sensors

MDPI

Article

Design of Secure Protocol for Cloud-Assisted Electronic Health Record System Using Blockchain

MyeongHyun Kim [1], SungJin Yu [1,*], JoonYoung Lee [1], YoHan Park [2] and YoungHo Park [1,*]

[1] School of Electronics Engineering, Kyungpook National University, Daegu 41566, Korea; kimmyeong123@knu.ac.kr (M.K.); harry250@knu.ac.kr (J.L.)
[2] School of Computer Engineering, Keimyung University, Daegu 42601, Korea; yhpark@kmu.ac.kr
* Correspondence: darkskiln@knu.ac.kr (S.Y.); parkyh@knu.ac.kr (Y.P.); Tel.: +82-53-950-7842 (Y.P.)

Received: 15 April 2020; Accepted: 20 May 2020; Published: 21 May 2020

Abstract: In the traditional electronic health record (EHR) management system, each medical service center manages their own health records, respectively, which are difficult to share on the different medical platforms. Recently, blockchain technology is one of the popular alternatives to enable medical service centers based on different platforms to share EHRs. However, it is hard to store whole EHR data in blockchain because of the size and the price of blockchain. To resolve this problem, cloud computing is considered as a promising solution. Cloud computing offers advantageous properties such as storage availability and scalability. Unfortunately, the EHR system with cloud computing can be vulnerable to various attacks because the sensitive data is sent over a public channel. We propose the secure protocol for cloud-assisted EHR system using blockchain. In the proposed scheme, blockchain technology is used to provide data integrity and access control using log transactions and the cloud server stores and manages the patient's EHRs to provide secure storage resources. We use an elliptic curve cryptosystems (ECC) to provide secure health data sharing with cloud computing. We demonstrate that the proposed EHR system can prevent various attacks by using informal security analysis and automated validation of internet security protocols and applications (AVISPA) simulation. Furthermore, we prove that the proposed EHR system provides secure mutual authentication using BAN logic analysis. We then compare the computation overhead, communication overhead, and security properties with existing schemes. Consequently, the proposed EHR system is suitable for the practical healthcare system considering security and efficiency.

Keywords: security protocol; cloud; blockchain; electronic health record; BAN logic; AVISPA simulation

1. Introduction

As patient healthcare records have been developed from traditional paper management to electronic record management, they can be safely stored and accessed and authorized only by legitimate medical centers [1]. With the electronic health record (EHR) management system, storage availability and historical errors can be minimized, improving the availability and accuracy of healthcare records. EHR systems can help people to prevent diseases and enhance the cure rate, and ensures great convenience for medical centers and patients. However, health-related information from each healthcare system is stored in their own medical servers, respectively, in traditional EHR systems [2]. Therefore, when the patients transfer from a hospital to another one, hospitals should establish a point-to-point channel to share patients information. Furthermore, the traditional EHR system is generally established as a centralized system so that it has a single point of failure. Blockchain can serve as a helpful method to solve these problems.

In the last few years, numerous blockchain-based EHR system studies have been presented to address the problems of traditional EHR system and improve efficiency [3–5]. Blockchain is a network technology that ensures the decentralization and integrity of information by sharing records with multiple distributed nodes [6,7]. Blockchain is considered as a trusted distributed ledger that keeps transactions in a chain of chronological blocks linked through hash values. In addition, the blockchain has properties such as data anonymity, decentralization, and so on. In particular, many blockchain studies have presented various models such as ethereum and hyperledger [8]. Although both models have similar structures, hyperledger is relatively better in terms of network performance and energy efficiency [9]. Furthermore, hyperledger fabric [10] aims to solve the bottleneck problem of a cloud system and enables users to keep ownership of their own data, as well as to share data securely with feedback. However, the EHR system should consider that it is hard to store whole EHR data in blockchain because of the size and the price of blockchain [11]. Thus, if there is a sudden and unexpected demand for storage and resources, blockchain-based EHR systems should guarantee sufficient capacities.

In the last few years, many blockchain-based EHR systems have adopted cloud computing to enlarge scalability and to solve the storage problem associated with blockchain [12,13]. As an important technology to improve the development of smart medical services, cloud technology can serve as a platform for sharing information between remote hospitals and can solve the problem of remote collaboration diagnostic [14,15]. The health information can be efficiently managed on a cloud server facilitating precise and accurate diagnosis and treatment, as well as the development of various healthcare services [16]. Unfortunately, the cloud-based EHR system can be vulnerable to potential attacks because the sensitive data is sent over a public channel. To resolve these security problems, the cloud-based EHR systems require a secure and efficient protocol. Thus, we develop the security protocol using elliptic curve cryptosystems (ECC) that provides high security level, and efficient computation and communication overheads even in small storage spaces.

Recently, numerous EHR systems have been presented that combine blockchain, cloud, and authentication to solve each problem associated with cloud and blockchain [17,18]. Kaur et al. [17] presented a model architecture for EHR data using blockchain in the cloud environment to provide secure healthcare services. Furthermore, Nagasubramanian et al. [18] presented a cloud-assisted secure E-health record system using blockchain to provide integrity and decentralization for the EHR sharing and health diagnosis. However, these cloud-assisted EHR systems using blockchain [17,18] do not specifically address a secure protocol for registration, authentication, transaction uploading, and so on. Therefore, we propose the secure protocol for cloud-assisted EHR system using blockchain to guarantee security, integrity, and decentralization for EHR sharing and health diagnosis. The proposed EHR system utilizes the cloud technology to achieve storage efficiency, and the data in each block only stores metadata to increase block construction efficiency and minimize distributed storage waste. Furthermore, in the proposed EHR system, blockchain technology is used to efficiently provide data integrity and access control using log transactions. Moreover, the proposed EHR system provides secure health data sharing in a public channel using ECC.

1.1. Research Contributions

The detailed contributions in this paper are summarized as below.

- We propose the secure protocol for cloud-assisted EHR system using blockchain. The proposed scheme combines cloud computing, blockchain, and authentication to provide a secure and effective medical diagnosis for legitimate patients.
- The proposed scheme withstands various attacks, including impersonation, session key disclosure, and replay attacks, and also provides secure mutual authentication and anonymity.
- We present the Burrows–Abadi–Needham (BAN) logic analysis [19,20] to analyze that the proposed scheme provides secure mutual authentication.

- We perform the automated validation of internet security protocols and applications (AVISPA) [21,22] to analyze against man-in-the-middle (MITM) and replay attacks. Furthermore, we show the performance analysis of the proposed scheme with existing schemes.

1.2. Organization

The remainder of this paper is organized as follows. Section 2 presents the related works, and Section 3 shows the preliminaries for help explanation of this paper. In Sections 4 and 5, we introduce the system model and also propose a secure protocol for cloud-assisted EHR system using blockchain. Section 6 performs the security analysis of the proposed scheme using informal and formal security analysis. In Section 7, we compare the performance analysis of the proposed scheme with related schemes. Finally, we summarize the paper in Section 8.

2. Related Works

In the past decades, many authentication schemes in the healthcare system have been presented to ensure secure healthcare service and EHR sharing [23–26]. Kumar et al. [23] presented an efficient authentication scheme for healthcare applications in wireless medical sensor networks to provide secure healthcare services. Wu et al. [24] presented a reliable RFID-based authentication scheme in healthcare environments. Their scheme [24] does not reveal any private data, including the identity number and the health data of the legitimate patient. Liu et al. [25] presented a remote authentication protocol for wireless body area networks. Their scheme [25] is not suitable for limited-resource wearable sensor devices because it utilizes bilinear pairing cryptography with high computation and communication overheads. Renuka et al. [26] presented a three-factor authentication protocol for smart healthcare using ECC. Renuka et al. [26] demonstrated that their scheme can prevent against various attacks. However, their schemes for the healthcare system [23–26] are essentially a centralized system so that these schemes do not solve problems such as the single point of failure. Therefore, a blockchain mechanism with decentralized properties is essential for solving the problems of centralized systems.

In the last few years, many EHR system studies have been presented using blockchain to ensure data integrity along with decentralized properties [27–29]. Pandey and Litoriya [27] presented secure e-health networks from counterfeit medicine penetration using blockchain. Their scheme [27] ensures data integrity and security capability properties against drug data to provide secure healthcare services. Agbo and Mahmoud [28] presented a comparison of blockchain frameworks for healthcare applications. Tanwar et al. [29] presented a blockchain-based EHR system for secure medical data sharing. Their scheme [29] can avoid the reliability problem of the trusted third parties, and also can provide secure medical services between each entity. However, these schemes for the EHR systems using blockchain [27–29] should consider that it is hard to store whole EHR data in blockchain because of the size and price of blockchain [11]. Therefore, if there is a sudden and unexpected demand for storage and resources, the EHR systems using blockchain have to guarantee sufficient capacities. Therefore, these schemes require a cloud-based mechanism in the EHR system to provide cloud storage technology and decentralized properties using blockchain.

Recently, numerous cloud-based EHR system studies using the blockchain have been presented to solve the storage overload problem associated with blockchain [30–32]. Wang et al. [30] presented a cloud-assisted EHR sharing to ensure security and privacy using blockchain. Their scheme [30] uses searchable encryption and proxy re-encryption to realize data security and access control. Chen et al. [31] designed a secure storage scheme based on blockchain and cloud storage to manage personal health data. Cheng et al. [32] presented a secure medical data sharing scheme based on blockchain utilizing cloud techniques. Their scheme [32] uses bilinear mapping to provide secure medical data sharing and low storage and computing overhead. However, these cloud-based EHR systems using blockchain [30–32] have been studied so far, but a secure authentication scheme for EHR sharing has not been specifically considered. Therefore, we present a secure cloud-assisted EHR system using blockchain to ensure secure EHR sharing.

