Received September 25, 2020, accepted October 16, 2020, date of publication October 21, 2020, date of current version November 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3032680

Design of Secure Authentication Protocol for Cloud-Assisted Telecare Medical Information System Using Blockchain

SEUNGHWAN SON[1], JOONYOUNG LEE[1], MYEONGHYUN KIM[1], SUNGJIN YU[1], ASHOK KUMAR DAS[2], (Senior Member, IEEE), AND YOUNGHO PARK[1,3], (Member, IEEE)

[1]School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, South Korea
[2]Center for Security, Theory, and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad 500032, India
[3]School of Electronics Engineering, Kyungpook National University, Daegu 41566, South Korea

Corresponding author: Youngho Park (parkyh@knu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A3058605, in part by the Ripple Centre of Excellence Scheme, CoE in Blockchain (Sanction No. IIIT/Research and Development Office/Internal Projects/001/2019), IIIT Hyderabad, India, and in part by the Mathematical Research Impact Centric Support (MATRICS) project funded by the Science and Engineering Research Board (SERB), India, under Grant MTR/2019/000699.

ABSTRACT Telecare medical information system (TMIS) implemented in wireless body area network (WBAN) is convenient and time-saving for patients and doctors. TMIS is realized using wearable devices worn by a patient, and wearable devices generate patient health data and transmit them to a server through a public channel. Unfortunately, a malicious attacker can attempt performing various attacks through such a channel. Therefore, establishing a secure authentication process between a patient and a server is essential. Moreover, wearable devices have limited storage power. Cloud computing can be considered to resolve this problem by providing a storage service in the TMIS environment. In this environment, access control of the patient health data is essential for the quality of healthcare. Furthermore, the database of the cloud server is a major target for an attacker. The attacker can try to modify, forge, or delete the stored data. To resolve these problems, we propose a secure authentication protocol for a cloud-assisted TMIS with access control using blockchain. We employ ciphertext-policy attribute-based encryption (CP-ABE) to establish access control for health data stored in the cloud server, and apply blockchain to guarantee data integrity. To prove robustness of the proposed protocol, we conduct informal analysis and Burrows-Adabi-Needham (BAN) logic analysis, and we formally validate the proposed protocol using automated validation of internet security protocols and applications (AVISPA). Consequently, we show that the proposed protocol provides more security and has better efficiency compared to related protocols. Therefore, the proposed protocol is proper for a practical TMIS environment.

INDEX TERMS Attribute-based encryption, bilinear pairing, blockchain, cloud computing, mutual authentication and key agreement, telecare medical information system.

I. INTRODUCTION

Telecare medical information system (TMIS) implemented in wireless body area network (WBAN) is a rising service that enables doctors to diagnose patients remotely [1]. In a TMIS environment, WBAN nodes are wearable devices worn by a patient that generate the health data including the blood pressure, body temperature, and the heart rate. Then, these devices transmit health data to a server through a public channel. However, a malicious attacker can attempt performing various attacks including replay and impersonation attacks through a public channel. Therefore, a patient and a server must be securely authenticated each other [2]–[4]. Furthermore, wearable devices have limited storage power, and therefore, it is difficult to store an entire set of the health data generated in real time [5], [6]. Cloud computing can offer sufficient storage service for WBAN nodes. By means of cloud computing, patients can transmit their health data to

The associate editor coordinating the review of this manuscript and approving it for publication was Srinivas Sampalli.

the cloud server, and doctors can make diagnosis relying on the health data in the cloud server.

However, patients should be able to determine which doctors can access to their health data to get better TMIS service. Therefore, access control is an indispensable requirement in cloud-assisted TMIS environment. Attribute-based encryption (ABE) [7] is a widely-used encryption technique that provides fine-grained access control. Under other encryption methods, a plaintext is encrypted with a public key and a user who has the corresponding private key can decrypt the ciphertext. However, under ABE, a plaintext is encrypted under a set of attributes, and the users who have proper attribute sets can decrypt a ciphertext. ABE is categorized into key-policy ABE (KP-ABE) [8] and ciphertext-policy ABE (CP-ABE) [9]. In KP-ABE, users have attribute keys associated with their access structure. If an attribute set of a ciphertext satisfies the access structure of a user's key, the user can decrypt the ciphertext. In CP-ABE, each ciphertext is encrypted associated with an access structure set by an encrypter. A user can decrypt a ciphertext only if the user's attributes set satisfies the access structure of the ciphertext. Accordingly, the patients can determine access structures of their health data using CP-ABE, and therefore, CP-ABE is more proper for TMIS environment compared with KP-ABE.

Furthermore, the database of the cloud server can be a major target for an attacker because it is a centralized system [10]. If an attacker intrudes the database of the cloud server and modifies, forges, or deletes stored data, it can cause serious issues to patients. Blockchain technology [11] as a distributed ledger can be considered as a solution for the centralized problem assiociated with the cloud server. Under this technology, every transaction is recorded in the ledger, and ledgers are chained with hash values to form the blockchain. As every participant of the blockchain keeps the ledgers, an attacker cannot change the transactions on the blockchain. However, the public blockchain model [11] consumes an amount of computation cost because every node in the blockchain participates in a consensus process, and it can cause scalability problem. Therefore, numerous researchers utilized consortium blockchain for cloud-based medical environment [12]–[14]. In these schemes, the cloud server stores health data, and the related data such as a keyword, a hash, an address of the data are recorded on the blockchain. These schemes utilize the cloud server for data storage and apply consortium blockchain so that data integrity and scalablilty is guaranteed. However, these schemes [12]–[14] do not deal with the mutual authentication and key agreement process. Therefore, we design a secure authentication protocol in a cloud-assisted TMIS. Moreover, we adopt blockchain technology for data integrity of the cloud server and CP-ABE to realize access control for health data stored in the cloud server.

A. RESEARCH CONTRIBUTIONS

The main contributions of this paper are in the following manners:

- We propose a secure authentication protocol for a cloud-assisted TMIS with access control using blockchain. The cloud server stores the health data, and the blockchain stores the related data including a hash, an address, and an access tree of health data.
- We apply consortium blockchain to ensure data integrity and to provide scalability, and we adopt the CP-ABE to establish access control for health data. Patients establish access structure so only doctors who satisfy the access structure can access the patients' health data.
- We conduct informal analysis to demonstrate that the proposed protocol provides a variety of security features, and we perform BAN logic for proving that the proposed protocol attains mutual authentication.
- We employ AVISPA to demonstrate that the proposed protocol is safe. Moreover, we make a comparison of computation and communication costs, and security features between the proposed protocol and the related protocols.

B. PAPER ORGANIZATION

We present previous works related to our system in Section II, and we explain preliminaries in Section III. Section IV and Section V demonstrate the system model and the proposed scheme, respectively. Section VI analyzes the proposed protocol in terms of security. In Section VII, we make a performance comparison between the proposed protocol and the related protocols. Section VIII presents the conclusion of this paper.

II. RELATED WORK

In past decades, many researchers proposed secure authentication schemes for WBAN. Liu *et al.* [15] introduced an anonymous authentication for WBAN using bilinear pairing. However, Zhao [16] indicated that the scheme presented in [15] could not defend from stolen-verifier attack and guarantee user anonymity. Zhao suggested an authentication scheme using elliptic curve cryptosystem for efficiency. Nevertheless, Wang and Zhang [17] showed that the scheme presented in [16] used constant user identity, and it could not offer user anonymity. Wang and Zhang used bilinear pairing and developed an improved scheme. Wang and Zhang asserted that their scheme ensures user anonymity and resists impersonation attack. But, Jiang *et al.* [18] indicated that the scheme presented in [17] could not defend from impersonation attack and proposed an enhanced scheme which could resist the user impersonation attack and provide mutual authentication. Mwitende *et al.* [19] indicated that the scheme presented in [18] was centralized and was not able to offer data verifiability. Mwitende *et al.* utilized certificateless ring signature in blockchain-based WBANs to enable decentralization and data verifiability. In recent years, Liu *et al.* [20] suggested a two-layer authentication scheme that provides various security features. Chen and Peng [21] proposed authentication scheme using asymmetric bilinear pairing. However, the schemes in [20] and [21] have same vulnerabilities with the scheme proposed in [18]. Khatoon *et al.* [22] also suggested a privacy-preserved key agreement protocol in a TMIS environment. However, Nikooghadam and Amintoosi [23] indicated that the scheme presented in [22] was prone to known session-specific temporary information attack and could not ensure perfect forward secrecy. Chatterjee *et al.* [24] suggested an authentication scheme with access control in TMIS environment. However, including the scheme in [24], the schemes designed in [15]–[22] were not cloud-based. Therefore, they encountered difficulties in storing health-related data.

In recent years, many cloud-assisted TMIS authentication schemes were introduced. Chen *et al.* [25] suggested an authentication protocol in a cloud-based medical environment. However, Chiou *et al.* [26] indicated that the scheme presented in [25] failed to fulfill the telemedicine and could not support patient anonymity. Chiou *et al.* compensated the security flaws of the scheme presented in [25], and suggested an improved scheme in a telemedicine environment. However, Mohit *et al.* [27] indicated that the scheme presented in [26] could not guarantee patient anonymity and resist stolen smart device attack. Mohit *et al.* suggested a standard mutual authentication scheme in the same environment. Nevertheless, Li *et al.* [28] revealed that the scheme presented in [27] was not able to support patient untraceability and anonymity. Li *et al.* proposed an enhanced scheme that resolved the flaws of the scheme presented in the scheme [27]. Nevertheless, these schemes [25]–[28] could not guarantee data integrity and realize fine-grained access control of health data.

In recent years, numerous researchers applied blockchain technology and ABE to the cloud-based medical environment. Guo *et al.* [29] employed blockchain technology in cloud-based EHR system. Guo *et al.* also utilized multi-authorities to resist collusion attack, and attribute-based signature for hiding information about patients. Guo *et al.* [30] proposed blockchain-based ABE protocol with multi-authorities in telemedicine system. However, in their scheme, patients should keep the attribute keys on their own, and it is not suitable for a real environment. Wang and Song [13] used ABE and blockchain to bulid a cloud computing based EHR sharing system. Wang and Song utilized ID-based cryptosystem and attribute-based cryptosystem for the medical data integrity and confidentiality. Yang *et al.* [31] utilized decentralized attribute-based signature and outsourced decryption ABE to improve the efficiency of the scheme presented in [13]. Their scheme consumes less computation cost compared to the scheme presented in the scheme [13]. However, these schemes [13], [29]–[31] do not deal with the mutual authentication and session key agreement process.

III. PRELIMINARY

We describe the preliminaries to facilitate readability of this paper.

A. ACCESS STRUCTURE

We utilize access tree presented in [9] as the access structure. Let Γ be an access tree, then Γ contains $(\nu, n_\nu, v_\nu, par(\nu), ind(\nu))$. To explain each notation, ν denotes a node of Γ. If ν is an internal node, then ν is a threshold gate represented as *AND* and *OR*, and if ν is a leaf node, then ν is an attribute. n_ν denotes the number of childnodes of ν, v_ν denotes a threshold value of ν, $par(\nu)$ denotes a parent node of ν, and $ind(\nu)$ is unique index of ν. When ν is an internal node and if $n_\nu = v_\nu$, then ν is an *AND* gate, and If $v_\nu = 1$, then ν is an *OR* gate. If ν is a leaf node, ν is an attribute and $v_\nu = 1$. To satisfy the access tree Γ with set of attributes $att(k)$, $att(k)$ must satisfy the threshold gate of the root node γ of Γ. In the first case, if γ is an attribute and the corresponding key is in $att(k)$, it satisfies access tree. In the second case, if γ is a threshold gate with childnodes being attributes, then if $att(k)$ satisfies the threshold gate of γ, it satisfies access tree. In the other cases such as γ is a threshold gate with childnodes are also threshold gates, it can be solved with applying the method of the second case recursively.

