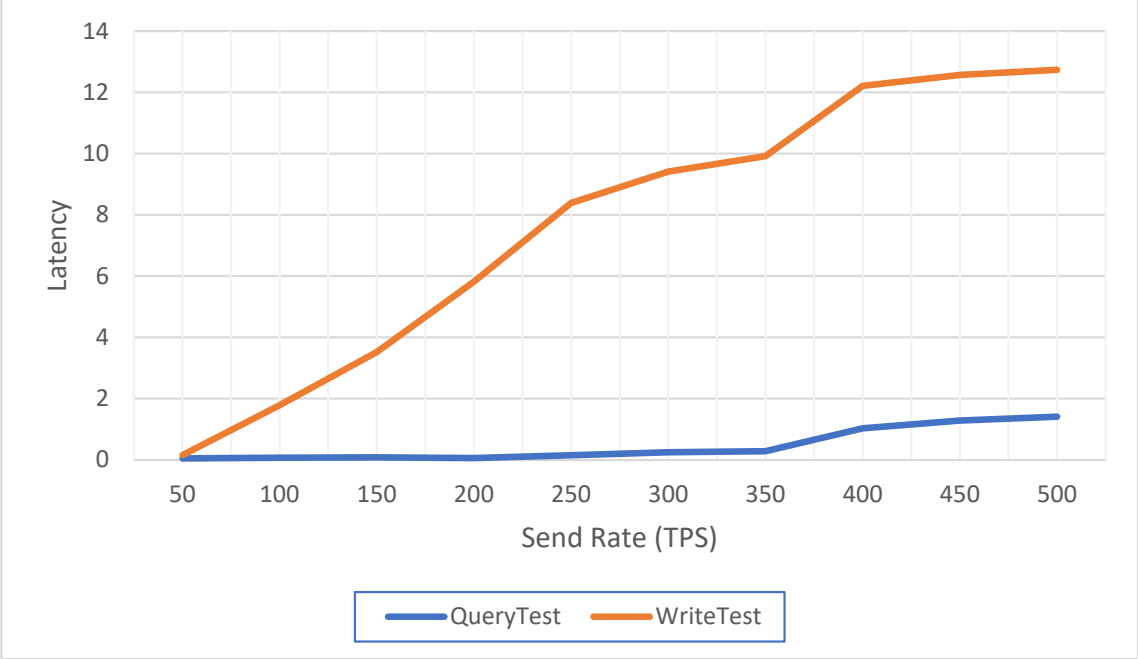

Figure 12. Latency of the system at various send rates.

5.4. Comparison of Blockchain Platforms

Blockchain technology can be divided into three categories in total after development: public blockchain, private blockchain, and consortium blockchain. Private blockchain operation is centralized, and its node verification is usually operated by a single group only, which is not conducive to data sharing and traceability. Furthermore, the private blockchain is more susceptible to data tampering by unscrupulous elements, so they are not considered in this paper. Table 4 shows a comparison of the three blockchain platforms.

Table 4. Comparison of the three blockchain platforms.

Characteristics	Bitcoin	Ethereum	Hyperledger Fabric
Type	Public blockchain	Public blockchain	Consortium blockchain
Consensus	Proof of work (POW)	Proof of work (POW)	PBFT
Scripting	Limited stack-based scripting	Solidity	Go, Java, JavaScript
Authentication	No	No	Yes
Smart Contract	No	No	Yes
Scalability	Low scalability	Low scalability	High scalability
Currency	Bitcoin	Ether	No
Speed of transactions	7 TPS	20 TPS	1000 TPS-10000 TPS

Compared to the public blockchain, consortium blockchain share information selectively, and nodes without permission cannot access the corresponding data. It can better protect the privacy of sensitive data, such as vaccine information. Moreover, Hyperledger Fabric has better performance and higher scalability. It also allows smart contracts to be written in multiple programming languages, making it easy to develop the system.

6. Conclusions

This paper proposes a vaccine record management system based on blockchain technology. The system protects the privacy of sensitive vaccine information while making the entire process of vaccine production, distribution, and vaccination transparent and traceable. It reduces the possibility of tampering with vaccine information and indirectly prevents vaccine quality failures.

Next, we analyzed the security of this vaccine information management system, including the use of BAN logic to achieve mutual authentication between communication nodes, and the EdDSA algorithm to ensure the integrity and non-repudiation of data. The EdDSA algorithm has the feature of selecting high-quality random numbers with excellent performance, which improves the efficiency of digital signatures and effectively avoids the security problems caused by low-quality random numbers that existed in some digital signature algorithms in the past. Furthermore, the proposed scheme can resist malicious attacks, such as man-in-the-middle attacks and replay attacks, to a certain extent.