3. Preliminaries

In this section, we introduce the preliminaries for help explanation of this paper.

3.1. Adversary Model

We present the widely used Dolev–Yao (DY) model [33] to analyze the security of the proposed protocol. The detailed assumptions of the DY model are as follows.

- An attacker can delete, inject, eavesdrop, and intercept the messages transmitted over a public channel.
- An attacker can steal the smartcard of legitimate patients and can extract secret values stored in a smartcard using power-analysis [34,35].
- An attacker may attempt various attacks such as impersonation, MITM, replay, session key disclosure attacks, and so on [36,37].

3.2. Hyperledger Fabric

In 2015, hyperledger fabric [10] was presented as an open source blockchain proposed by the Linux Foundation. The goal of this technology is to promote cross-industry cooperation using blockchain. Hyperledger fabric does not require digital currency and provides various advantages such as blockchain performance and reliability. Hyperledger fabric uses practical byzantine fault-tolerant (PBFT) consensus algorithm [38,39]. Therefore, we apply the PBFT algorithm to the proposed system to provide an effective consensus ability. The hyperledger architecture consists of six blockchain components:

1. Membership Service Provider (MSP): MSP is a component that validates and authenticates credentials and defines the rules for accessing a network. The MSP manages user identities and authenticates all participants in the network, making hyperledger fabric available as both private and permissioned networks. This includes providing credentials for the clients to propose transactions. As a result, a single hyperledger fabric network can be controlled by multiple MSPs.
2. Smart Contract: The smart contract of hyperledger fabric is called chaincode. Chaincode is software that defines assets and related transactions. The chaincode is called when the application needs to interact with the ledger. Every chaincode has an attached endorsement policy, which applies to all smart contracts defined in it. This identifies the organizations that need to sign transactions generated by smart contracts. In addition, smart contracts have the advantage of being able to make different smart contracts within the channel or across different channels.
3. Ordering Service: The ordering service packages a transaction in blocks and delivers it to the channel's peers. It ensures the transaction delivery via the network. It communicates with peers and endorsing peers.
4. Identity: Each node in the network peer, client, ordered, and the manager has a digital identity with the format of certificate X.509. This identity is used to verify at every stage of the transaction to ensure if the source of the transaction is a valid source. In addition to multiple assurances, validation, and version control checks that occur, there are ongoing identity verifications happening during each stage of the transaction flow.
5. Channels: Hyperledger fabric networks can have multiple channels. Channels allow organizations to use the same network while maintaining separation between multiple blockchain. Only the peer of the channel can provide to see transactions made by all members of the channel.
6. Peer Nodes: Peer nodes constitute a fundamental element of the network as they host smart contracts within the ledger. Peer nodes execute chaincode, access ledger data, approve transactions, and interface with applications.

4. Cloud-Assisted EHR System Model Using Blockchain

We introduce a cloud-assisted EHR system based on a hyperledger fabric in Figure 1. To improve the security and efficiency of medical data, this system is built on medical centers that share EHR in specific regions. The system model for the EHR comprises the four entities: the patient, the medical center, the cloud server, and the network administrator. The detailed descriptions of each entity are described as follows.

1. Patient: A patient transmits the health data to the medical center in order to receive healthcare services through healthcare devices and wearable sensors. Health data of the patient are recorded in EHR with healthcare services provided by the medical center.
2. Medical Centers: The medical centers are registered by the network administrator and participate in the private blockchain. The medical centers generate EHR and store it to the cloud server for sharing with other medical centers. When the medical centers view the EHRs of other medical center's the patient, they upload a log of EHR data to the blockchain as a transaction form.
3. Network Administrator: A network administrator is a trusted entity, responsible for the registration of participants, that manages the private blockchain.
4. Cloud Server: A cloud server is a trusted entity that has sufficient computing power and capacity. The cloud server stores and manages the patient's EHRs to provide secure data sharing and storage resources. A cloud server receives the EHR data from the medical center and sends the EHR to other medical centers requesting the EHR using a pre-shared secret key.

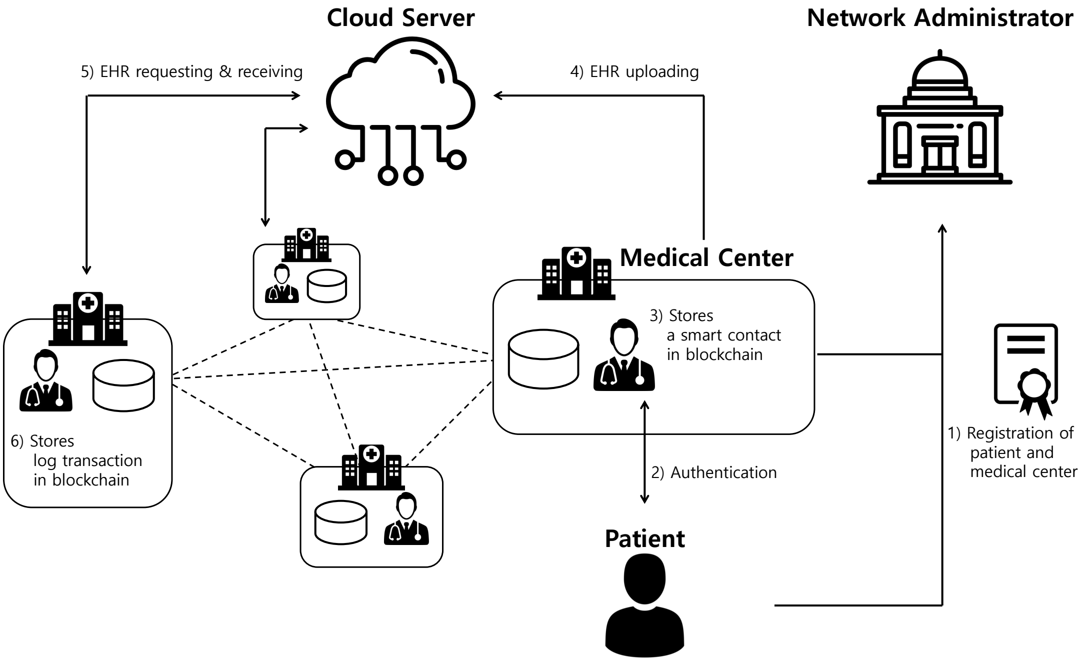


Figure 1. Proposed cloud-assisted electronic health record (EHR) system model using blockchain.

The communication flows of the proposed EHR system are described as follows.

1. Patient and doctor register their identities with the help of a network administrator to access EHR services.
2. Patient and doctor authenticate each other and establish a session key for future secure communication.
3. The medical center receives the information for a smart contract from the patient using a session key. Then, the medical center generates a patient's smart contract and EHR. After that, the medical center uploads a smart contract at the blockchain.

4. The medical center encrypts EHRs of the legitimate patient using a pre-shared secret key and sends it to the cloud server. Then, the cloud server decrypts the encrypted EHR data and stores EHR data in the database.
5. The other medical center requests the EHR data of the medical center to the cloud server. Next, the cloud server encrypts EHR data of the medical center using a pre-shared secret key and sends it to the other medical center.
6. Finally, the medical center decrypts the encrypted EHR data and then uploads the log transaction, including the patient and medical center masked identities, signatures, and timestamps at the blockchain.

5. Proposed Protocol for Cloud-Assisted EHR System Using Blockchain

We present a secure protocol for cloud-assisted EHR system using hyperledger fabric. The proposed EHR system is that only the EHRs can be outsourced by authenticated participants and each operation on outsourcing EHRs is integrated into the blockchain as a transaction. The proposed scheme consists of six phases: the registration, authentication, smart contract uploading, EHR storing, EHR requesting, and log transaction uploading. Before the registration phase, a network administrator (NA) sets up the networks. The NA selects a base point G over an elliptic curve E_p with order p that is a large prime number. P of order q is one of G's generators, in which q is a large prime number. Then, the NA selects a secret key s_{NA} and generates a public key $PK_{NA} = s_{NA} \cdot G$. Finally, NA shares the network configuration and policies with all system participants. Furthermore, the NA publishes, $\{p, q, G, P, PK_{NA}\}$ as system parameters, and a cloud server (CS) establishes a secure pre-shared key with medical centers. Table 1 illustrates the notations used in the proposed scheme.

Table 1. Notations.

Notations	Meanings
P_i	i-th patient
MC_j	j-th medical center
NA	Network Administrator
ID_i, ID_j	Identity of P_i and MC_j
PW_i	Password of P_i
r_i, r_j	Secret keys of P_i and MC_j
s_{NA}	Secret key of NA
T_1, T_2	Timestamps
T_{up}, T_{access}	Uploading/accessing time of EHR
K_{NA}, r_{NA}	Random numbers generated by NA
PK_i, PK_j	Public keys of P_i and MC_j
$Cert_i, Cert_j$	Certificates of P_i and MC_j
$E_p(a, b)$	A nonsingular elliptic curve $y^2 = x^3 + ax + b \pmod{p}$
G	A base point for elliptic curve
HID_i, PID_j	Pseudo-identities of P_i and MC_j
T_x	Log transaction
KMS_j	Secure pre-shared key among MC_j and CS
EHR	Electronic health record
RI	Information of health record
RE	Request message of EHR
SK	Common session key shared among P_i and MC_j
$h(*)$	Collision resistant one-way hash function
$\oplus$	XOR operation
$\vert\vert$	Concatenation operation

5.1. Registration Phase

In the proposed scheme, the registration phase consists of the patient registration and the medical center registration.