B. BILINEAR PAIRING

Let G_1 and G_2 be cyclic groups with a large prime order q, and they are an additive group and a multiplicative group, respectively. A bilinear map $\check{e} : G_1 \times G_1 \rightarrow G_2$ satisfies the following conditions [32]:

- **Bilinearity**: $\forall P, Q \in G_1$, and $\forall a, b \in Z_p^*$, $\check{e}(aP, bQ) = \check{e}(P, Q)^{ab}$.
- **Non-degeneracy**: $\exists P, Q \in G_1$, such that $\check{e}(P, Q) \neq 1_{G_1}$, where 1_{G_1} is the identity element in G_1.
- **Efficiency** : $\forall P, Q \in G_1$, $\check{e}(P, Q)$ can be calculated in polynomial time.

C. BLOCKCHAIN

To be suitable for a cloud-assisted TMIS environment, blockchain network should provide scalability and have decentralized characteristics. Blockchain can be categorized into public blockchain, private blockchain, and consortium blockchain [33]. Public blockchain such as bitcoin has difficulty to apply it to a TMIS environment because the whole nodes should participate in a consensus process. It demands an amount of computation cost and encounters the scalability problem. Private blockchain is managed by an authorized organization. Therefore, it requires low computation cost and provides scalability but it has centralized characteristics [34]. Consortium blockchain is partially decentralized because it is managed by several consortium nodes that consent transactions in blockchain. In consortium blockchain, only authorized nodes can get access to ledgers or upload transactions to the blockchain. Consortium blockchain is decentralized compared to private blockchain and provide scalability compared to public blockchain. Therefore, consortium blockchain is deemed suitable for a cloud-assisted TMIS environment.

D. ADVERSARY MODEL

We consider the widely-used "Dolev-Yao (DY) threat model" [35]–[37] for analyzing security of the proposed authentication protocol. The capabilities of an adversary model can be defined in the following manner:

- An attacker has the entire control of the messages transmitted through a public channel. The attacker can eavesdrop, modify, forge, and delete messages.
- An attacker can obtain the smart card of a patient. The attacker can attempt the power analysis attack [38], [39] to get the stored values in the smart card.
- An attacker can guess either the identity or the password of a patient, but cannot guess both of them simultaneously.
- An attacker can attempt diverse attacks such as replay, man-in-the middle (MITM), session key disclosure, impersonation attacks, etc. [40].

IV. SYSTEM MODEL

We describe a system model of the cloud-assisted TMIS with access control using blockchain in Figure 1. The proposed model comprises five entities: a trusted authority (TA), a cloud server, a patient, a doctor, and blockchain. TA is defined as a trusted entity and initializes the system. The cloud server stores health data of patients and diagnosis results provided by doctors and uploads transactions about the stored data. A patient uploads the personal health data encrypted with ABE for being diagnosed. If a doctor satisfies the access tree of the health data stored in the cloud server, the doctor can request the cloud server to get the health data. Health centers and local hospitals organize the consortium blockchain. Patients and doctors can read the ledgers of the blockchain and the cloud server can upload transactions to the blockchain. The detailed descriptions of the entities are as below.

- **TA**: TA is a trusted entity that corresponds to a higher level of institution compared with general hospitals and health centers. TA acts as a registration and key generation center for participants including patients, doctors, and the cloud server.
- **Cloud Server**: The cloud server has a sufficient storage ability to store the health data of patients and doctors. However, the cloud server is a centralized storage system, and therefore, it can be a major target for a malicious attacker. The malicious attacker can attempt to access the data stored in the database and tamper or forge it. Therefore, as soon as the data upload process is completed, the cloud server transmits the address, hash, and the access tree of the data to the blockchain. Consequently, doctors and patients can verify that the data from the cloud server are not corrupted using the blockchain.
- **Patient**: Patients wearing medical devices transmit their health data to the cloud server via a public channel. Therefore, a patient has to authenticate to the cloud before uploading the data. During the authentication process, an attacker must not be able to obtain the personal data of the patient using transmitted messages. In addition, the patient data stored in the cloud server must be protected. Therefore, the patient sets the access tree of the data and sends the encrypted data with the access tree to the cloud server. Furthermore, when patients checkup their diagnosis, they can verify whether the data are corrupted using the hash of the data obtained from the blockchain.

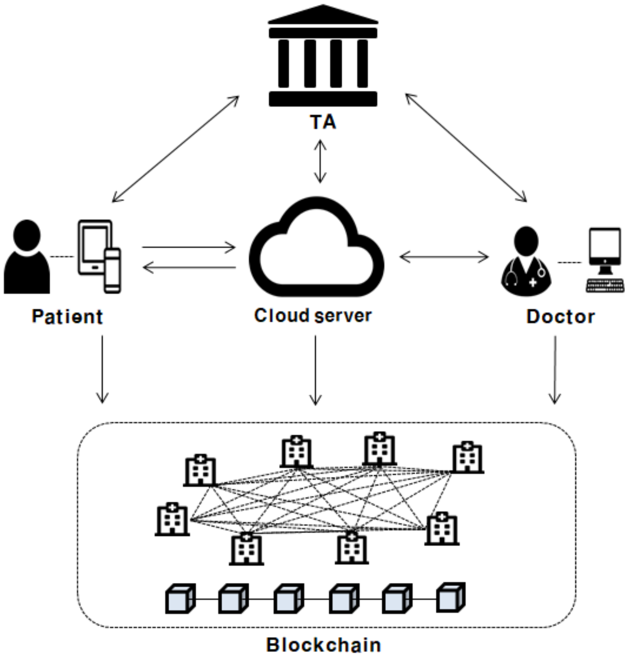


FIGURE 1. System model of cloud-assisted TMIS using blockchain.

- **Doctor**: Doctors can request the patient health data from the cloud server, and doctors should be able to obtain the health data appropriate for their capabilities. Therefore, each doctor is issued attribute keys from TA related to their field, location, affiliation, or etc. After receiving attribute keys from TA, the doctor's identity and attributes are stored in the blockchain. The doctor can read the blockchain and obtain a hash and access tree of health data stored in the cloud server. If a doctor satisfies the access tree of the data, the doctor can request the data from the cloud server. After obtaining the data from the cloud server, the doctor can verify whether the data are corrupted using the hash of the data. Thereafter, the doctor can decrypt the data using attribute keys and upload the diagnostic results to the cloud server.
- **Blockchain**: Consortium blockchain is realized in the proposed scheme. Health centers and local hospitals constitute consortium blockchain. Blockchain transactions contain the public key of the data uploader, address, hash, and the access tree of the data, which are related with the data stored in the cloud server. Consortium nodes consent these transactions using the Proof-of-Authority (PoA) algorithm [41]. Only consortium nodes participate in the consensus process so that it consumes

low computation cost and provides scalability. In the blockchain, doctors and patients can read ledgers, and the cloud server can upload transactions. Furthermore, the doctors' identities and attributes are managed within the blockchain. If a doctor requests the data from the cloud server, then it confirms whether the doctor's attributes satisfy the access tree of the requested data through the blockchain. If the condition is satisfied, the cloud server sends the data to the doctor.

V. PROPOSED SCHEME

We propose a secure authentication scheme for a cloud-assisted TMIS with access control using blockchain. The proposed protocol includes initialization, registration, key generation, authentication, data upload, treatment, and checkup. Table 1 represents the notations of the proposed protocol. In order to protect replay attack, we use both random numbers (secrets) along with the current timestamps generated by the entities in TMIS. It is then assumed that the entities in the network will be synchronized with their clocks. It becomes a typical assumption that is applied in many recent authentication protocols [42]–[45], [46]–[49], [50]–[52].

TABLE 1. Notations and their meanings.

Notation	Description
P_i, D_k	i-th patient and k-th doctor
CS	The cloud server
ID_i, PW_i	Identity and password of P_i
SC_i	Smart card of P_i
r_i, r_{CS}	Random numbers generated by P_i and CS
s_{TA}, α	Secret keys of TA
$\breve{e}$	Bilinear map $\breve{e} : G_1 \times G_1 \rightarrow G_2$
PK_i, PK_{CS}	Public keys of P_i and CS
$h(\cdot)$	Hash function $\{0,1\}^* \rightarrow Z_q$
$H(\cdot)$	Map-to-point hash function $\{0,1\}^* \rightarrow G_1$
ATT_k	Attributes of D_k
att_k	Attribute private keys of D_k
SID_i	Secret Identity of P_i
SK_{i-CS}	Session key between P_i and CS
Γ, γ	Access tree and the root node
$\oplus$	XOR operation
$\|\|$	Concatenation operation

A. INITIALIZATION

The system initialization phase is conducted by TA. TA generates G_1 as an additive cyclic group and G_2 as a multiplicative cyclic group with the same order q, $e : G_1 \times G_1 \rightarrow G_2$ as a bilinear map, and generates s_{TA}, $\alpha \in Z_q^*$, a generator $P \in G_1$, and hash functions $h : \{0,1\}^* \rightarrow Z_q$, $H : \{0,1\}^* \rightarrow G_1$. Then, TA generates a public key $PK_{TA} = s_{TA} * P$ where $s_{TA} * P$ denotes the "elliptic curve point (scalar) multiplication of the point P in G_1" and computes $Q = \frac{P}{s_{TA}}$ for generating attribute keys, and $\breve{e}(P,P)^{\alpha}$ for decryption. TA publishes $(e, G_1, G_2, PK_{TA}, P, Q, \breve{e}(P,P)^{\alpha}, q, h, H)$ and keeps (s_{TA}, α) as secret keys.

B. KEY GENERATION

In the key generation phase, D_k with attributes ATT_k is issued attribute keys from TA.

- **Step 1**: D_k with attributes ATT_k is in the hospital j that corresponds to a consortium node. Hospital j securely sends a request message (ID_j, ID_k, ATT_k) to TA for key generation.
- **Step 2**: After TA receives the message (ID_j, ID_k, ATT_k), TA generates a random $r_k \in Z_q^*$ and computes $s_k = h(ID_k||s_{TA})$, and $A_k = Q(\alpha + r_k)$ and for all $s \in ATT_k$, TA generates a random number $r_{k_s} \in Z_q^*$, and computes $A_{k_s} = r_k P + r_{k_s} H(s)$, and $A'_{k_s} = r_{k_s} P$. Then TA securely sends the doctor's private key s_k and attribute keys $att_k = (A_k, A_{k_s}, A'_{k_s})$ to hospital j.
- **Step 3**: Hospital j computes $PK_k = s_k P$, sends (s_k, att_k) to D_k and uploads (ID_k, PK_k, ATT_k) to the blockchain.