In summary, this paper makes the following contributions:

1. Blockchain technology and smart contracts are used to ensure the security and traceability of vaccines from manufacturing, distribution, vaccination, and side-effect reporting. The system protects the privacy of each role while providing certain information about the vaccine based on the role's identity.
2. The entire vaccine supply management system architecture and usage scenarios are presented.
3. The use of the EdDSA algorithm for digital signatures not only guarantees the integrity of vaccine-related records but also improves the security and efficiency of digital signatures.
4. Use BAN logic to guarantee mutual authentication between unfamiliar nodes.
5. Analyzed the potential security risks of the system.

Author Contributions: Conceptualization, Y.A. and C.-L.C.; methodology, Y.A. and C.-L.C.; software, Y.A.; resources, Y.A.; validation, M.-L.C., Y.-Y.D. and Z.-Y.L.; writing–original draft preparation, Y.A.; writing–review and editing, C.-L.C., W.W., Y.-Y.D. and Z.-Y.L.; supervision, C.-L.C. and W.W.; project administration, C.-L.C.; funding acquisition, W.W. and M.-L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Natural Science Foundation of Fujian Province of China (Nos. 2021J011187 and 2021J011182) and the Ministry of Science and Technology, Taiwan, R.O.C., under contract MOST 111-2218-E-305-001–MBK and MOST 110-2410-H-324-004-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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. Nardo, D. *The Black Death*; Greenhaven Publishing LLC: New York, NY, USA, 2011; pp. 76–77.
2. Taubenberger, J.K.; Morens, D.M. 1918 Influenza: The mother of all pandemics. *Rev. Biomed.* **2006**, *17*, 69–79. [CrossRef]
3. Henderson, D.A.; Moss, B. *Smallpox and Vaccinia*; Saunders: Yorba Linda, CA, USA, 1999.
4. Roos, D. *How 5 of History's Worst Pandemics Finally Ended*; History News Network: Seattle, WA, USA, 2020.
5. World Health Organization. *Global Vaccine Safety Blueprint*; World Health Organization: Geneva, Switzerland, 2012.
6. Chen, R.T.; Shimabukuro, T.T.; Martin, D.B.; Zuber, P.L.; Weibel, D.M.; Sturkenboom, M. Enhancing vaccine safety capacity globally: A lifecycle perspective. *Vaccine* **2015**, *33*, D46–D54. [CrossRef] [PubMed]
7. Almagrabi, A.O.; Ali, R.; Alghazzawi, D.; AlBarakati, A.; Khurshaid, T. Blockchain-as-a-utility for next-generation healthcare internet of things. *CMC Comput. Mater. Contin.* **2021**, *68*, 359–376. [CrossRef]
8. Imran, M.; Zaman, U.; Imtiaz, J.; Fayaz, M.; Gwak, J. Comprehensive survey of iot, machine learning, and blockchain for health care applications: A topical assessment for pandemic preparedness, challenges, and solutions. *Electronics* **2021**, *10*, 2501. [CrossRef]
9. Gordon, W.J.; Catalini, C. Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability. *Comput. Struct. Biotechnol. J.* **2018**, *16*, 224–230. [CrossRef]
10. Chen, S.; Wu, Z.; Christofides, P.D. Cyber-security of centralized, decentralized, and distributed control-detector architectures for nonlinear processes. *Chem. Eng. Res. Des.* **2021**, *165*, 25–39. [CrossRef]
11. Mbunge, E.; Dzinamarira, T.; Fashoto, S.G.; Batani, J. Emerging technologies and COVID-19 digital vaccination certificates and passports. *Public Health Pract.* **2021**, *2*, 100136. [CrossRef]
12. Ganty, S. The veil of the COVID-19 vaccination certificates: Ignorance of poverty, injustice towards the poor. *Eur. J. Risk Regul.* **2021**, *12*, 343–354. [CrossRef]
13. Kamerow, D. Immunized? There's an app for that. *BMJ* **2021**, *372*, n85. [CrossRef]
14. Karopoulos, G.; Hernandez-Ramos, J.L.; Kouliaridis, V.; Kambourakis, G. A survey on digital certificates approaches for the covid-19 pandemic. *IEEE Access* **2021**, *9*, 138003–138025. [CrossRef]
15. Ministry of Health, Labour and Welfare. Available online: https://www.mhlw.go.jp/stf/covid-19/certificate.html (accessed on 20 December 2021).
16. Hernández-Ramos, J.L.; Karopoulos, G.; Geneiatakis, D.; Martin, T.; Kambourakis, G.; Fovino, I.N. Sharing pandemic vaccination certificates through blockchain: Case study and performance evaluation. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1530–8669. [CrossRef]
17. World Health Organization. Available online: https://www.who.int/news-room/articles-detail/world-health-organization-open-call-for-nomination-of-experts-to-contribute-to-the-smart-vaccination-certificate-technical-specifications-and-standards-application-deadline-14-december-2020 (accessed on 2 December 2020).
18. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, 21260.
19. Zhou, L.; Wang, L.; Sun, Y. MIStore: A blockchain-based medical insurance storage system. *J. Med. Syst.* **2018**, *42*, 1–17. [CrossRef]
20. Zhang, P.; White, J.; Schmidt, D.C.; Lenz, G.; Rosenbloom, S.T. FHIRChain: Applying blockchain to securely and scalably share clinical data. *Comput. Struct. Biotechnol. J.* **2018**, *16*, 267–278. [CrossRef] [PubMed]
21. Uddin, M.; Salah, K.; Jayaraman, R.; Pesic, S.; Ellahham, S. Blockchain for drug traceability: Architectures and open challenges. *Health Inform. J.* **2021**, *27*, 4571–4579. [CrossRef]
22. European Parliamentary Research Service. Available online: https://www.europarl.europa.eu/thinktank/en/document/EPRS_IDA(2020)641543 (accessed on 22 April 2020).
23. Mettler, M. Blockchain technology in healthcare: The revolution starts here. In Proceedings of the 2016 IEEE 18th International Conference on E-Health Networking, Applications and Services (Healthcom), Munich, Germany, 14–16 September 2016. [CrossRef]
24. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. Medrec: Using blockchain for medical data access and permission management. In Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016. [CrossRef]
25. Kumar, M.; Chand, S. MedHypChain: A patient-centered interoperability hyperledger-based medical healthcare system: Regulation in COVID-19 pandemic. *J. Netw. Comput. Appl.* **2021**, *179*, 102975. [CrossRef]
26. Kumar, R.; Tripathi, R. A Secure and Distributed Framework for sharing COVID-19 patient Reports using Consortium Blockchain and IPFS. In Proceedings of the 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Waknaghat, India, 6–8 November 2020. [CrossRef]
27. Sigwart, M.; Borkowski, M.; Peise, M.; Schulte, S.; Tai, S. Blockchain-based data provenance for the Internet of Things. In Proceedings of the 9th International Conference on the Internet of Things, Bilbao, Spain, 22–25 October 2019. [CrossRef]
28. Yong, B.; Shen, J.; Liu, X.; Li, F.; Chen, H.; Zhou, Q. An intelligent blockchain-based system for safe vaccine supply and supervision. *Int. J. Inf. Manag.* **2020**, *52*, 102024. [CrossRef]