5.1.1. Patient Registration Phase

If a patient (P_i) wants to receive a medical diagnosis, the P_i must first register his/her information with the NA and generate a private key and a public key. The patient registration phase is executed over a secure channel. Figure 2 shows the patient registration phase and detailed steps are as follows.

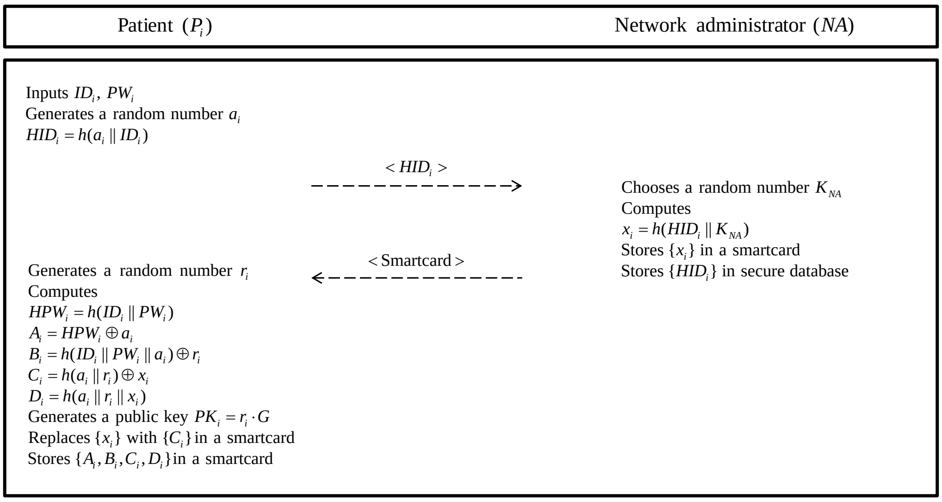


Figure 2. Patient registration phase of the proposed protocol.

Step 1: The P_i requests registration to the network administrator NA. First, P_i inputs identity ID_i and password PW_i. Then, the P_i generates a random number a_i and computes $HID_i = h(a_i||ID_i)$ and sends HID_i to the NA.

Step 2: The NA chooses a random number K_{NA} and computes $x_i = h(HID_i||K_{NA})$ using the HID_i received from the P_i. Then, the NA stores $\{x_i\}$ into the smartcard and issues it to the P_i in the blockchain. Finally, the NA stores $\{HID_i\}$ in secure database.

Step 3: After the P_i receives smartcard from the NA, the P_i generates a random number r_i as a secret key. P_i computes $HPW_i = h(ID_i||PW_i)$, $A_i = HPW_i \oplus a_i$, $B_i = h(ID_i||PW_i||a_i) \oplus r_i$, $C_i = h(a_i||r_i) \oplus x_i$ and $D_i = h(a_i||r_i||x_i)$. And then, the P_i generates a public key $PK_i = r_i \cdot G$ and replaces $\{x_i\}$ with $\{C_i\}$ in a smartcard. Finally, P_i stores $\{A_i, B_i, C_i, D_i\}$ in the smartcard.

5.1.2. Medical Center Registration Phase

A medical center (MC_j) must register with the NA to have a key agreement with patients and exchange information with other related medical centers. The masked identity of the MC_j is shared with other entities. This registration phase is also executed over a secure channel. The detailed steps are described as follows and are illustrated in Figure 3.

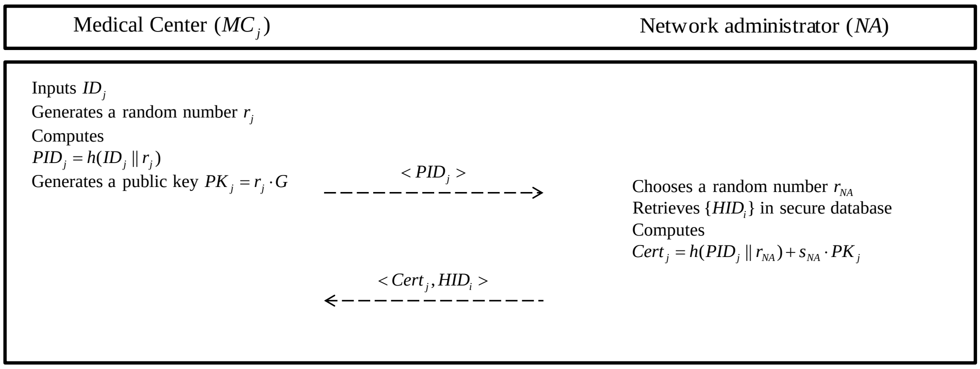


Figure 3. Medical center registration phase of the proposed protocol.

Step 1: A medical center MC_j chooses a unique identity ID_j and generates a random number r_j as a its secret key. Then, the MC_j computes a masked identity $PID_j = h(ID_j||r_j)$ and generates a public key $PK_j = r_j \cdot G$. MC_j sends PID_j to the NA.

Step 2: After receiving registration request message, the NA chooses a random number r_{NA} and retrieves $\{HID_i\}$ in secure database. Then, the NA computes $Cert_j = h(PID_j||r_{NA}) + s_{NA} \cdot PK_j$. The NA stores $Cert_j$ with PID_j and sends $\{Cert_i, HID_i\}$ to MC_j.

Step 3: After the MC_j receives the messages, the MC_j stores $\{Cert_j, HID_i\}$ in secure database.

5.2. Authentication Phase

If the P_i wants a secure health diagnosis, the patient and medical center must establish a session key. The detailed steps are as following in Figure 4.

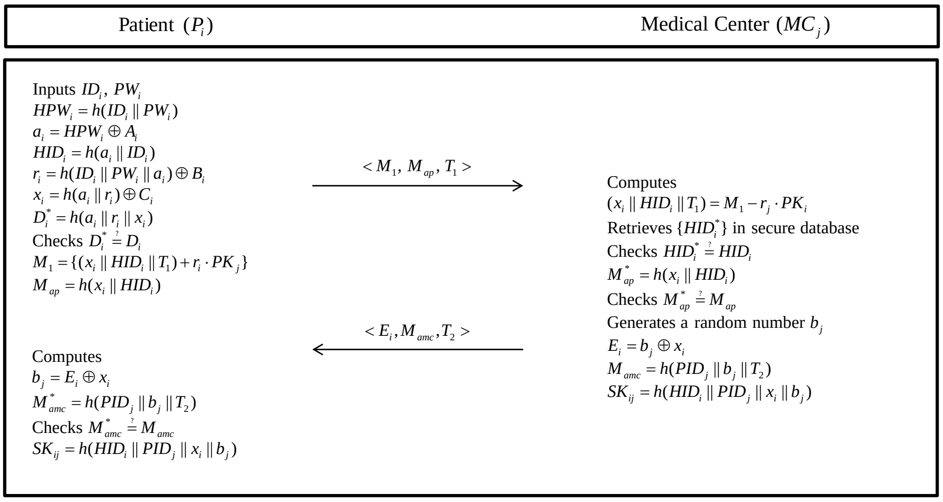


Figure 4. Authentication phase of the proposed protocol.

Step 1: The P_i inputs his/her ID_i, PW_i, and smartcard. Then, the smartcard computes $HPW_i = h(ID_i||PW_i)$, $a_i = HPW_i \oplus A_i$, $HID_i = h(a_i||ID_i)$, $r_i = h(ID_i||PW_i||a_i) \oplus B_i$, $x_i = h(a_i||r_i) \oplus C_i$, and $D_i^* = h(a_i||r_i||x_i)$. Then, the smartcard checks whether $D_i^* \overset{?}{=} D_i$. If it is correct, the P_i generates a timestamp T_1 and encrypts messages $M_1 = \{(x_i||HID_i||T_1) + r_i \cdot PK_j\}$ and computes $M_{ap} = h(x_i||HID_i)$. Next, the P_i sends a message $< M_1, M_{ap}, T_1 >$ to MC_j via a public channel.

Step 2: After receiving the message $< M_1, M_{ap}, T_1 >$, the MC_j decrypts $(x_i||HID_i||T_1) = M_1 - r_j \cdot PK_i$. After that, the MC_j retrieves HID_i^* in secure database and checks whether $HID_i^* \overset{?}{=} HID_i$. If it is correct, the MC_j computes $M_{ap}^* = h(x_i||HID_i)$ and checks whether $M_{ap}^* \overset{?}{=} M_{ap}$. If it is valid, the MC_j generates a random number b_j and timestamp T_2 and calculates $E_i = b_j \oplus x_i$, $M_{amc} = h(PID_j||b_j||T_2)$. HID_i updates at the proper period. After that, the MC_j generates a session key $SK_{ij} = h(HID_i||PID_j||x_i||b_j)$. Finally, MC_j sends message $< E_i, M_{amc}, T_2 >$ to P_i over an open channel.

Step 3: When the P_i receives the message from the MC_j, the P_i computes $b_j = E_i \oplus x_i$, and $M_{amc}^* = h(PID_j||b_j||T_2)$. Then, the P_i checks whether $M_{amc}^* \overset{?}{=} M_{amc}$. If it is valid, the P_i computes a session key $SK_{ij} = h(HID_i||PID_j||x_i||b_j)$.

5.3. Smart Contract Uploading Phase

After receiving information for the smart contract from the P_i, the MC_j generates a smart contract and then uploads the smart contract in the blockchain. The detailed steps are as following in Figure 5.

Step 1: The P_i generates message $M_{sc} = h(HID_i||PID_j||SK_{ij})$ and encrypts his/her information with SK_{ij}; $M_{inf} = (HID_i||PID_j)_{SK_{ij}}$. Then, the P_i sends $< M_{sc}, M_{inf} >$ to the MC_j.