C. REGISTRATION

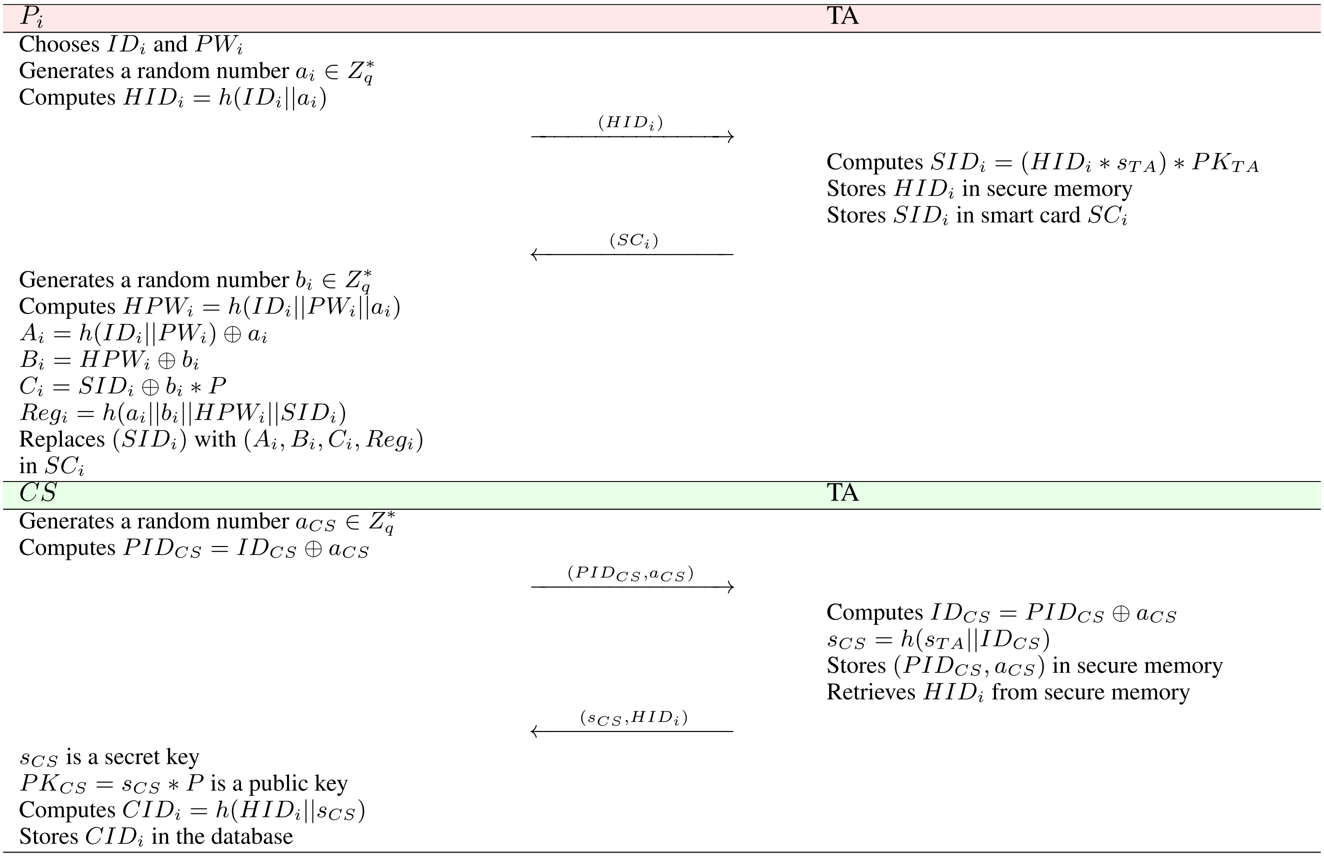
P_i and CS register to TA for participating in the network. Figure 2 represents the registration phase of the proposed protocol.

- **Patient registration**: P_i generates $a_i \in Z_q^*$, computes $HID_i = h(ID_i||a_i)$, and then transmits HID_i to TA securely. Then, TA computes $SID_i = (HID_i * s_{TA}) * PK_{TA}$, and stores HID_i in the secure memory. Thereafter, TA sends SC_i with $\{SID_i\}$ to P_i securely. P_i generates $b_i \in Z_q^*$, computes $HPW_i = h(ID_i||PW_i||a_i)$, $A_i = h(ID_i||PW_i) \oplus a_i$, $B_i = HPW_i \oplus b_i$, $C_i = SID_i \oplus b_i * P$, and $Reg_i = h(a_i||b_i||HPW_i||SID_i)$. Next, P_i replaces SID_i with (A_i, B_i, C_i, Reg_i) in SC_i.
- **Cloud server registration**: CS generates $a_{CS} \in Z_q^*$, computes $PID_{CS} = ID_{CS} \oplus a_{CS}$ and sends (PID_{CS}, a_{CS}) to TA securely. Then, TA computes $PID_{CS} \oplus a_{CS} = ID_{CS}$, and $s_{CS} = h(s_{TA}||ID_{CS})$. After that, TA stores (PID_{CS}, a_{CS}) and retrieves (HID_i) in the secure memory. Next, TA securely sends (s_{CS}, HID_i) to CS. Afterwards, CS computes $PK_{CS} = s_{CS} * P$ as a public key, computes $CID_i = h(HID_i||s_{CS})$, and stores CID_i in the database.

D. AUTHENTICATION

In the authentication phase, P_i and CS authenticate each other and establish a session key SK_{i-CS}. Figure 3 represents the authentication between P_i and CS.

- **Step 1**: P_i inputs ID_i^* and PW_i^* into SC_i. Then, SC_i computes $a_i^* = A_i \oplus h(ID_i^*||PW_i^*)$, $HID_i^* = h(ID_i^*||a_i^*)$, $HPW_i^* = h(ID_i^*||PW_i^*||a_i^*)$, $b_i^* = B_i \oplus HPW_i^*$, and $SID_i^* = C_i \oplus b_i^* * P$. Next, the SC_i checks whether $Reg_i \overset{?}{=} h(a_i^*||b_i^*||HPW_i^*||SID_i^*)$. If this equality holds, P_i is logged in SC_i.
- **Step 2**: SC_i generates a random secret $r_i \in Z_q^*$ and current timestamp T_1, and calculates the public key $PK_i = (a_i * r_i) * P$. Then, it computes $X_i = (a_i * r_i) * PK_{CS}$, $D_i = HID_i \oplus h(X_i)$, $L_{i1} = h(X_i||HID_i||T_1||ID_{CS})$, and $PID_i = SID_i * L_{i1}$. Thereafter, P_i sends the message (PK_i, D_i, PID_i, T_1) to CS via public channel.
- **Step 3**: After CS receives (PK_i, D_i, PID_i, T_1), checks the validity of received timestamp T_1 by the condition: $|T_1 - T_1^*| < \Delta T$, where T_1^* is the "time when the



P_i TA

Chooses ID_i and PW_i
Generates a random number $a_i \in Z_q^*$
Computes $HID_i = h(ID_i||a_i)$

$\xrightarrow{(HID_i)}$

Computes $SID_i = (HID_i * s_{TA}) * PK_{TA}$
Stores HID_i in secure memory
Stores SID_i in smart card SC_i

$\xleftarrow{(SC_i)}$

Generates a random number $b_i \in Z_q^*$
Computes $HPW_i = h(ID_i||PW_i||a_i)$
$A_i = h(ID_i||PW_i) \oplus a_i$
$B_i = HPW_i \oplus b_i$
$C_i = SID_i \oplus b_i * P$
$Reg_i = h(a_i||b_i||HPW_i||SID_i)$
Replaces (SID_i) with (A_i, B_i, C_i, Reg_i)
in SC_i

CS TA

Generates a random number $a_{CS} \in Z_q^*$
Computes $PID_{CS} = ID_{CS} \oplus a_{CS}$

$\xrightarrow{(PID_{CS}, a_{CS})}$

Computes $ID_{CS} = PID_{CS} \oplus a_{CS}$
$s_{CS} = h(s_{TA}||ID_{CS})$
Stores (PID_{CS}, a_{CS}) in secure memory
Retrieves HID_i from secure memory

$\xleftarrow{(s_{CS}, HID_i)}$

s_{CS} is a secret key
$PK_{CS} = s_{CS} * P$ is a public key
Computes $CID_i = h(HID_i||s_{CS})$
Stores CID_i in the database

FIGURE 2. Registration phase of P_i and CS.

message was received'' and ΔT represents the ''maximum transmission delay associated with a message''. If it is valid, CS computes $X_i = PK_i * s_{CS}, HID_i = h(X_i) \oplus D_i$, and matches $h(HID_i||s_{CS}) \overset{?}{=} CID_i$ in the database. If this equality holds, P_i is registered.

- **Step 4**: Next, CS computes $L_{i1} = h(X_i\ ||HID_i\ ||T_1\ ||ID_{CS})$ and checks $\check{e}(PID_i, PK_{CS}) \overset{?}{=} \check{e}((HID_i * L_{i1}) * PK_{TA}, PK_{TA})$. If this equality holds, P_i is authenticated. Then, CS generates a random secret $r_{CS} \in Z_q^*$ and current timestamp T_2, and then computes $R_{CS} = r_{CS} * P$, $V_{CS} = r_{CS} * PK_i$, $SK_{i-CS} = h(HID_i\ ||V_{CS}\ ||X_i)$ and $L_{i2} = h(V_{CS}\ ||SK_{i-CS}\ ||ID_{CS}\ ||T_2)$. Thereafter, CS sends the message (R_{CS}, L_{i2}, T_2) to P_i via open channel.
- **Step 5**: After receiving the message (R_{CS}, L_{i2}, T_2), P_i first checks the validity of received timestamp T_2 by the condition: $|T_2 - T_2^*| < \Delta T$, where T_2^* is the ''time when the message was received''. If the timestamp validation passes, P_i computes $V_{CS} = (a_i * r_i) * R_{CS}$ and $SK_{i-CS} = h(HID_i\ ||V_{CS}\ ||X_i)$. After that, P_i checks $L_{i2} \overset{?}{=} h(V_{CS}\ ||SK_{i-CS}\ ||ID_{CS}\ ||T_2)$. If this equality holds, the session key SK_{i-CS} is established between P_i and CS.

E. DATA UPLOAD

After the authentication phase, P_i can upload the health data HD_i to CS.

- **Step 1**: P_i selects access tree Γ. Then, γ is a root of Γ and P_i selects random polynomial $q_\gamma(x)$ with degree $d_\gamma = v_\gamma - 1$. Thereafter, P_i chooses a random number x_i, sets $x_i = q_\gamma(0)$, and chooses d_γ other nodes randomly to complete the polynomial. P_i computes $C_{i1} = HD_i * \check{e}(P, P)^{\alpha x_i}$, and $C_{i2} = PK_{TA} * x_i$. Next, for other nodes y of Γ, P_i sets $q_y(0) = q_{par(y)}(ind(y))$, and chooses d_y other points randomly to complete polynomial $q_y(x)$. After that, P_i computes $C_{il} = P * q_l(0)$, and $C'_{il} = H(att(l)) * q_l(0)$ for all leaf nodes l of Γ. The ciphertext is defined as $CT_i = (\Gamma, C_{i1}, C_{i2}, C_{il}, C'_{il})$, and P_i sends $((CT_i, T_3)_{SK_{i-CS}}, h(ID_{CS}||PK_i||CT_i||T_3))$ to CS.
- **Step 2**: After CS receives the message, CS checks timestamp T_3, decrypts $(CT_i||T_3)$, and verifies $h(ID_{CS}||PK_i||CT_i||T_3) \overset{?}{=} h(ID_{CS}||PK_i||CT_i||T_3)$. If this equality holds, CS stores CT_i in the database and add_i set as a data record address of CT_i. After that, CS uploads $(PK_i, \Gamma, h(CT_i||PK_i), add_i)$ to the blockchain.