29. Ricci, L.; Maesa, D.D.F.; Favenza, A.; Ferro, E. Blockchains for covid-19 contact tracing and vaccine support: A systematic review. *IEEE Access* **2021**, *9*, 37936–37950. [CrossRef]
30. Deka, S.K.; Goswami, S.; Anand, A. A blockchain-based technique for storing vaccination records. In Proceedings of the 2020 IEEE Bombay section signature conference (IBSSC), Mumbai, India, 4–6 December 2020. [CrossRef]
31. Antal, C.; Cioara, T.; Antal, M.; Anghel, I. Blockchain platform for COVID-19 vaccine supply management. *IEEE Open J. Comput. Soc.* **2021**, *2*, 164–178. [CrossRef]
32. Chauhan, H.; Gupta, D.; Gupta, S.; Singh, A.; Aljahdali, H.M.; Goyal, N.; Noya, I.D.; Kadry, S. Blockchain Enabled Transparent and Anti-Counterfeiting Supply of COVID-19 Vaccine Vials. *Vaccines* **2021**, *9*, 1239. [CrossRef]
33. Chen, J.; Chen, X.; Chen, C.L. A Traceable Blockchain-Based Vaccination Record Storage and Sharing System. *J. Healthc. Eng.* **2022**, *2022*, 2211065. [CrossRef] [PubMed]
34. Szabo, N. Smart contracts: Building blocks for digital markets. *Extropy J. Transhumanist Thought* **1996**, *18*, 28.
35. Buterin, V. A next-generation smart contract, and decentralized application platform. *White Pap.* **2014**, *3*, 2-1.
36. Bernstein, D.J.; Duif, N.; Lange, T.; Schwabe, P.; Yang, B.Y. High-speed high-security signatures. *J. Cryptogr. Eng.* **2012**, *2*, 77–89. [CrossRef]
37. Josefsson, S.; Liusvaara, I. Edwards-curve digital signature algorithm (EdDSA). *RFC 8032* **2017**. [CrossRef]
38. Wang, D.I. Secure Implementation of ECDSA Signatures in Bitcoin. *MSc Inf. Secur.* **2014**, 1–78.
39. Burrows, M.; Abadi, M.; Needham, R.M. A logic of authentication. *Proc. R. Soc. London. Math. Phys. Sci.* **1989**, *426*, 233–271. [CrossRef]