Step 2: The MC_j computes $M^*_{sc} = h(HID_i||PID_j||SK_{ij})$ and checks $M^*_{SC} \overset{?}{=} M_{SC}$. If it is valid, MC_j decrypts M_{inf} and generates a smart contract Sc using $(HID_i, PID_j, Cert_j)$. Finally, the MC_j uploads Sc in the blockchain.

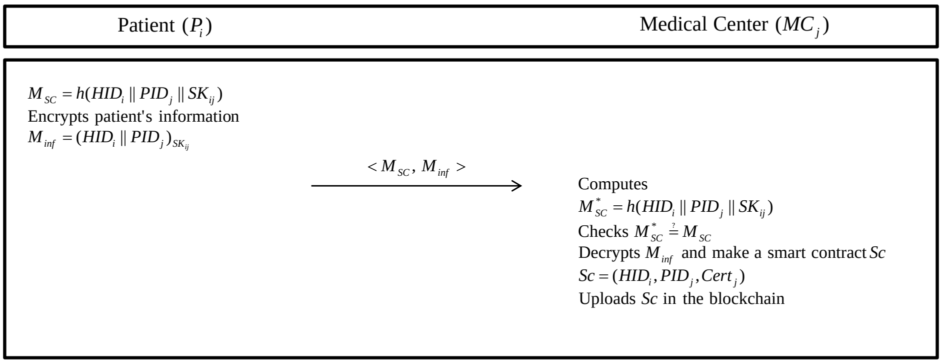


Figure 5. Smart contract uploading phase of the proposed protocol.

5.4. EHR Storing Phase

After uploading smart contract, the MC_j generates EHR_i and stores EHR_i in CS. Detailed steps are as follows in Figure 6.

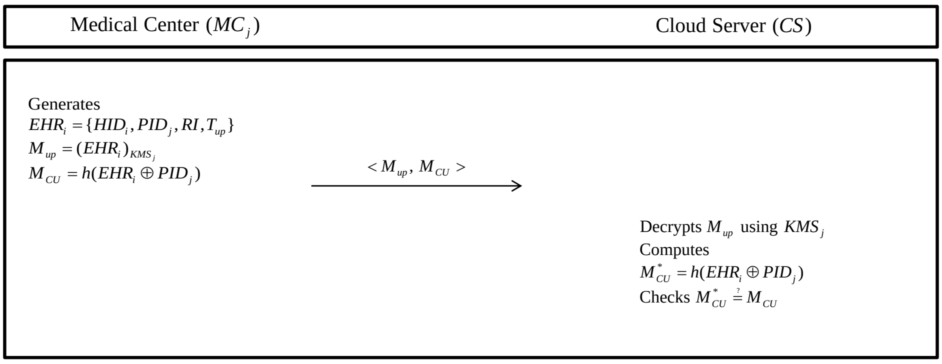


Figure 6. EHR storing phase of the proposed protocol.

Step 1: The MC_j generates EHR_i including HID_i, PID_j, an information of health record RI, and EHR's uploading time T_{up}. Then, the MC_j encrypts EHR_i using a secure pre-shared key $M_{up} = (EHR_i)_{KMS_j}$ and computes $M_{CU} = h(EHR_i \oplus PID_j)$. Finally, the MC_j sends $< M_{up}, M_{CU} >$ to the CS.

Step 2: The CS decrypts M_{up} with KMS_j, computes $M^*_{CU} = h(EHR_i \oplus PID_j)$ and checks $M^*_{CU} \overset{?}{=} M_{CU}$. If it is correct, the CS stores EHR_i in the server database.

5.5. EHR Requesting Phase

If the MC_j wants to confirm EHR_i, MC_j sends request messages to the CS. Then, the CS sends EHR_i to MC_j. Detailed steps are as follows in Figure 7.

Step 1: The MC_j generates request messages RE and encrypts $M_{req} = (RE||PID_j)_{KMS_j}$ using KMS_j and computes $M_{CR} = h(RE \oplus PID_j)$. Then, the MC_j sends $< M_{req}, M_{CR} >$ to the CS.

Step 2: After receiving the messages $< M_{req}, M_{CR} >$, the CS decrypts M_{req} with KMS_j. After that, the CS computes $M^*_{CR} = h(RE \oplus PID_j)$ and checks $M^*_{CR} \overset{?}{=} M_{CR}$. If it is correct, the CS

retrieves EHR_i corresponding request. The CS encrypts EHR_i with KMS_j and calculates $M_{CE} = h(RE||EHR_i||PID_j)$. After then, the CS sends $< M_E, M_{CE} >$ to the MC_j.

Step 3: MC_j decrypts the received M_E with KMS_j and computes $M^*_{CE} = h(RE||EHR_i||PID_j)$. Then, the MC_j checks $M^*_{CE} \overset{?}{=} M_{CE}$. If it is not valid, the MC_j eliminates communication and received data.

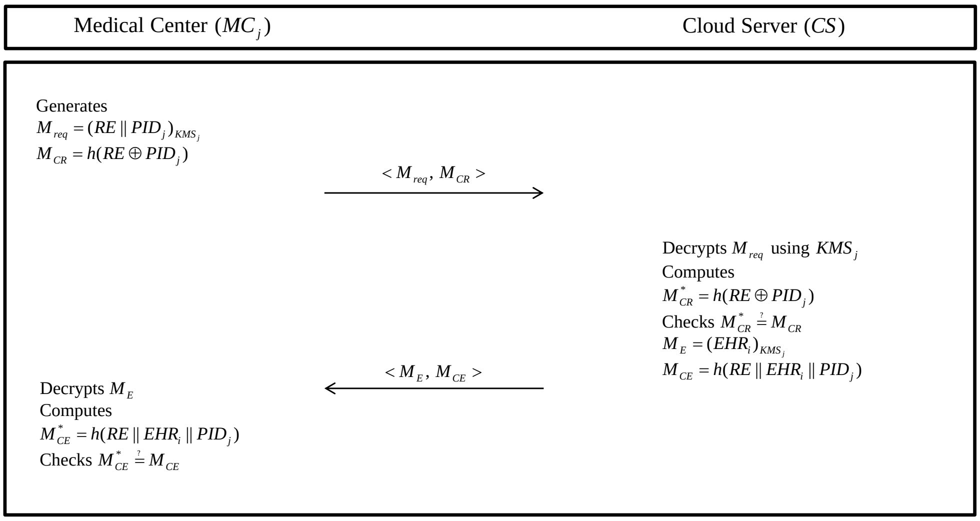


Figure 7. EHR requesting phase of the proposed protocol.

5.6. Log Transaction Uploading Phase

After MC_j receives EHR_i from CS, MC_j generates a log transaction and uploads the log transaction in the blockchain. The MC_j generates a log transaction $Tx = \{HID_i, PID_j, T_{access}, Sig_j\}$, where T_{access} is accessing time of EHR_i and Sig_j is a signature of the MC_j. Finally, the MC_j uploads Tx in the blockchain. The detailed step is as following in Figure 8.

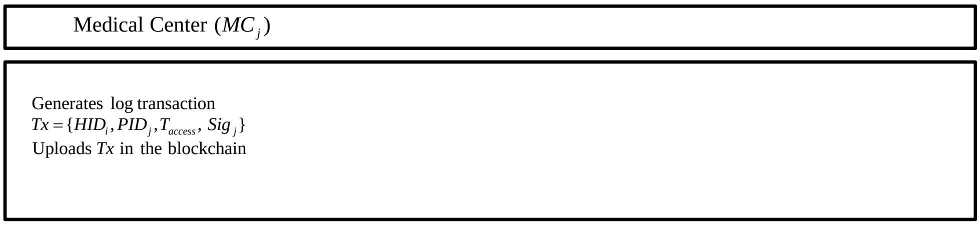


Figure 8. Log transaction uploading phase of the proposed protocol.

6. Security Analysis

In this section, we analyze the proposed protocol as a security aspect. We show that the proposed protocol is secure against malicious attacks using informal analysis. We also prove that the proposed protocol can provide secure mutual authentication using a widely adopted BAN logic. In addition, we simulate Automated Validation of Internet Security Protocols and Applications (AVISPA) to prove that the proposed protocol is secure against MITM and replay attacks.

6.1. Informal Security Analysis

We analyze the proposed protocol to perform informal security analysis and show the protocol can resist various attacks. Moreover, we show that our protocol can provide secure mutual authentication and patient's anonymity.

6.1.1. Impersonation Attack

A malicious adversary M_A tries to impersonate a legitimate patient P_i to obtain sensitive information. To impersonate P_i, the M_A has to successfully compute a message $< M_1, M_{ap}, T_1 >$. However, the M_{ap} is masked with a secret value x_i and the adversary cannot compute x_i because he/she does not know a random number K_{NA}. Moreover, the M_1 is encrypted by the P_i's secret key. Therefore, the proposed protocol is secure against impersonation attacks.

6.1.2. Session Key Disclosure Attack

If the M_A wants to generate a legitimate session key $SK_{ij} = h(HID_i||PID_j||x_i||b_j)$, the M_A must know random number b_j. However, the M_A cannot obtain b_j. Moreover, the M_A cannot reveal real the identities of P_i and MC_j because they are masked with random numbers a_i and r_j. Therefore, the proposed protocol can prevent session key disclosure attacks.