F. TREATMENT

D_k can request HD_i from the cloud server through the transaction obtained from the blockchain.

- **Step 1**: If D_k obtains transaction $(P_i, \Gamma, h(CT_i||P_i), add_i)$ and has access to the corresponding data, they can request the data from CS. D_k generates random $r_k \in Z_q^*$,

P_i		CS
Inputs ID_i^* and PW_i^* to SC_i SC_i computes $a_i^* = A_i \oplus h(ID_i^*\|\|PW_i^*)$, $HID_i^* = h(ID_i^*\|\|a_i^*)$, $HPW_i^* = h(ID_i^*\|\|PW_i^*\|\|a_i^*)$, $b_i^* = HPW_i^* \oplus B_i$, $SID_i^* = C_i \oplus b_i^* * P$ Checks $Reg_i \overset{?}{=} h(a_i^*\|\|b_i^*\|\|HPW_i^*\|\|SID_i^*)$ Generates a random number $r_i \in Z_q^*$, current timestamp T_1 Computes $PK_i = (a_i * r_i) * P$, Computes $X_i = (a_i * r_i) * PK_{CS}$, $D_i = HID_i \oplus h(X_i)$, $L_{i1} = h(X_i\|\|HID_i\|\|T_1\|\|ID_{CS})$, $PID_i = SID_i * L_{i1}$		
	$\xrightarrow{\{PK_i, D_i, PID_i, T_1\}}$	
		Checks if $\|T_1 - T_1^*\| < \Delta T$? If valid, computes $X_i = PK_i * s_{CS}$, $HID_i = h(X_i) \oplus D_i$ Matches $h(HID_i\|\|s_{CS}) \overset{?}{=} CID_i$ in the database Computes $L_{i1} = h(X_i\|\|HID_i\|\|T_1\|\|ID_{CS})$ Checks $\check{e}(PID_i, P) \overset{?}{=}$ $\check{e}((HID_i * L_{i1}) * PK_{TA}, PK_{TA})$ Generates a random number $r_{CS} \in Z_q^*$, current timestamp T_2 Computes $R_{CS} = r_{CS} * P$, $V_{CS} = r_{CS} * PK_i$, $SK_{i-CS} = h(HID_i\|\|V_{CS}\|\|X_i)$, $L_{i2} = h(V_{CS}\|\|SK_{i-CS}\|\|ID_{CS}\|\|T_2)$
	$\xleftarrow{\{R_{CS}, L_{i2}, T_2\}}$	
Checks if $\|T_2 - T_2^*\| < \Delta T$? If valid, computes $V_{CS} = (a_i * r_i) * R_{CS}$, $SK_{i-CS} = h(HID_i\|\|V_{CS}\|\|X_i)$ Checks $L_{i2} \overset{?}{=} h(V_{CS}\|\|SK_{i-CS}\|\|ID_{CS}\|\|T_2)$		

FIGURE 3. Authentication phase between P_i and CS.

and then computes $M_1 = (ID_k||add_k||r_k||T_4) + s_k * PK_{CS}$, and $M_2 = h(ID_k||add_i||r_k)$. Thereafter, D_k sends a request message (M_1, M_2, T_4) to CS.

- **Step 2**: After CS receives (M_1, M_2, T_4), CS computes $(ID_k||add_i||r_k||T_4) = M_1 - s_{CS} * PK_k$, and checks $M_2 \overset{?}{=} h(ID_k||add_i||r_k)$. Then, CS retrieves ID_k from the blockchain, and confirms whether ATT_k satisfies access tree of CT_i. If this condition is satisfied, CS computes $M_3 = (CT_i||T_5) + s_{CS} * PK_k$, and sends (M_3, T_5) to D_k.
- **Step 3**: D_k receives the message and computes $(CT_i||T_5) = M_3 - s_k * PK_{CS}$. Then, it checks $h(CT_i||PK_i) \overset{?}{=} h(CT_i||PK_i)$ obtained from the blockchain. If the root node γ is a leaf node, D_k computes $\check{e}(A_{k_s}, C_{i\gamma})$ and $\check{e}(A'_{k_s}, C'_{i\gamma})$. Thereafter, D_k calculates $\frac{\check{e}(A_{k_s}, C_{i\gamma})}{\check{e}(A'_{k_s}, C'_{i\gamma})} = \check{e}(P,P)^{r_k q_\gamma(0)} = K$, and computes $\frac{C_{i1}}{\check{e}(C_{i2}, A_k)/K} = HD_i$.

Correctness:

$$\begin{aligned}
\frac{\check{e}(A_{k_s}, C_{i\gamma})}{\check{e}(A'_{k_s}, C'_{i\gamma})} &= \frac{\check{e}(r_k P + r_{k_s} H(att(\gamma)), q_\gamma(0)P)}{\check{e}(r_{k_s} P, H(att(\gamma)) q_\gamma(0))} \\
&= \frac{\check{e}(r_k P, q_\gamma(0)P)\check{e}(r_{k_s} H(att(\gamma)), q_\gamma(0)P)}{\check{e}(r_{k_s} P, H(att(\gamma)) q_\gamma(0))} \\
&= \frac{\check{e}(P,P)^{r_k q_\gamma(0)}\check{e}(H(att(\gamma)), P)^{r_{k_s} q_\gamma(0)}}{\check{e}(P, H(att(\gamma)))^{r_{k_s} q_\gamma(0)}} \\
&= \check{e}(P,P)^{r_k q_\gamma(0)} = K
\end{aligned}$$

When γ is a threshold gate and childnodes are attributes, we define some notations for convenience of calculation. c_γ is a set of childnodes of γ, and Lagrange coefficient $\Delta_{ind(l),c_\gamma}(x) = \Pi_{j \in c_\gamma, ind(j) \neq ind(l)} \frac{x - ind(j)}{ind(l) - ind(j)}$. First, D_k calculates $\frac{\check{e}(A_{k_s}, C_{il})}{\check{e}(A'_{k_s}, C'_{il})} = \check{e}(P,P)^{r_k q_l(0)} = K_l$ for all leaf nodes l. Next, D_k computes

$$\begin{aligned}\prod_l K_l^{\Delta_{ind(l),c_\gamma}(0)} &= \prod_l (\check{e}(P,P)^{r_k q_l(0)})^{\Delta_{ind(l),c_\gamma}(0)} \\ &= \prod_l (\check{e}(P,P)^{r_k q_\gamma(ind(l))})^{\Delta_{ind(l),c_\gamma}(0)} \\ &= \check{e}(P,P)^{r_k q_\gamma(0)} = K\end{aligned}$$

Then, D_k computes

$$\begin{aligned}\frac{C_{i1}}{\check{e}(C_{i2}, A_k)/K} &= \frac{HD_i * \check{e}(P,P)^{\alpha x_i}}{\check{e}(x_i PK_{TA}, Q(\alpha + r_k))/K} \\ &= \frac{HD_i * \check{e}(P,P)^{\alpha x_i}}{\check{e}(P,P)^{x_i(\alpha+r_k)}/K} = HD_i\end{aligned}$$

- **Step 4**: D_k generates diagnosis Dig_k and computes $M_4 = (ID_k||Dig_k||PK_i) + s_k * PK_i, M_5 = h(ID_k||Dig_k||PK_i), M_6 = (ID_k||R_i||M_4||M_5) + s_k * PK_{CS}$, and $M_7 = h(M_4||M_5||T_6)$. Then, D_k sends (M_6, M_7, T_6) to CS.
- **Step 5**: CS computes $(ID_k||PK_i||M_4||M_5) = M_6 - s_{CS} * PK_i$, and checks $M_7 \overset{?}{=} h(M_4||M_5||T_6)$. If this equality holds, CS stores M_4 in the database, generates data address add_k, and uploads (PK_i, ID_k, M_5, add_k) to the blockchain.

G. CHECKUP

P_i can obtain the diagnosis result Dig_k from the cloud server.

- **Step 1**: P_i obtains (PK_i, ID_k, M_5, add_k) from the blockchain. P_i computes $M_8 = (PK_i||add_k)_{SK_{i-CS}}$, and $M_9 = h(PK_i||add_k||T_7)$. Then, P_i sends (M_8, M_9, T_7) to CS.
- **Step 2**: If CS receives the message, CS decrypts $(PK_i||add_k)$ and checks $M_9 \overset{?}{=} h(PK_i||add_k||T_7)$. If this equality holds, CS computes $M_{10} = (M_4||M_5||T_8)_{SK_{i-CS}}$, and sends (M_{10}, T_8) to P_i
- **Step 3**: Thereafter, P_i decrypts $(M_4||M_5||T_8)$, and computes $(ID_k||Dig_k||PK_i) = M_4 - s_i * PK_k$. Finally, P_i checks $M_5 \overset{?}{=} h(ID_k||Dig_k||PK_i)$.

VI. SECURITY ANALYSIS

In this section, we demonstrate that the proposed protocol defeats a variety of attacks using informal analysis, and we implement formal analysis including the "Burrows–Abadi–Needham (BAN) logic" [54] and "Automated Validation of Internet Security Protocols and Applications (AVISPA) software validation tool" [57], [58].

A. INFORMAL ANALYSIS

We conduct the informal analysis for demonstrating that the proposed protocol prevents from a variety of attacks and supports patient anonymity, untraceability, and mutual authentication.

1) REPLAY AND MITM ATTACKS

The assumed adversary model of the proposed protocol can obtain transmitted messages through a public channel. However, an attacker cannot replay and MITM attacks with these messages because every transmitted message contains timestamp. Each timestamp is generated by the sender and included in the calculation process of hash values $L_{i1} = h(X_i||HID_i||T_1||ID_{CS})$, and $L_{i2} = h(V_{CS}||SK_{i-CS}||ID_{CS}||T_2)$. An attacker cannot forge X_i and HID_i of L_{i1} and V_{CS} of L_{i2}. Therefore, an attacker cannot forge these hash values and the proposed protocol successfully prevents from the replay and MITM attacks.

2) SESSION KEY DISCLOSURE ATTACK

An attacker can attempt to obtain session key SK_{i-CS} directly. It is computed using HID_i, V_{CS}, and X_i. However, an attacker should obtain (a_i, r_i) or r_{CS} to calculate X_i and V_{CS}. Also, HID_i is encrypted with X_i. However, an attacker cannot obtain values a_i, r_i, and r_{cs} using transmitted messages over a public channel. Therefore, the attacker fails to obtain session key SK_{i-CS}.

3) IMPERSONATION ATTACK

An attacker can impersonate legitimate P_i and attempt to send an authentication message. In this attack, an attacker must be able to generate a legitimate authentication message (PK_i, D_i, PID_i, T_1). However, the attacker cannot generate legal PID_i, as PID_i is calculated using secret identity SID_i. Therefore, CS checks $\check{e}(PID_i, PK_{CS}) \overset{?}{=} \check{e}((HID_i * L_{i1}) * PK_{TA}, X_i)$ and if it is not equal, then the attacker is aborted by CS. Therefore, the proposed protocol allows preventing from the impersonation attack.

4) SMART CARD STOLEN ATTACK

If an attacker obtains or steals smart card SC_i of legitimate P_i, then the attacker can extract the stored value (A_i, B_i, C_i, Reg_i) from SC_i using the power analysis attack. However, the attacker cannot obtain any information about P_i such as ID_i and PW_i, and an attacker cannot calculate PID_i to generate a legitimate authentication message. Therefore, the proposed protocol defends from the smart card stolen attack.