6.1.3. Perfect Forward Secrecy

Even if a M_A knows a long-term private secret key s_{NA}, the M_A cannot obtain the previous session key, because a session key $SK_{ij} = h(HID_i||PID_j||x_i||b_j)$ does not include s_{NA}. Further, if the long-term private parameter K_{NA} is compromised, the M_A cannot obtain x_i. Because x_i is masked with HID_i and HID_i is masked with a random number a_i. Therefore, the proposed protocol guarantees perfect forward secrecy.

6.1.4. Replay Attack

Suppose a M_A learns transmitted messages performing a replay attack. However, the M_A cannot use previous messages, because transmitted messages include timestamps, and P_i and MC_j check the timestamps are correct. Then, they check that $M^*_{ap} \stackrel{?}{=} M_{ap}$ and $M^*_{amc} \stackrel{?}{=} M_{amc}$ are correct. Thus, the proposed protocol can resist replay attacks.

6.1.5. Privileged Insider Attack

Suppose a privileged insider user of the system, the user is an insider adversary. The insider adversary knows the registration information $< HID_i >$ of a legitimate user. Moreover, the adversary also can know stored values $\{A_i, B_i, C_i, D_i\}$ in the smartcard to perform power analysis attacks. However, stored values in the smartcard are masked with HPW_i. Therefore, the adversary cannot know HPW_i that cannot guess a valid password. Therefore, the proposed protocol prevents privileged insider attack.

6.1.6. Anonymity

A M_A cannot reveal a legitimate patient's real identity ID_i, because ID_i is masked by hash function or encryption with random numbers or secret key. Therefore, our protocol provides the patient's anonymity.

6.1.7. Mutual Authentication

According to Section 6.1.1, the M_A cannot compute a valid session key and cannot impersonate a legitimate patient. Moreover, P_i and MC_j check a legitimate entity to verify whether $M^*_{ap} \stackrel{?}{=} M_{ap}$ and $M^*_{amc} \stackrel{?}{=} M_{amc}$ are correct. If the conditions are correct, the P_i and MC_j authenticate each other. Therefore, our protocol can provide secure mutual authentication.

6.2. BAN Logic Analysis

We demonstrate that the proposed protocol provides secure mutual authentication between P and MC using BAN logic [19,20]. Table 2 presents BAN logic notations. In addition, we define the rules, goals, idealized forms, and assumptions for performing BAN logic analysis.

Table 2. Notations of Burrows–Abadi–Needham (BAN) logic.

Notation	Description
$X\mid\equiv Q$	X **believes** statement Q
$X\mid\sim Q$	X once **said** Q
$X\Rightarrow Q$	X **controls** statement Q
#Q	Statement Q is **fresh**
$X \lhd Q$	X **sees** statement Q
$< Q >_Z$	Formula Q is **combined** with formula Z
$\{Q\}_K$	Q is **encrypted** under key K
$\overset{K}{\longrightarrow} Y$	Y has K as a **public key**
$X \overset{K}{\leftrightarrow} Y$	X and Y may use **shared key** K to communicate
SK	Session key used in the current session

6.2.1. BAN Logic Rules

The BAN logic rules are defined as follows.

1. **1.** Message meaning rule:

$$\frac{X \mid\equiv X \overset{K}{\leftrightarrow} Y, \quad X \lhd \{Q\}_K}{X \mid\equiv Y \mid\sim Q}$$

2. **2.** Nonce verification rule:

$$\frac{X \mid\equiv \#(Q), \quad X \mid\equiv Y \mid\sim Q}{X \mid\equiv Y \mid\equiv Q}$$

3. **3.** Jurisdiction rule:

$$\frac{X \mid\equiv Y \mid\Longrightarrow Q, \quad X \mid\equiv Y \mid\equiv Q}{X \mid\equiv Q}$$

4. **4.** Freshness rule:

$$\frac{X \mid\equiv \#(Q)}{X \mid\equiv \#(Q,Z)}$$

5. **5.** Belief rule:

$$\frac{X \mid\equiv (Q,Z)}{X \mid\equiv Q}$$

6.2.2. Goals

We define the security goals to prove that the proposed system is capable of performing secure mutual authentication.

Goal 1: $P \mid\equiv (P \xleftrightarrow{SK} MC)$

Goal 2: $P \mid\equiv MC \mid\equiv (P \xleftrightarrow{SK} MC)$

Goal 3: $MC \mid\equiv (P \xleftrightarrow{SK} MC)$

Goal 4: $MC \mid\equiv P \mid\equiv (P \overset{SK}{\longleftrightarrow} MC)$

6.2.3. Idealized Forms

We define the idealized forms as below.

Msg_1**:** $P \rightarrow MC$: $(x_i, HID_i, T_1)\overset{PK_j}{\longrightarrow}_{MC}$

Msg_2**:** $MC \rightarrow P$: $(PID_j, b_j, T_2)_{x_i}$

6.2.4. Assumptions

The initial assumptions are given below.

A_1**:** $P \mid\equiv (P \overset{x_i}{\longleftrightarrow} MC)$

A_2**:** $MC \mid\equiv \#(PK_j)$

A_3**:** $P \mid\equiv \#(b_1)$

A_4**:** $P \mid\equiv MC \Rightarrow (P \overset{SK}{\longleftrightarrow} MC)$

A_5**:** $MC \mid\equiv P \Rightarrow (P \overset{SK}{\longleftrightarrow} MC)$

A_6**:** $MC \mid\equiv \#(x_i)$

A_7**:** $MC \mid\equiv \#(T_1)$

A_8**:** $P \mid\equiv \#(T_2)$

6.2.5. Proof Using BAN Logic

We perform the BAN logic analysis. The detailed steps are as follows.

Step 1: From Msg_1 we can get,

$$S_1 : MC \lhd (x_i, HID_i, T_1)\overset{PK_j}{\longrightarrow}_{MC}$$

Step 2: From the message meaning rule with S_1 and A_2,

$$S_2 : MC \mid\equiv P\mid \sim (x_i, HID_i, T_1)$$

Step 3: We use the freshness rule with S_2 and A_6,

$$S_3 : MC \mid\equiv \#(x_i, HID_i, T_1)$$

Step 4: Using the nonce verification rule with S_2 and S_3,

$$S_4 : MC \mid\equiv P \mid\equiv (x_i, HID_i, T_1)$$

Step 5: By the Belief rule with S_4 and A_7,

$$S_5 : MC \mid\equiv P \mid\equiv (x_i, HID_i)$$

Step 6: Because of the session key $SK = h(HID_i||PID_j||x_i||b_j)$, from S_5 and A_3,

$$S_6 : MC \mid\equiv P \mid\equiv (P \overset{SK}{\longleftrightarrow} MC) \qquad \textbf{(Goal 4)}$$

Step 7: Using the jurisdiction rule with S_6 and A_5,

$$S_7 : MC \mid\equiv (P \overset{SK}{\longleftrightarrow} MC) \qquad \textbf{(Goal 3)}$$

Step 8: From Msg_2 we can get,

$$S_8 : P \lhd (PID_j, b_j, T_2)_{x_i}$$

Step 9: From the message meaning rule with S_8 and A_1,

$$S_9 : P \mid\equiv MC\mid \sim (PID_j, b_j, T_2)_{x_i}$$

Step 10: We use the freshness rule with S_9 and A_3,

$$S_{10} : P \mid\equiv \#(PID_j, b_j, T_2)_{x_i}$$

Step 11: Using the nonce verification rule with S_8 and S_9,

$$S_{11} : P \mid\equiv MC \mid\equiv (PID_j, b_j, T_2)_{x_i}$$

Step 12: By the belief rule with S_{11} and A_8,

$$S_{12} : P \mid\equiv MC \mid\equiv (PID_j, b_j)_{x_i}$$

Step 13: Because of the session key $SK = h(HID_i||PID_j||x_i||b_j)$, from S_{12} and A_6,

$$S_{13} : P \mid\equiv MC \mid\equiv (P \overset{SK}{\longleftrightarrow} MC) \qquad \textbf{(Goal 2)}$$

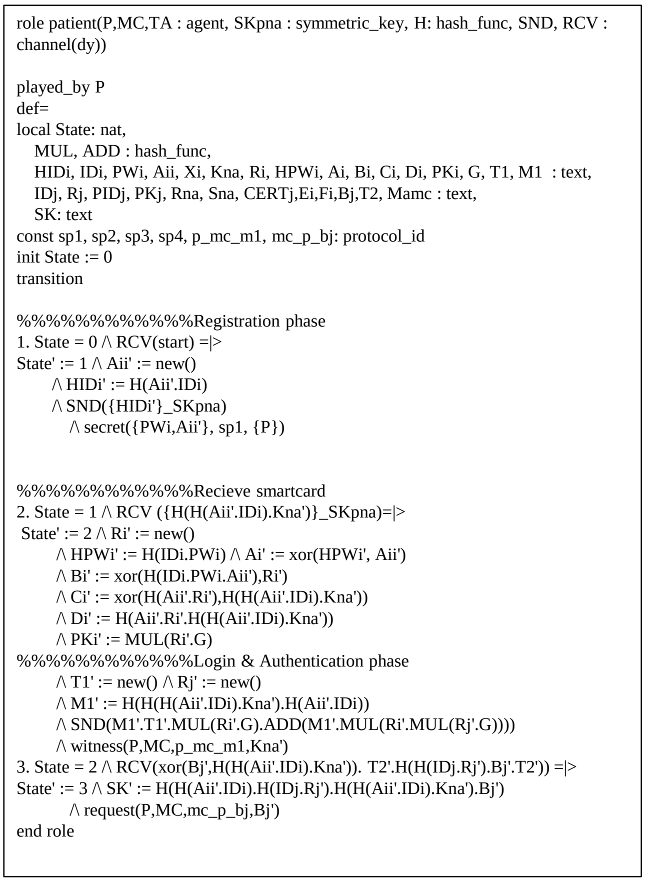
Step 14: Using the jurisdiction rule with S_{13} and A_4,

$$S_{14} : P \mid\equiv (P \overset{SK}{\longleftrightarrow} MC) \qquad \textbf{(Goal 1)}$$

Therefore, the goals 1–4 clearly show that the proposed protocol provides secure mutual authentication between P_i and MC_j.