5) OFF-LINE GUESSING ATTACK

The assumed adversary model allowed that an adversary can guess any one of the identity ID_i and password PW_i of a patient P_i at the same time. The attacker can also extract the credentials (A_i, B_i, C_i, Reg_i) from the smart card SC_i of the patient P_i and eavesdrop transmitted messages through a public channel, where $HPW_i = h(ID_i||PW_i||a_i)$, $A_i = h(ID_i||PW_i) \oplus a_i$, $B_i = HPW_i \oplus b_i$, $C_i = SID_i \oplus b_i * P$, and $Reg_i = h(a_i||b_i||HPW_i||SID_i)$. However, the adversary cannot calculate $a_i = A_i \oplus h(ID_i||PW_i)$ without knowing both correct guessing of ID_i and PW_i at the same time. Thus, the adversary can not verify either ID_i or PW_i using the extracted HPW_i. Accordingly, the proposed protocol can prevent off-line guessing attack.

6) PERFECT FORWARD SECRECY

Let us suppose that an attacker obtains secret key s_{CS} of the cloud server. Then, the attacker can calculate X_i and HID_i using a transmitted message (PK_i, D_i, PID_i, T_1). However, the attacker still cannot calculate session key $SK_{i-CS} = h(HID_i||V_{CS}||X_i)$, as the attacker cannot calculate V_{CS} without (a_i, r_i) or r_{CS} that are secret or random numbers. Thus, the proposed protocol ensures perfect forward secrecy.

7) PRIVILEGED-INSIDER ATTACK

If an attacker is a privileged insider, then an attacker can obtain HID_i during the patient registration process and s_{CS} during the cloud server registration process. Then, the attacker can calculate $X_i = PK_i * s_{CS}$. However, CS generates a random number r_{CS} in the session, and the attacker cannot calculate the session key SK_{i-CS}, as the attacker cannot calculate $V_{CS} = r_{CS} * PK_i$ without obtaining r_{CS}. Therefore, the proposed protocol can prevent from the privileged-insider attack.

8) STOLEN VERIFIER ATTACK

If an attacker steals a verification table CID_i stored in CS, the attacker can try to guess ID_i of a legitimate patient P_i. For guessing ID_i, the attacker should calculate $h(HID_i||s_{CS})$ and check $CID_i \overset{?}{=} h(HID_i||s_{CS})$. However, the attacker cannot obtain s_{CS}, which is a secret key of CS. Therefore, the attacker cannot obtain real identity of P_i. Therefore, the proposed protocol is secure against the stolen verifier attack.

9) KNOWN SESSION-SPECIFIC TEMPORARY INFORMATION ATTACK

If an attacker can obtain random numbers r_i and r_{CS} generated in the session, then the attacker can compute $V_{CS} = r_{CS} * PK_i$. However, the attacker still cannot calculate $X_i = PK_i * s_{CS}$ without obtaining s_{CS} or a_i. Therefore, the attacker cannot calculate $SK_{i-CS} = h(HID_i||V_{CS}||X_i)$, and the proposed protocol can prevent from the known session-specific temporary information attack.

10) PATIENT ANONYMITY

Patient anonymity is guaranteed in the proposed protocol. P_i sends (PK_i, D_i, PID_i, T_i) in the authentication phase. However, an attacker cannot obtain the real identity ID_i of P_i from this message, as it is dependent on random number r_i. Therefore, the attacker is not able to obtain the real identity ID_i of P_i and the proposed protocol ensures patient anonymity.

11) PATIENT UNTRACEABILITY

To provide patient untraceability, an attacker must not be able to trace a patient through transmitted messages. In the proposed protocol, authentication request message (PK_i, D_i, PID_i, T_1) is dependent on random number r_i. Authentication request messages differ in every session so the attacker cannot trace a patient through the messages of past sessions. Therefore, the proposed protocol ensures patient untraceability.

12) MUTUAL AUTHENTICATION

According to 1), an attacker cannot generate a legal PID_i. Therefore, CS can authenticate P_i through calcuating $\check{e}(PID_i, P) \overset{?}{=} \check{e}((HID_i * L_{i1}) * PK_{TA}, PK_{TA})$. Furthermore, the attacker cannot generate a legal L_{i2} so that P_i can authenticate CS by checking $L_{i2} \overset{?}{=} h(V_{CS}||SK_{i-CS}||ID_{CS}||T_2)$. Therefore, mutual authentication is enabled in the proposed protocol.

13) DATA VERIFIABILITY

After data are uploaded in the cloud server, the hash of the data is recorded on the blockchain as a transaction, and patients and doctors can obtain the hash of the data from the blockchain. If an attacker succeeds to modify or forge the health data stored in the cloud server, patients and doctors can verify whether the data are corrupted using the hash of the data. Therefore, the proposed protocol enables data verifiability.

14) ACCESS CONTROL

The proposed protocol can provide fine-grained access control of a patient's health data. P_i sets access tree for their health data and encrypts the data with the access tree, and then uploads the encrypted data to the CS. Then, only doctors who have a proper attribute set which satisfies the access tree of the health data can request the data to CS and decrypt it with their attribute keys. Therefore, the proposed scheme can provide fine-grained access control of the patient's health data.

B. BAN LOGIC ANALYSIS

The BAN logic analysis [53]–[55] can prove secure mutual authentication of a communication protocol. We conduct the BAN logic analysis of the proposed protocol in this section. Table 3 describes the notations and the following statements represent the basic rules of the BAN logic.

TABLE 2. BAN logic notations.

Notation	Description
ρ_1, ρ_2	Two principals
μ_1, μ_2	Two statements
SK	The session key
$\rho_1 \mid\equiv \mu_1$	ρ_1 **believes** μ_1
$\rho_1 \mid\sim \mu_1$	ρ_1 once **said** μ_1
$\rho_1 \Rightarrow \mu_1$	ρ_1 **controls** μ_1
$\rho_1 \lhd \mu_1$	ρ_1 **receives** μ_1
$\#\mu_1$	μ_1 is **fresh**
$(\mu_1)_K$	μ_1 is **encrypted** with K
$\rho_1 \overset{K}{\leftrightarrow} \rho_2$	ρ_1 and ρ_2 have **shared** key K

1. Message meaning rule (MMR) :

$$\frac{\rho_1 \mid\equiv \rho_1 \overset{K}{\leftrightarrow} \rho_2, \quad \rho_1 \lhd (\mu_1)_K}{\rho_1 \mid\equiv \rho_2 \mid\sim \mu_1}$$

2. Nonce verification rule (NVR) :

$$\frac{\rho_1| \equiv \#(\mu_1), \quad \rho_1| \equiv \rho_2 \,\Big|\! \sim \mu_1}{\rho_1| \equiv \rho_2| \equiv \mu_1}$$

3. Jurisdiction rule (JR) :

$$\frac{\rho_1| \equiv \rho_2| \Longrightarrow \mu_1, \quad \rho_1| \equiv \rho_2| \equiv \mu_1}{\rho_1 \,\Big|\! \equiv \mu_1}$$

4. Belief rule (BR) :

$$\frac{\rho_1 \,\Big|\! \equiv (\mu_1, \mu_2)}{\rho_1 \,\Big|\! \equiv \mu_1}$$

5. Freshness rule (FR) :

$$\frac{\rho_1 \,\Big|\! \equiv \#(\mu_1)}{\rho_1 \,\Big|\! \equiv \#(\mu_1, \mu_2)}$$

1) GOALS

The goals for proving mutual authentication of the proposed protocol are defined as follows:

Goal 1: $P_i| \equiv P_i \xleftrightarrow{SK_{i-CS}} CS$

Goal 2: $P_i| \equiv CS| \equiv P_i \xleftrightarrow{SK_{i-CS}} CS$

Goal 3: $CS| \equiv P_i \xleftrightarrow{SK_{i-CS}} CS$

Goal 4: $CS| \equiv P_i| \equiv P_i \xleftrightarrow{SK_{i-CS}} CS$

2) IDEALIZED FORMS

The idealized forms based on the BAN logic of the proposed protocol are as below:

$Msg_1 \ : \ P_i \rightarrow CS : (PK_i, HID_i, T_1)_{X_i}$
$Msg_2 \ : \ CS \rightarrow P_i : (V_{CS}, HID_i, T_2)_{X_i}$

3) ASSUMPTIONS

The assumptions of the BAN logic are as below:

A_1: $CS| \equiv \#(T_1)$
A_2: $P_i| \equiv \#(T_2)$
A_3: $P_i| \equiv CS \Rightarrow (P_i \xleftrightarrow{SK_{i-CS}} CS)$
A_4: $CS| \equiv P_i \Rightarrow (P_i \xleftrightarrow{SK_{i-CS}} CS)$
A_5: $P_i| \equiv P_i \overset{X_i}{\leftrightarrow} CS$
A_6: $CS| \equiv P_i \overset{X_i}{\leftrightarrow} CS$
A_7: $P_i| \equiv (HID_i)$
A_8: $P_i| \equiv CS| \equiv (HID_i)$

4) BAN LOGIC PROOF

We implement the BAN logic analysis of the proposed protocol as below:

Step 1: S_1 is obtained from Msg_1.

$$S_1 : CS \lhd (PK_i, HID_i, T_1)_{X_i}$$

Step 2: S_2 is obtained by applying the MMR using S_1 and A_6.

$$S_2 : CS| \equiv P_i| \sim (PK_i, HID_i, T_1)_{X_i}$$

Step 3: S_3 is obtained by applying the FR using S_2 and A_1.

$$S_3 : CS| \equiv \#(PK_i, HID_i, T_1)_{X_i}$$

Step 4: S_4 is obtained by applying the NVR using S_2 and S_3.

$$S_4 : CS| \equiv P_i| \equiv (PK_i, HID_i, T_1)_{X_i}$$

Step 5: S_5 is obtained from S_4 and the BR.

$$S_5 : CS| \equiv P_i| \equiv (PK_i, HID_i)$$

Step 6: S_6 is obtained from $V_{CS} = r_{cs} * PK_i$, and the session key $SK_{i-CS} = h(HID_i||V_{CS}||X_i)$.

$$S_6 : CS| \equiv P_i| \equiv (P_i \xleftrightarrow{SK_{i-CS}} CS) \quad \textbf{(Goal 4)}$$

Step 7: S_7 is obtained by applying the JR using A_4 and S_6.

$$S_7 : CS| \equiv (P_i \xleftrightarrow{SK_{i-CS}} CS) \quad \textbf{(Goal 3)}$$

Step 8: S_8 is obtained from Msg_2.

$$S_8 : P_i \lhd (V_{CS}, HID_i, T_2)_{X_i}$$

Step 9: S_9 is obtained by applying the MMR using A_5 and S_8.

$$S_9 : P_i| \equiv CS| \sim (V_{CS}, HID_i, T_2)_{X_i}$$

Step 10: S_{10} is obtained by applying the FR using A_2 and S_9.

$$S_{10} : P_i| \equiv \#(V_{CS}, HID_i, T_2)_{X_i}$$

Step 11: S_{11} is obtained by applying the NVR using S_9 and S_{10}.

$$S_{11} : P_i| \equiv CS| \equiv (V_{CS}, HID_i, T_2)_{X_i}$$

Step 12: S_{12} is obtained by applying the BR using A_8 and S_{11}.

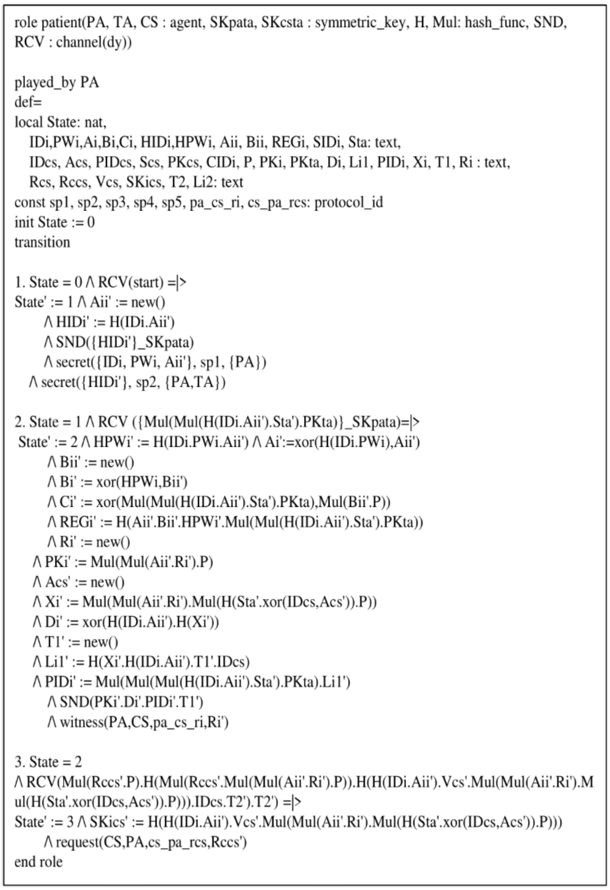
$$S_{12} : P_i| \equiv CS| \equiv (V_{CS})$$

Step 13: S_{13} is obtained from A_7, S_{12}, and the session key $SK_{i-CS} = h(HID_i||V_{CS}||X_i)$.

$$S_{13} : P_i| \equiv CS| \equiv (P_i \xleftrightarrow{SK_{i-CS}} CS) \quad \textbf{(Goal 2)}$$

Step 14: S_{14} is obtained by applying the JR using A_3 and S_{11}.

$$S_{14} : P_i| \equiv (P_i \xleftrightarrow{SK_{i-CS}} CS) \quad \textbf{(Goal 1)}$$



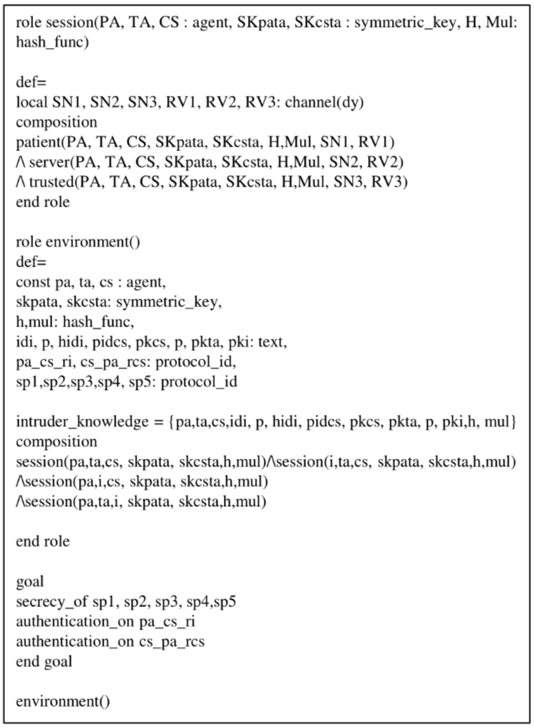
```
role patient(PA, TA, CS : agent, SKpata, SKcsta : symmetric_key, H, Mul: hash_func, SND,
RCV : channel(dy))

played_by PA
def=
local State: nat,
   IDi,PWi,Ai,Bi,Ci, HIDi,HPWi, Aii, Bii, REGi, SIDi, Sta: text,
   IDcs, Acs, PIDcs, Scs, PKcs, CIDi, P, PKi, PKta, Di, Li1, PIDi, Xi, T1, Ri : text,
   Rcs, Rccs, Vcs, SKics, T2, Li2: text
const sp1, sp2, sp3, sp4, sp5, pa_cs_ri, cs_pa_rcs: protocol_id
init State := 0
transition

1. State = 0 ∧ RCV(start) =|>
State' := 1 ∧ Aii' := new()
      ∧ HIDi' := H(IDi.Aii')
      ∧ SND({HIDi'}_SKpata)
      ∧ secret({IDi, PWi, Aii'}, sp1, {PA})
   ∧ secret({HIDi'}, sp2, {PA,TA})

2. State = 1 ∧ RCV ({Mul(Mul(H(IDi.Aii').Sta').PKta)}_SKpata)=|>
 State' := 2 ∧ HPWi' := H(IDi.PWi.Aii') ∧ Ai':=xor(H(IDi.PWi),Aii')
       ∧ Bii' := new()
       ∧ Bi' := xor(HPWi,Bii')
       ∧ Ci' := xor(Mul(Mul(H(IDi.Aii').Sta').PKta),Mul(Bii'.P))
       ∧ REGi' := H(Aii'.Bii'.HPWi'.Mul(Mul(H(IDi.Aii').Sta').PKta))
       ∧ Ri' := new()
    ∧ PKi' := Mul(Mul(Aii'.Ri').P)
    ∧ Acs' := new()
    ∧ Xi' := Mul(Mul(Aii'.Ri').Mul(H(Sta'.xor(IDcs,Acs')).P))
    ∧ Di' := xor(H(IDi.Aii').H(Xi'))
    ∧ T1' := new()
    ∧ Li1' := H(Xi'.H(IDi.Aii').T1'.IDcs)
    ∧ PIDi' := Mul(Mul(Mul(H(IDi.Aii').Sta').PKta).Li1')
       ∧ SND(PKi'.Di'.PIDi'.T1')
       ∧ witness(PA,CS,pa_cs_ri,Ri')

3. State = 2
∧ RCV(Mul(Rccs'.P).H(Mul(Rccs'.Mul(Mul(Aii'.Ri').P)).H(H(IDi.Aii').Vcs'.Mul(Mul(Aii'.Ri').M
ul(H(Sta'.xor(IDcs,Acs')).P))).IDcs.T2').T2') =|>
State' := 3 ∧ SKics' := H(H(IDi.Aii').Vcs'.Mul(Mul(Aii'.Ri').Mul(H(Sta'.xor(IDcs,Acs')).P)))
      ∧ request(CS,PA,cs_pa_rcs,Rccs')
end role
```

FIGURE 4. ***Role*** **of** P_i

C. AVISPA SIMULATION

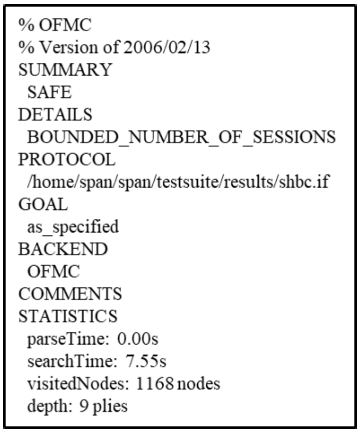
The broadly-accepted "Automated Validation of Internet Security Protocols and Applications (AVISPA)" simulation tool [57], [58] can verify that an authentication protocol is secure against replay and MITM attacks.

In this section, we prove the security against replay and MITM attacks using the AVISPA simulation tool. The AVISPA tool implements communication using the High-Level Protocol Specification Language (HLPSL) [59]. HLPSL takes as input one of four back-end models, namely, "On-the-Fly Model Checker (OFMC)" [60], "Tree Automata based on Automatic Approximations for Analysis of Security Protocol (TA4SP)", "Constraint Logic-based Attack Searcher (CL-AtSe)" [61], and "SAT-based Model Checker (SATMC)". This input is converted to "Intermediate Format (IF)" then output is "Output Format (OF)". In general, the AVISPA tool uses two models OFMC and CL-AtSe for formal verification. If OF is SAFE for OFMC and CL-AtSe models, we can say that the protocol has security against replay and MITM attacks. We provide the implementation details of P_i in Figure 4. The implementation details of *TA* and *CS* is similar to P_i. And the Figure 5 presents the role of goals, and environment. The simulation summary is represented in Figure 6. Under CL-AtSe, the translation time is 0.08 seconds and summary is SAFE and it takes 7.55 seconds as a search time for visiting 1168 nodes with depth 9 piles in OFMC model. The summaries indicate that the proposed protocol is safe. Thus, we can say that the proposed protocol ensures the security against replay and MITM attacks.



```
role session(PA, TA, CS : agent, SKpata, SKcsta : symmetric_key, H, Mul:
hash_func)

def=
local SN1, SN2, SN3, RV1, RV2, RV3: channel(dy)
composition
patient(PA, TA, CS, SKpata, SKcsta, H,Mul, SN1, RV1)
∧ server(PA, TA, CS, SKpata, SKcsta, H,Mul, SN2, RV2)
∧ trusted(PA, TA, CS, SKpata, SKcsta, H,Mul, SN3, RV3)
end role

role environment()
def=
const pa, ta, cs : agent,
skpata, skcsta: symmetric_key,
h,mul: hash_func,
idi, p, hidi, pidcs, pkcs, p, pkta, pki: text,
pa_cs_ri, cs_pa_rcs: protocol_id,
sp1,sp2,sp3,sp4, sp5: protocol_id

intruder_knowledge = {pa,ta,cs,idi, p, hidi, pidcs, pkcs, pkta, p, pki,h, mul}
composition
session(pa,ta,cs, skpata, skcsta,h,mul)∧session(i,ta,cs, skpata, skcsta,h,mul)
∧session(pa,i,cs, skpata, skcsta,h,mul)
∧session(pa,ta,i, skpata, skcsta,h,mul)

end role

goal
secrecy_of sp1, sp2, sp3, sp4,sp5
authentication_on pa_cs_ri
authentication_on cs_pa_rcs
end goal

environment()
```

FIGURE 5. ***Role*** **of goals, and environment.**



```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/shbc.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 7.55s
  visitedNodes: 1168 nodes
  depth: 9 plies
```

FIGURE 6. **Simulation summary.**

VII. PERFORMANCE ANALYSIS

In this section, we provide the result of comparing the computation and communication costs of the proposed protocol and compared security features with the related protocols [17], [18], [20]–[22].

A. COMPUTATION COST

According to the experiments performed in [56], the computation cost of each operation is obtained on a computer with

Intel Pentium Dual CPU E2200 2.20GHz processor, 2 GB RAM and the Ubuntu 12.04.1 LTS 32 bit operation system. The time complexity of each operation is as follows:

- T_{bp}: time complexity of the bilinear pairing operation $\approx$ 5.811 ms
- T_{hp}: time complexity of the map-to-point hash operation $\approx$ 12.418 ms
- T_{exp}: time complexity of the modular exponentiation operation $\approx$ 3.85 ms
- T_{mul}: time complexity of the scalar multiplication operation $\approx$ 2.226 ms
- T_{rng}: time complexity of the random number generation $\approx$ 0.539 ms
- T_{ed}: time complexity of the symmetric encryption/decryption $\approx$ 0.0046 ms
- T_h: time complexity of the one-way hash operation $\approx$ 0.0023 ms

TABLE 3. Computation cost comparison.