6.3. AVISPA Analysis

This section shows the proposed protocol can resist against adversary's replay and MITM attacks to perform AVISPA simulation [21,22]. The AVISPA tool consists of High-Level Protocol Specification Language (HLPSL) [40] to generate input format (IF) of four back-ends, i.e., "On-the-Fly Model Checker (OFMC)", "Constraint Logic-based Attack Searcher (CL-AtSe)", "Tree automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP)", and "SAT-based Model Checker (SATMC)". Then, the output format (OF) is created and the safety of the protocol is verified using OF. Generally, verification is performed with OFMC and CL-AtSe. The HLPSL syntax of each entity is shown in Figures 9–11. Furthermore, the goal and environment of the protocol are shown in Figure 12. Goal and environment describe participants, security goals, and environment conditions. As a Figure 13, the results of AVISPA simulation under OFMC and CL-AtSe is safe. The results show that OFMC has 5.88 search time and visits 1040 nodes with 9 piles depths. Furthermore, the CL-AtSe analyzed in 0.07 seconds. Therefore, our proposed protocol provides security against MITM and replay attacks.



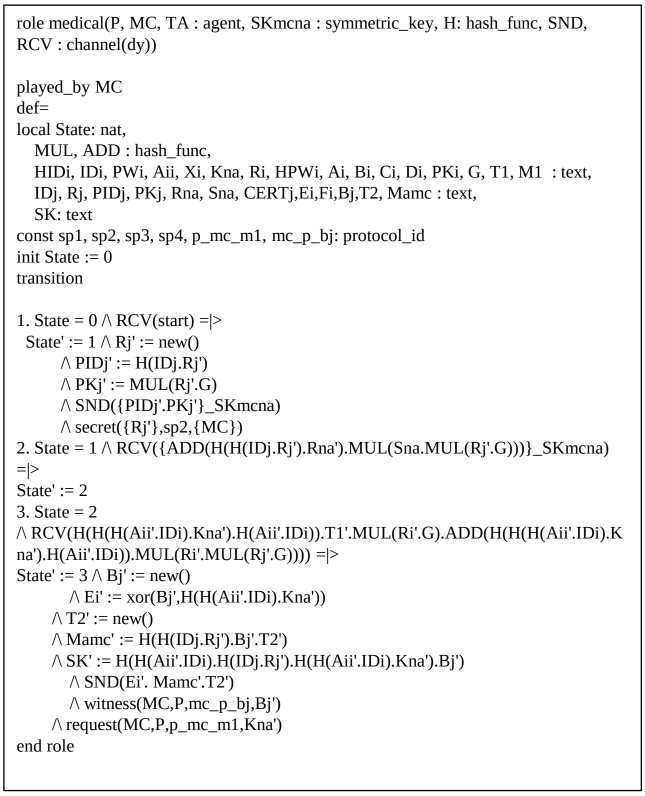
```
role patient(P,MC,TA : agent, SKpna : symmetric_key, H: hash_func, SND, RCV :
channel(dy))

played_by P
def=
local State: nat,
   MUL, ADD : hash_func,
   HIDi, IDi, PWi, Aii, Xi, Kna, Ri, HPWi, Ai, Bi, Ci, Di, PKi, G, T1, M1  : text,
   IDj, Rj, PIDj, PKj, Rna, Sna, CERTj,Ei,Fi,Bj,T2, Mamc : text,
   SK: text
const sp1, sp2, sp3, sp4, p_mc_m1, mc_p_bj: protocol_id
init State := 0
transition

%%%%%%%%%%Registration phase
1. State = 0 /\ RCV(start) =|>
State' := 1 /\ Aii' := new()
      /\ HIDi' := H(Aii'.IDi)
      /\ SND({HIDi'}_SKpna)
         /\ secret({PWi,Aii'}, sp1, {P})


%%%%%%%%%%Recieve smartcard
2. State = 1 /\ RCV ({H(H(Aii'.IDi).Kna')}_SKpna)=|>
 State' := 2 /\ Ri' := new()
       /\ HPWi' := H(IDi.PWi) /\ Ai' := xor(HPWi', Aii')
       /\ Bi' := xor(H(IDi.PWi.Aii'),Ri')
       /\ Ci' := xor(H(Aii'.Ri'),H(H(Aii'.IDi).Kna'))
       /\ Di' := H(Aii'.Ri'.H(H(Aii'.IDi).Kna'))
       /\ PKi' := MUL(Ri'.G)
%%%%%%%%%%Login & Authentication phase
       /\ T1' := new() /\ Rj' := new()
       /\ M1' := H(H(H(Aii'.IDi).Kna').H(Aii'.IDi))
       /\ SND(M1'.T1'.MUL(Ri'.G).ADD(M1'.MUL(Ri'.MUL(Rj'.G))))
       /\ witness(P,MC,p_mc_m1,Kna')
3. State = 2 /\ RCV(xor(Bj',H(H(Aii'.IDi).Kna')). T2'.H(H(IDj.Rj').Bj'.T2')) =|>
State' := 3 /\ SK' := H(H(Aii'.IDi).H(IDj.Rj').H(H(Aii'.IDi).Kna').Bj')
         /\ request(P,MC,mc_p_bj,Bj')
end role
```

Figure 9. High-Level Protocol Specification Language (HLPSL) syntax of patient.



```
role medical(P, MC, TA : agent, SKmcna : symmetric_key, H: hash_func, SND,
RCV : channel(dy))

played_by MC
def=
local State: nat,
   MUL, ADD : hash_func,
   HIDi, IDi, PWi, Aii, Xi, Kna, Ri, HPWi, Ai, Bi, Ci, Di, PKi, G, T1, M1  : text,
   IDj, Rj, PIDj, PKj, Rna, Sna, CERTj,Ei,Fi,Bj,T2, Mamc : text,
   SK: text
const sp1, sp2, sp3, sp4, p_mc_m1, mc_p_bj: protocol_id
init State := 0
transition

1. State = 0 /\ RCV(start) =|>
  State' := 1 /\ Rj' := new()
        /\ PIDj' := H(IDj.Rj')
        /\ PKj' := MUL(Rj'.G)
        /\ SND({PIDj'.PKj'}_SKmcna)
        /\ secret({Rj'},sp2,{MC})
2. State = 1 /\ RCV({ADD(H(H(IDj.Rj').Rna').MUL(Sna.MUL(Rj'.G)))}_SKmcna)
=|>
State' := 2
3. State = 2
/\ RCV(H(H(H(Aii'.IDi).Kna').H(Aii'.IDi)).T1'.MUL(Ri'.G).ADD(H(H(H(Aii'.IDi).K
na').H(Aii'.IDi)).MUL(Ri'.MUL(Rj'.G)))) =|>
State' := 3 /\ Bj' := new()
         /\ Ei' := xor(Bj',H(H(Aii'.IDi).Kna'))
      /\ T2' := new()
      /\ Mamc' := H(H(IDj.Rj').Bj'.T2')
      /\ SK' := H(H(Aii'.IDi).H(IDj.Rj').H(H(Aii'.IDi).Kna').Bj')
         /\ SND(Ei'. Mamc'.T2')
         /\ witness(MC,P,mc_p_bj,Bj')
      /\ request(MC,P,p_mc_m1,Kna')
end role
```

Figure 10. HLPSL syntax of medical center.

```
role admin(P,MC,TA : agent, SKpna, SKmcna : symmetric_key, H: hash_func, SND,
RCV : channel(dy))

played_by TA
def=
local State: nat,
   MUL, ADD : hash_func,
   HIDi, IDi, PWi, Aii, Xi, Kna, Ri, HPWi, Ai, Bi, Ci, Di, PKi, G, T1, M1  : text,
   IDj, Rj, PIDj, PKj, Rna, Sna, CERTj,Ei,Fi,Bj,T2, Mamc : text,
   SK: text
const sp1, sp2, sp3, sp4, p_mc_m1, mc_p_bj: protocol_id
init State := 0
transition

1. State = 0 /\ RCV({H(Aii'.IDi)}_SKpna) =|>
State' := 1 /\ Kna' := new() /\ Xi' := H(H(Aii'.IDi).Kna')
      /\ SND({Xi'}_SKpna)
         /\ secret({Kna'},sp3,{TA})

2. State = 1 /\ RCV({H(IDj.Rj').MUL(Rj'.G)}_SKmcna) =|>
State' := 2 /\ Rna' := new()
      /\ CERTj' := ADD(H(H(IDj.Rj').Rna').MUL(Sna.MUL(Rj'.G)))
      /\ secret({Rna',Sna},sp4,{TA})
      /\ SND({CERTj'}_SKmcna)

end role
```

Figure 11. HLPSL syntax of network administrator.

```
role session(P, MC, TA : agent, SKpna, SKmcna : symmetric_key, H: hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
patient(P, MC, TA, SKpna, H, SN1, RV1)
/\ medical(P, MC, TA, SKmcna, H, SN2, RV2)
/\ admin(P, MC, TA, SKpna, SKmcna, H, SN3, RV3)
end role

role environment()
def=
const p, mc, ta : agent,
skpna, skmcna: symmetric_key,
h,mul,add: hash_func,
idi,idj: text,
p_mc_m1, mc_p_bj: protocol_id,
sp1,sp2,sp3,sp4: protocol_id

intruder_knowledge = {p,mc,ta,idi,idj,h}
composition
session(p,mc,ta, skpna, skmcna,h)/\session(i,mc,ta, skpna, skmcna,h)
/\session(p,i,ta, skpna, skmcna,h)
/\session(p,mc,i, skpna, skmcna,h)

end role

goal
secrecy_of sp1, sp2, sp3, sp4
authentication_on p_mc_m1, mc_p_bj
end goal

environment()
```