Scheme	Computation cost
Wang and Zhang [17]	$2T_{bp} + 4T_{hp} + 5T_{mul} + 2T_{rng} + 6T_h$ ≈ 73.5158 ms
Jiang *et al.* [18]	$2T_{bp} + 4T_{hp} + 6T_{mul} + 2T_{rng} + 2T_{ed} + 4T_h$ ≈ 75.7464 ms
Liu *et al.* [20]	$2T_{hp} + 16T_{mul} + 2T_{rng} + 10T_h$ ≈ 61.553 ms
Chen and Peng [21]	$T_{bp} + T_{exp} + 9T_{mul} + 2T_{rng} + 2T_{ed} + 8T_h$ ≈ 30.8006 ms
Khatoon *et al.* [22]	$2T_{bp} + 4T_{hp} + 7T_{mul} + 2T_{rng} + 2T_{ed} + 4T_h$ ≈ 77.9724 ms
Proposed	$2T_{bp} + 13T_{mul} + 2T_{rng} + 9T_h \approx 41.6587$ ms

We do not consider the computation cost of an exclusive OR operation because it is negligible. The total computation cost of the scheme proposed in [17] is $2T_{bp} + 4T_{hp} + 5T_{mul} + 2T_{rng} + 6T_h \approx 73.5158$ ms. And, the total computation cost of the scheme proposed in [18] is $2T_{bp} + 4T_{hp} + 6T_{mul} + 2T_{rng} + 2T_{ed} + 4T_h \approx 75.7464$ ms. The total computation cost of the scheme proposed in [20] is $2T_{hp} + 16T_{mul} + 2T_{rng} + 10T_h \approx 61.553$ ms. The scheme proposed in [21] has $T_{bp} + T_{exp} + 9T_{mul} + 2T_{rng} + 2T_{ed} + 8T_h \approx 30.8006$ ms as the total computation cost. Also, the total computation cost of the scheme proposed in [22] is $2T_{bp} + 4T_{hp} + 7T_{mul} + 2T_{rng} + 2T_{ed} + 4T_h \approx 77.9724$ ms. The proposed protocol incurs $2T_{bp} + 13T_{mul} + 2T_{rng} + 9T_h \approx 41.6587$ ms as the computation cost. The summary is represented in Table 3. As represented in Table 3, the proposed protocol has slightly higher computation cost compared to that in the scheme [21]. However, the proposed protocol has lower communication cost and provides superior security as compare to those for other existing competing schemes.

B. COMMUNICATION COST

We compare the total communication cost of the proposed protocol and the related protocols [17], [18], [20]–[22]. According to the scheme in [56], we also define the bit sizes of a one-way cryptographic hash output (message digest) and the group element of G_1 as 160 bits and 1024 bits, respectively. Furthermore, according to the scheme [36], we define bit sizes of the identity and timestamp as 128 bits and 32 bits, respectively.

TABLE 4. Communication cost comparsion.

Scheme	Communication cost
Wang and Zhang [17]	2432 bits
Jiang *et al.* [18]	2592 bits
Liu *et al.* [20]	4704 bits
Chen and Peng [21]	4608 bits
Khatoon *et al.* [22]	2592 bits
Proposed	3456 bits

In the scheme proposed in [17], the message $M_1 = (R_C, T_C, Auth_C)$ requires $1024 + 32 + 160 = 1216$ bits, and the message $M_2 = (R_{AP}, T_{AP}, Auth_{AP})$ needs $1024 + 32 + 160 = 1216$ bits. Therefore, the communication cost required in the scheme [17] is $1216 + 1216 = 2432$ bits. In the scheme [18], the message $M_1 = (R_C, T_C, Auth_C)$ with $Auth_C = E_{K_C}(ID_C||T_C||r_C)$ needs $1024 + 32 + 320 = 1376$ bits, whereas the message $M_2 = (R_{AP}, T_{AP}, Auth_{AP})$ demands $1024 + 32 + 160 = 1216$ bits. The total communication cost of the scheme proposed in [18] is then $1376 + 1216 = 2592$ bits. In the scheme proposed in [20], the message $M_1 = (ID_C, PK_C, P_C)$ requires $128 + 1024 + 1024 = 2176$ bits, the message $M_2 = (ID_A, PK_A, P_A)$ needs 2176 bits, the message $M_3 = (MAC_C, T)$ needs $160 + 32 = 192$ bits, and the final message $M_4 = (MAC_A)$ demands 160 bits. Accordingly, the total communication cost of the scheme proposed in [20] is $2176 + 2176 + 192 + 160 = 4704$ bits. In the scheme proposed in [21], the message $M_1 = (V_C, Auth_c, T_C)$ is $1024+2368+32 = 3424$ bits, whereas the message $M_2 = (R_{AP}, Auth_{AP})$ requires $1024 + 160 = 1184$ bits. Therefore, the total communication cost of the scheme proposed in [21] is $3424 + 1184 = 4608$ bits. In the scheme proposed in [22], the message $LR_i = (R_i, T_i, Auth_i)$ with $Auth_i = E_{k_i}(ID_i||T_i||r_i)$ needs $1024 + 32 + 320 = 1376$ bits, whereas the message $MA = (R_s, T_s, Auth_s)$ demands $1024 + 32 + 160 = 1216$ bits. Therefore, the total communication cost of the scheme in [22] is $1376 + 1216 = 2592$ bits. In the proposed protocol, the authentication request message (PK_i, D_i, PID_i, T_1) needs $1024 + 160 + 1024 + 32 = 2240$ bits, and the response message (R_{CS}, L_{i2}, T_2) requires $1024 + 160 + 32 = 1216$ bits. Therefore, the proposed protocol incurs $2240 + 1216 = 3456$ bits as the communication cost. Table 4 represents a comparative study on communication costs among the proposed protocol and other competing schemes [17], [18], [20]–[22]. As represented in Table 4, the proposed protocol has low communication cost as compared to the schemes [20], [21]. Though compared to the schemes [17], [18], [22], the proposed protocol has slightly higher communication cost, but the proposed protocol has significantly lower computation cost and provides more security features as compared to these schemes.

C. SECURITY FEATURES

Table 5 represents the comparison of security features with the related protocols proposed by Wang and Zhang [17], Jiang *et al.* [18], Liu *et al.* [20], Chen and Peng [21], and

TABLE 5. Comparison of security features.

Security features	[17]	[18]	[20]	[21]	[22]	Proposed
Replay and MITM attacks	O	O	O	O	O	O
Session key disclosure attack	O	O	O	O	O	O
Off-line guessing attack	O	O	–	–	O	O
Impersonation attack	X	O	O	O	O	O
Perfect forward secrecy	O	O	O	O	X	O
Privileged-insider attack	–	–	–	–	–	O
Stolen verifier attack	O	O	O	–	–	O
Known session-specific temporary information	–	–	–	–	X	O
Patient anonymity	O	O	X	O	O	O
Patient unlinkability	O	O	X	O	O	O
Mutual authentication	X	O	O	O	O	O
Decentralized	X	X	X	X	X	O
Verifiability	X	X	X	X	X	O
Access control	X	X	X	X	X	O

X : Insecure. O : Secure. – : Not considered.

Khatoon *et al.* [22]. We have considered several security and functionality features, such as a) ''resistant to replay and MITM attacks'', b) ''resistant to session key disclosure attack'', c) ''resistant to off-line guessing attack'', d) ''resistant to impersonation attack'', e) ''preservation of perfect forward secrecy'', f) ''resistant to privileged-insider attack'', g) ''resistant to stolen verifier attack'', h) ''resistant to known session-specific temporary information attack'', i) ''preservation of patient anonymity'', j) ''preservation of patient unlinkability'', k) ''support to mutual authentication'', l) ''support to decentralization'', m) ''verifiability'', and n) ''support to access control''. From Table 5, it is clear to observe that the proposed scheme provides superior security and more functionality features as compared to those for other existing schemes [17], [18], [20]–[22].

VIII. CONCLUSION

We proposed a secure protocol for a cloud-assisted TMIS with access control using blockchain. The proposed model utilized the blockchain technology to guarantee data integrity in the cloud server and applied consortium blockchain for scalability and low computation cost. Moreover, we employed CP-ABE for access control of stored data in the cloud so that the proposed model achieved fine-grained access control. Furthermore, the proposed protocol included registration, authentication, data upload, treatment, and checkup. We conducted informal analysis to show that the proposed protocol prevents from a variety of attacks and we compared the security features of the proposed protocol with the related protocols. We also utilized the BAN logic analysis for proving that it supports secure mutual authentication, and AVISPA to show that it is safe for MITM and replay attacks. Furthermore, we compared computation blue and communication costs of the proposed protocol with the related protocols. We demonstrated that the proposed protocol is efficient and has better safety compared to the related protocols. Thus, the proposed protocol is proper for a practical TMIS environment. In the future work, our goal is to simulate a whole network and secure protocol to design a new scheme being more pracitcal in TMIS.

REFERENCES

[1] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, ''Wireless body area networks: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1658–1686, 3rd Quart., 2014.

[2] Y. Park and Y. Park, ''Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks,'' *Sensors*, vol. 16, no. 12, p. 2123, Dec. 2016.

[3] Y. Park, K. Park, and Y. Park, ''Secure user authentication scheme with novel server mutual verification for multiserver environments,'' *Int. J. Commun. Syst.*, vol. 32, no. 7, p. e3929, May 2019.

[4] S. Yu, J. Lee, K. Lee, K. Park, and Y. Park, ''Secure authentication protocol for wireless sensor networks in vehicular communications,'' *Sensors*, vol. 18, no. 10, p. 3191, Sep. 2018.

[5] G. Sharma and S. Kalra, ''A lightweight user authentication scheme for cloud-IoT based healthcare services,'' *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 43, no. S1, pp. 619–636, Jul. 2019.

[6] D. He, S. Zeadally, and L. Wu, ''Certificateless public auditing scheme for cloud-assisted wireless body area networks,'' *IEEE Syst. J.*, vol. 12, no. 1, pp. 64–73, Mar. 2018.

[7] A. Sahai and B. Waters, ''Fuzzy identity-based encryption,'' in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*. Berlin, Germany: Springer, 2005, pp. 457–473.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, ''Attribute-based encryption for fine-grained access control of encrypted data,'' in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, Alexandria, VA, USA, Nov. 2006, pp. 89–98.

[9] J. Bethencourt, A. Sahai, and B. Waters, ''Ciphertext-policy attributebased encryption,'' in *Proc. IEEE Symp. Secur. Privacy (SP)*, Berkeley, CA, USA, May 2007, pp. 321–334.

[10] S. Xie, Z. Zheng, W. Chen, J. Wu, H.-N. Dai, and M. Imran, ''Blockchain for cloud exchange: A survey,'' *Comput. Electr. Eng.*, vol. 81, Jan. 2020, Art. no. 106526.

[11] S. Nakamoto. (2008).*Bitcoin: A Peer-To-Peer Electronic Cash System.* Accessed: Jul. 2020. [Online]. Available: http://bitcoin.org/bitcoin.pdf

[12] A. Zhang and X. Lin, ''Towards secure and privacy-preserving data sharing in e-Health systems via consortium blockchain,'' *J. Med. Syst.*, vol. 42, no. 8, p. 140, Aug. 2018.