Figure 12. HLPSL syntax of session and environment.

```
% OFMC

% Version of 2006/02/13

SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL

  /home/span/span/testsuite/results/block.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 5.88s
  visitedNodes: 1040 nodes
  depth: 9 plies
```

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/block.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.07 seconds
  Computation: 0.00 seconds
```

Figure 13. Automated Validation of Internet Security Protocols and Applications (AVISPA) analysis result using OFMC and CL-AtSe.

7. Performance Analysis

In this section, we analyze the computation and communication costs of the proposed protocol compared with related schemes [25,26].

7.1. Computation Cost

Referring to the work in [41–43], we compare computation costs during authentication phase for the proposed system with related schemes [25,26].

- T_{bp} : The computation time of a bilinear pairing operation ≈ 4.211 ms.
- T_{bp-sm} : The computation time of a scalar multiplication operation on bilinear pairing ≈ 1.709 ms.
- T_{bp-ad} : The computation time of a point addition operation on bilinear pairing ≈ 0.0071 ms.
- T_{ec-sm} : The computation time of a scalar multiplication operation on elliptic curve cryptography ≈ 0.442 ms.
- T_{ec-ad} : The computation time of a point addition operation on elliptic curve cryptography ≈ 0.0018 ms.
- T_{ec-enc} : The computation time of a encryption with elliptic curve cryptography ≈ 0.5102 ms.
- T_{ec-dec} : The computation time of a decryption with elliptic curve cryptography ≈ 0.7399 ms.
- T_h : The computation time of a one-way hash function operation ≈ 0.0001 ms.
- T_{exp} : The computation time of an exponentiation operation ≈ 3.886 ms.

Table 3 shows computation costs of the proposed scheme with related schemes [25,26]. In Liu et al.'s scheme [25], a client computes $\{T = tP, T^{'} = tQ_{AP}\}$ with multiplication on bilinear pairing, $\{I^{'}\}$ with addition on bilinear pairing, $\{r\}$ with exponential function, $\{U = kS_2 - vs_1Q_2\}$ with two multiplication and one addition on bilinear pairing, and $\{v, key, MAC_{key}(v)\}$ with hash function. Then, a application provider computes $\{T\}$ with multiplication on bilinear pairing, $\{I\}$ with addition on bilinear pairing, $\{v, key, MAC_{key}(v)\}$ with hash function, $\{r\}$ with one bilinear pairing operation, one multiplication on bilinear pairing, and one exponential function.

Table 3. Computation costs of the proposed scheme with related schemes.

	Liu et al. [25]	**Renuka et al. [26]**	**Proposed**
Patient/Client	$4T_{bp-sm} + 2T_{bp-ad} + T_{exp} + 3T_h \approx 10.8643$ ms	$3T_{ec-sm} + 10T_h \approx 1.327$ ms	$T_{ec-enc} + 7T_h \approx 0.5109$ ms
Medical center	$2T_{bp-sm} + T_{bp-ad} + T_{exp} + T_{bp} + 3T_h \approx 11.5863$ ms	$3T_{ec-sm} + 5T_h \approx 1.3265$ ms	$T_{ec-dec} + 3T_h \approx 0.7402$ ms

In Renuka et al.'s scheme [26], a user computes $\{V_i, A_i, F_i, sk\}$ with two hash functions, $\{R_i, E_i, E_s\}$ with multiplication on ECC, $\{D_i, H_i\}$ with one hash function. Moreover, in the registration phase, a server computes $H(B_i)$ and stores it in memory. After that, in authentication phase, the server extracts the $H(B_i)$. Thus, we do not include $H(B_i)$ in the operation. Then, server computes $\{ID_i, h(x \oplus ID_i), h(C_i||T_1||E_i||H(B_i)), sk, H_i\}$ with one hash function, $\{E_i, R_s, E_s\}$ with multiplication on ECC.

In the proposed scheme, a patient computes $\{HPW_i, r_i, x_i, D_i^*, M_{ap}, M_{amc}^*, SK_{ij}\}$ with hash function, M_1 with ECC encryption. Moreover, the medical center computes $\{M_1 - r_j \cdot PK_i\}$ with ECC decryption, $\{M_{ap}^*, M_{amc}, SK_{ij}\}$ with hash function. As a result, we provide better efficiency than existing schemes [25,26] because our scheme uses only hash function and ECC encryption/decryption.

7.2. Communication Cost

We compare communication costs during authentication phase for the proposed system with related schemes [25,26]. We assume that the ECC-based encryption (EN_{ecc}), timestamp (T), identity (I) hash function (H), and message authentication code (MAC) are 320, 32, 128, 160, and 160 bits [44,45], respectively. We also define that additive groups on super singular (G_1), and additive group (G) are 1024 and 320 bits [44,45], respectively. Table 4 shows communication costs of the proposed scheme with related schemes [25,26].

In Liu et al.'s scheme [25], transmitted messages $\{v, U, t_c, T', I'\}$ and $\{MAC\}$. U, T', and I' are elements of G_1. Moreover, v is the element of hash function, t_c is a timestamp, and MAC is the element of message authentication code. In Liu et al.'s scheme, transmitted messages require (160 + 1024 + 32 + 1024 + 1024 = 3264 bits) and (160 bits), respectively.

In Renuka et al.'s scheme [26], transmitted messages $\{D_i, R_i, F_i, T_1\}$ and $\{R_s, H_i, T_2\}$. R_i and R_s are elements of G. D_i, F_i, and H_i are elements of hash function. And also, T_1 and T_2 are elements of timestamp. In Renuka et al.'s scheme, transmitted messages require (160 + 320 + 160 + 32 = 672 bits) and (320 + 160 + 32 = 512 bits), respectively.

In the proposed scheme, transmitted messages $\{M_1, M_{op}, T_1\}$ and $\{E_i, M_{acm}, T_2\}$. M_{op}, M_{acm}, and E_i are the elements of hash function and M_1 is the element of ECC-based encryption. And also, T_1 and T_2 are the elements of timestamp. In proposed scheme, transmitted messages require (320 + 160 + 32 = 512 bits) and (160 + 160 + 32 = 352 bits), respectively. Consequently, we provide better efficiency than related schemes [25,26] because our scheme uses hash function, timestamp, and ECC-based encryption/decryption.

Table 4. Communication costs of the proposed scheme with related schemes.

	Liu et al. [25]	**Renuka et al. [26]**	**Proposed**
Patient/Client	$H + 3G_1 + T = 3264$ bits	$2H + G + T = 672$ bits	$EN_{ecc} + H + T = 512$ bits
Medical center	$MAC = 160$ bits	$G + H + T = 512$ bits	$2H + T = 352$ bits

7.3. Security Properties

Table 5 shows the comparison between the security properties of the proposed scheme and related schemes [25,26]. Our scheme guarantees perfect forward secrecy, anonymity, and mutual authentication, and avoids the single point of failure and bottleneck. In addition, the proposed scheme has the resistance of impersonation, session key disclosure, replay, and privileged insider attacks.

Table 5. Security properties of the proposed scheme with related schemes.

	Liu et al. [25]	**Renuka et al. [26]**	**Proposed**
Impersonation attack	X	O	O
Session key disclosure attack	X	O	O
Perfect forward secrecy	X	O	O
Replay attack	O	O	O
Privileged insider attack	X	O	O
Single point of failure	X	X	O
Anonymity	O	O	O
Mutual authentication	X	O	O
Bottleneck	X	X	O

8. Conclusions

With the rapid development of the EHR system, medical centers obtain patient's health records to provide accurate medical services through medical wearable sensors. However, these health records contain sensitive information of patients, it is necessary to ensure the security from leakage or counterfeiting in the process of storing and sharing information. Furthermore, traditional protocols for the EHR system cannot prevent the single point of failure, and the EHR system should consider storage overload problems because of the large amounts of EHR data and scalability of the system. In this paper, we proposed the secure protocol for cloud-assisted EHR system using blockchain to resolve these problems. The proposed scheme presented detailed phases for six phases such as registration, authentication, smart contract uploading, EHR storing, EHR requesting, and log transaction uploading. We proved that the proposed scheme prevents various attacks and provides secure mutual authentication, anonymity, and perfect forward secrecy. We demonstrated the safety of the proposed scheme against MITM and replay attacks using AVISPA simulation. Furthermore, we proved that the proposed scheme ensures a secure mutual authentication between patient and medical server using BAN logic. We compared the security features and performance of the proposed scheme with some existing schemes. We proved that our scheme provides better safety and efficiency than related schemes. Therefore, the proposed EHR system can be suitable for the practical healthcare system for EHRs because it is more secure and efficient than other related schemes. In the future, we aim to develop a set of realistic simulations to test the protocol. If these practical simulations are available, it will help to develop a secure protocol for the cloud-assisted EHR system using blockchain.

Author Contributions: Conceptualization, M.K.; Formal analysis, S.Y., J.L. and Y.P. (YoHan Park); Software, M.K., and J.L.; Supervision, Y.P. (YoungHo Park); Validation, S.Y., J.L., Y.P. (YoHan Park) and Y.P. (YoungHo Park); Writing–original draft, M.K.;Writing–review and editing, S.Y., Y.P. (YoHan Park) and Y.P. (YoungHo Park). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning under Grant 2017R1A2B1002147 and in part by the BK21 Plus project funded by the Ministry of Education, Korea, under Grant 21A20131600011.

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

References

1. Greenhalgh, T.; Hinder, S.; Stramer, K.; Bratan, T.; Russell, J. Adoption, non-adoption, and abandonment of a personal electronic health record: Case study of healthspace. *Br. Med. J.* **2010**, *341*, c5814. [CrossRef] [PubMed]
2. Tang, F.; Ma, S.; Xiang, Y.; Lin, C. An efficient authentication scheme for blockchain-based electronic health records. *IEEE Access* **2019**, *7*, 41678–41689. [CrossRef]
3. Fan, K.; Ren, Y.; Wang, Y.; Li, H.; Yang, Y. Blockchain-based efficient privacy preserving and data sharing scheme of content-centric network in 5G. *IET Commun.* **2018**, *12*, 527–532. [CrossRef]
4. Dwivedi, A.D.; Malina, L.; Dzurenda, P.; Srivastava, G. Optimized blockchain model for internet of things based healthcare applications. In Proceedings of the 42nd International Conference on Telecommunications and Signal Processing (TSP), Budapest, Hungary, 1–3 July 2019; pp. 135–139.
5. Dwivedi, A.; Srivastava, G.; Dhar, S.; Singh, R. A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors* **2019**, *19*, 326. [CrossRef]
6. Rathee, G.; Sharma, A.; Iqbal, R.; Aloquaily, M.; Jaglan, N.; Kumar, R. A blockchain framework for securing connected and autonomous vehicles. *Sensors* **2019**, *19*, 3165. [CrossRef]
7. Tseng, L.; Wong, L.; Otoum, S.; Aloqaily, M.; Othman, J.B. Blockchain for managing heterogeneous internet of things: A perspective architecture. *IEEE Netw.* **2020**, *34*, 16–23. [CrossRef]
8. Kuo, T.T.; Rojas, H.Z.; Ohno-Machado, L. Comparison of blockchain platforms: A systematic review and healthcare examples. *J. Am. Med. Inform.* **2019**, *26*, 462–478. [CrossRef]
9. Chukwu, E.; Garg, L. A systematic review of blockchain in healthcare: Frameworks, prototypes, and implementations. *IEEE Access* **2020**, *8*, 2169–3536. [CrossRef]
10. Hyperledger: Open Source Blockchain Technologies. Available online: https://www.hyperledger.org/ (accessed on 8 March 2020).
11. Ma, H.; Huang, E.X.; Lam, K.Y. Blockchain-based mechanism for fine-grained authorization in data crowdsourcing. *Future Gener. Comput. Syst.* **2020**, *106*, 121–134. [CrossRef]
12. Thwin, T.T.; Vasupongayya, S. Blockchain-based access control model to preserve privacy for personal health record systems. *Secur. Commun. Netw.* **2019**, *2019*, 8315614. [CrossRef]
13. Zhu, X.; Shi, J.; Lu, C. Cloud health resource sharing based on consensus-oriented blockchain technology: Case study on a breast tumor diagnosis service. *J. Med. Internet Res.* **2019**, *21*, e13767. [CrossRef] [PubMed]
14. Li, M.; Yu, S.; Ren, K.; Lou, W. Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. In Proceedings of the 6th International ICST Conference on Security and Privacy in Communication Networks (SecureComm 2010), Singapore, 7–9 September 2010; pp. 89–106.
15. Ridhawi, I.A.; Otoum, S.; Aloquaily, M.; Jararweh, Y.; Baker, T. Providing secure and reliable communication for next generation networks in smart cities. *Sustain. Cities Soc.* **2020**, *56*, 102080. [CrossRef]
16. Park, Y.; Park, Y. A selective group authentication scheme for IoT-based medical information system. *J. Med. Syst.* **2017**, *41*, 48. [CrossRef] [PubMed]

17. Kaur, H.; Alam, M.A.; Jameel, R.; Mourya, A.K.; Chang, V. A proposed solution and future direction for blockchain-based heterogeneous medicare data in cloud environment. *J. Med. Syst.* **2018**, *42*, 156. [CrossRef]
18. Nagasubramanian, G.; Sakthivel, R.K.; Patan, R.; Gandomi, A.H.; Sankayya, M.; Balusamy, B. Securing e-health records using keyless signature infrastructure blockchain technology in the cloud. *Neural Comput. Appl.* **2020**, *32*, 639–647. [CrossRef]
19. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [CrossRef]
20. Lee, J.Y.; Yu, S.J.; Park, K.S.; Park, Y.H.; Park, Y.H. Secure three-factor authentication protocol for multi-gateway IoT environments. *Sensors* **2019**, *19*, 2358. [CrossRef]
21. AVISPA. Automated Validation of Internet Security Protocols and Applications. Available online: http://www.avispa-project.org/ (accessed on 8 March 2020).
22. SPAN: A Security Protocol Animator for AVISPA. Available online: http://www.avispa-project.org/ (accessed on 8 March 2020).
23. Kumar, P.; Lee, S.G.; Lee, H.J. E-SAP: Efficient-strong authentication protocol for healthcare applications using wireless medical sensor networks. *Sensors* **2012**, *12*, 1647. [CrossRef]
24. Wu, Z.Y.; Chen, L.; Wu, J.C. A reliable RFID mutual authentication scheme for healthcare environments. *J. Med. Syst.* **2013**, *37*, 9917. [CrossRef]
25. Liu, J.; Zhang, Z.; Chen, X.; Kwak, K.S. Certificateless remote anonymous authentication schemes for wireless body area networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 332–342. [CrossRef]
26. Renuka, K.; Kumari, S.; Li, X. Design of a secure three-factor authentication scheme for smart healthcare. *J. Med. Syst.* **2019**, *43*, 133. [CrossRef] [PubMed]
27. Pandey, P.; Litoriya, R. Securing e-health networks from counterfeit medicine penetration using blockchain. *Wirel. Pers. Commun.* **2020**. [CrossRef]
28. Agbo, C.C.; Mahmoud, Q.H. Comparison of blockchain frameworks for healthcare applications. *Internet Technol. Lett.* **2019**, *2*, e122. [CrossRef]
29. 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]
30. Wang, Y.; Zhang, A.; Zhang, P.; Wang, H. Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain. *IEEE Access* **2019**, *7*, 136704–136719. [CrossRef]
31. Chen, Y.; Ding, S.; Xu, Z.; Zheng, H.; Yang, S. Blockchain-based medical records secure storage and medical service framework. *J. Med. Syst.* **2019**, *43*, 5. [CrossRef]
32. Cheng, X.; Chen, F.; Xie, D.; Sun, H.; Huang, C. Design of a secure medical data sharing scheme based on blockchain. *J. Med. Syst.* **2020**, *44*, 52. [CrossRef]
33. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
34. Li, C.T.; Lee, C.C.; Weng, C.Y.; Chen, S.J. A secure dynamic identity and chaotic maps based user authentication and key agreement scheme for e-Healthcare systems. *J. Med. Syst.* **2016**, *40*, 233. [CrossRef]
35. Kocher, P.; Jaffe, J.; Jun, B. Differential power analysis. In Proceedings of the Annual International Cryptology Conference (CRYPTO), Santa Barbara, CA, USA, 15–19 August 1999; pp. 388–397.
36. Yu, S.J.; Lee, J.Y.; Lee, K.K.; Park, K.S.; Park, Y.H. Secure authentication protocol for wireless sensor networks in vehicular communications. *Sensors* **2018**, *18*, 3191. [CrossRef]
37. Park, Y.H.; Park, Y.H. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. *Sensors* **2016**, *16*, 2123. [CrossRef] [PubMed]
38. Novo, O. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet Things J.* **2018**, *5*, 1184–1195. [CrossRef]
39. Lu, N.; Zhang, Y.; Shi, W.; Kumari, S.; Choo, K.K.R. A secure and scalable data integrity auditing scheme based on hyperledger fabric. *Comput. Secur.* **2020**, *92*, 101741. [CrossRef]
40. Von Oheimb, D. The high-level protocol specification language HLPSL developed in the EU project avispa In Proceedings of the APPSEM 2005 Workshop, Tallinn, Finland, 13–15 September 2005; pp. 1–2.
41. Lei, A.; Cruickshank, H.; Cao, Y.; Asuquo. P.; Ogah, C.P.A.; Sun, Z. Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. *IEEE Internet Things J.* **2017**, *4*, 1832–1843. [CrossRef]

42. Islam, S.K.H.; Obaidat, M.S.; Vijayakumar, P.; Abdulhay, E.; Li, F.; Reddy, M.K.C. A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs. *Future Gener. Comput. Syst.* **2018**, *84*, 216–227. [CrossRef]
43. Zhang, Q.; Wang, X.; Yuan, J.; Liu, L.; Wang, R.; Huang, H.; Li, Y. A hierarchical group key agreement protocol using orientable attributes for cloud computing. *Inform. Sci.* **2019**, *480*, 55–69. [CrossRef]
44. Lee, H.; Lee, D.; Moon, J.; Jung, J.; Kang, D.; Kim, H.; Won, D. An improved anonymous authentication scheme for roaming in ubiquitous networks. *PLoS ONE* **2018**, *13*, e0193366. [CrossRef]
45. Ying, B.; Nayak, A. Lightweight remote user authentication protocol for multi-server 5G networks using self-certified public key cryptography. *J. Netw. Comput. Appl.* **2019**, *131*, 66–74. [CrossRef]