[13] H. Wang and Y. Song, ''Secure cloud-based EHR system using attribute-based cryptosystem and blockchain,'' *J. Med. Syst.*, vol. 42, no. 8, p. 152, Jul. 2018.

[14] Y. Wang, A. Zhang, P. Zhang, and H. Wang, ''Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain,'' *IEEE Access*, vol. 7, pp. 136704–136719, 2019.

[15] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, ''Certificateless remote anonymous authentication schemes for wireless body area networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 332–342, Feb. 2014.

[16] Z. Zhao, ''An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem,'' *J. Med. Syst.*, vol. 38, no. 2, p. 1, Jan. 2014.

[17] C. Wang and Y. Zhang, ''New authentication scheme for wireless body area networks using the bilinear pairing,'' *J. Med. Syst.*, vol. 39, no. 11, p. 136, Sep. 2015.

[18] Q. Jiang, X. Lian, C. Yang, J. Ma, Y. Tian, and Y. Yang, "A bilinear pairing based anonymous authentication scheme in wireless body area networks for mHealth," *J. Med. Syst.*, vol. 40, no. 11, p. 1, Nov. 2016.

[19] G. Mwitende, Y. Ye, I. Ali, and F. Li, "Certificateless authenticated key agreement for blockchain-based WBANs," *J. Syst. Archit.*, vol. 110, pp. 101777–101789, Apr. 2020.

[20] X. Liu, C. Jin, and F. Li, "An improved two-layer authentication scheme for wireless body area networks," *J. Med. Syst.*, vol. 42, no. 8, p. 143, Jun. 2018.

[21] R. Chen and D. Peng, "Analysis and improvement of a mutual authentication scheme for wireless body area networks," *J. Med. Syst.*, vol. 43, no. 2, p. 19, Dec. 2018.

[22] S. Khatoon, S. M. M. Rahman, M. Alrubaian, and A. Alamri, "Privacy-preserved, provable secure, mutually authenticated key agreement protocol for healthcare in a smart city environment," *IEEE Access*, vol. 7, pp. 47962–47971, 2019.

[23] M. Nikooghadam and H. Amintoosi, "Cryptanalysis of khatoon et al.'s ECC-based authentication protocol for healthcare systems," 2019, *arXiv:1906.08424*. [Online]. Available: http://arxiv.org/abs/1906.08424

[24] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, A. G. Reddy, K. Park, and Y. Park, "On the design of fine grained access control with user authentication scheme for telecare medicine information systems," *IEEE Access*, vol. 5, pp. 7012–7030, 2017.

[25] C.-L. Chen, T.-T. Yang, M.-L. Chiang, and T.-F. Shih, "A privacy authentication scheme based on cloud for medical environment," *J. Med. Syst.*, vol. 38, no. 11, p. 143, Nov. 2014.

[26] S.-Y. Chiou, Z. Ying, and J. Liu, "Improvement of a privacy authentication scheme based on cloud for medical environment," *J. Med. Syst.*, vol. 40, no. 4, p. 101, Apr. 2016.

[27] P. Mohit, R. Amin, A. Karati, G. P. Biswas, and M. K. Khan, "A standard mutual authentication protocol for cloud computing based health care system," *J. Med. Syst.*, vol. 41, no. 4, p. 50, Apr. 2017.

[28] C.-T. Li, D.-H. Shih, and C.-C. Wang, "Cloud-assisted mutual authentication and privacy preservation protocol for telecare medical information systems," *Comput. Methods Programs Biomed.*, vol. 157, pp. 191–203, Apr. 2018.

[29] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.

[30] R. Guo, H. Shi, D. Zheng, C. Jing, C. Zhuang, and Z. Wang, "Flexible and efficient blockchain-based ABE scheme with multi-authority for medical on demand in telemedicine system," *IEEE Access*, vol. 7, pp. 88012–88025, 2019.

[31] X. Yang, T. Li, X. Pei, L. Wen, and C. Wang, "Medical data sharing scheme based on attribute cryptosystem and blockchain technology," *IEEE Access*, vol. 8, pp. 45468–45476, 2020.

[32] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM J. Comput.*, vol. 32, no. 3, pp. 586–615, Jan. 2003.

[33] S. Ding, J. Cao, C. Li, K. Fan, and H. Li, "A novel attribute-based access control scheme using blockchain for IoT," *IEEE Access*, vol. 7, pp. 38431–38441, 2019.

[34] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Honolulu, HI, USA, Jun. 2017, pp. 557–564.

[35] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[36] M. Kim, S. Yu, J. Lee, Y. Park, and Y. Park, "Design of secure protocol for cloud-assisted electronic health record system using blockchain," *Sensors*, vol. 20, no. 10, p. 2913, May 2020.

[37] J. Lee, S. Yu, M. Kim, Y. Park, and A. K. Das, "On the design of secure and efficient three-factor authentication protocol using honey list for wireless sensor networks," *IEEE Access*, vol. 8, pp. 107046–107062, 2020.

[38] K. Park, Y. Park, A. K. Das, S. Yu, J. Lee, and Y. Park, "A dynamic privacy-preserving key management protocol for V2G in social Internet of Things," *IEEE Access*, vol. 7, pp. 76812–76832, 2019.

[39] S. Yu, J. Lee, Y. Park, Y. Park, S. Lee, and B. Chung, "A secure and efficient three-factor authentication protocol in global mobility networks," *Appl. Sci.*, vol. 10, no. 10, p. 3565, May 2020.

[40] S. Yu, K. Park, J. Lee, Y. Park, Y. Park, S. Lee, and B. Chung, "Privacy-preserving lightweight authentication protocol for demand response management in smart grid environment," *Appl. Sci.*, vol. 10, no. 5, p. 1758, Mar. 2020.

[41] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. and Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain," in *Proc. Italian Conf. Cyber Secur.*, Milan, Italy, Jun. 2018, pp. 1–11.

[42] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 391–406, Mar. 2020.

[43] S. Challa, M. Wazid, A. K. Das, N. Kumar, A. Goutham Reddy, E.-J. Yoon, and K.-Y. Yoo, "Secure signature-based authenticated key establishment scheme for future IoT applications," *IEEE Access*, vol. 5, pp. 3028–3043, 2017.

[44] C.-C. Chang and H.-D. Le, "A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 357–366, Jan. 2016.

[45] A. K. Das, M. Wazid, N. Kumar, A. V. Vasilakos, and J. J. P. C. Rodrigues, "Biometrics-based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4900–4913, Dec. 2018.

[46] S. Malani, J. Srinivas, A. K. Das, K. Srinathan, and M. Jo, "Certificate-based anonymous device access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9762–9773, Dec. 2019.

[47] M. Wazid, A. K. Das, V. Odelu, N. Kumar, M. Conti, and M. Jo, "Design of secure user authenticated key management protocol for generic IoT networks," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 269–282, Feb. 2018.

[48] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. P. C. Rodrigues, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 457–468, Jan. 2019.

[49] S. Banerjee, V. Odelu, A. K. Das, J. Srinivas, N. Kumar, S. Chattopadhyay, and K.-K.-R. Choo, "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8739–8752, Oct. 2019.

[50] J. Srinivas, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "TCALAS: Temporal credential-based anonymous lightweight authentication scheme for Internet of drones environment," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6903–6916, Jul. 2019.

[51] M. Wazid, A. K. Das, N. Kumar, and A. V. Vasilakos, "Design of secure key management and user authentication scheme for fog computing services," *Future Gener. Comput. Syst.*, vol. 91, pp. 475–492, Feb. 2019.

[52] S. Chatterjee, A. K. Das, and J. K. Sing, "An enhanced access control scheme in wireless sensor networks," *Ad Hoc Sensor Wireless Netw.*, vol. 21, nos. 1–2, pp. 121–149, Jan. 2014.

[53] J. Lee, S. Yu, K. Park, Y. Park, and Y. Park, "Secure three-factor authentication protocol for multi-gateway IoT environments," *Sensors*, vol. 19, no. 10, p. 2358, May 2019.

[54] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Trans. Comput. Syst.*, vol. 8, no. 1, pp. 18–36, 1990.

[55] K. Park, Y. Park, Y. Park, A. Goutham Reddy, and A. K. Das, "Provably secure and efficient authentication protocol for roaming service in global mobility networks," *IEEE Access*, vol. 5, pp. 25110–25125, 2017.

[56] H. H. Kilinc and T. Yanik, "A survey of SIP authentication and key agreement schemes," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1005–1023, 2nd Quart., 2014.

[57] AVISPA. (2020). *Automated Validation of Internet Security Protocols and Applications*. Accessed: Jul. 2020. [Online]. Available: http://www.avispa-project.org/

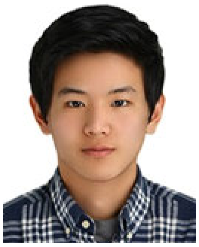
[58] AVISPA. *SPAN, A Security Protocol ANimator for AVISPA*. Accessed: Jul. 2020. [Online]. Available: http://www.avispa-project.org/

[59] D. Von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. 3rd APPSEM II Workshop Appl. Semantics (APPSEM)*, Frauenchiemsee, Germany, 2005, pp. 1–17.

[60] D. Basin, S. Mödersheim, and L. Vigano, "OFMC: A symbolic model checker for security protocols," *Int. J. Inf. Secur.*, vol. 4, no. 3, pp. 181–208, Jun. 2005.

[61] M. Turuani, "The CL-Atse protocol analyser," in *Proc. Int. Conf. Rewriting Techn. Appl.*, Seattle, WA, USA, Aug. 2006, pp. 227–286.

SEUNGHWAN SON received the B.S. degree in mathematics from Kyungpook National University, Daegu, South Korea, in 2019, where he is currently pursuing the M.S. degree with the School of Electronic and Electrical Engineering. His research interests include authentication, blockchain, cryptography, and information security.



JOONYOUNG LEE received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the School of Electronic and Electrical Engineering. His research interests include authentication, the Internet of Things, and information security.

MYEONGHYUN KIM received the B.S. and M.S. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree with the School of Electronic and Electrical Engineering. His research interests include authentication, blockchain, the Internet of Things, and information security.

SUNGJIN YU received the B.S. degree in electronics engineering from Daegu University, Daegu, South Korea, in 2017, and the M.S. degree in electronics engineering from Kyungpook National University, Daegu, in 2019, where he is currently pursuing the Ph.D. degree with the School of Electronic and Electrical Engineering. His research interests include authentication, post-quantum cryptography, VANET, blockchain, and information security.

ASHOK KUMAR DAS (Senior Member, IEEE) received the M.Sc. degree in mathematics, the M.Tech. degree in computer science and data processing, and the Ph.D. degree in computer science and engineering from IIT Kharagpur, India. He is currently an Associate Professor with the Center for Security, Theory and Algorithmic Research, International Institute of Information Technology Hyderabad, Hyderabad, India. His current research interests include cryptography, wireless sensor network security, hierarchical access control, security in vehicular ad hoc networks, smart grid, the Internet of Things (IoT), cyber-physical systems (CPS) and cloud computing, and remote user authentication. He has authored over 235 articles in international journals and conferences in the above areas, including over 200 reputed journal articles. Some of his research findings are published in top cited journals, such as the IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Smart Grid, IEEE Internet of Things Journal, IEEE Transactions on Industrial Informatics, IEEE Transactions on Vehicular Technology, IEEE Transactions on Consumer Electronics, IEEE Journal of Biomedical and Health Informatics (formerly IEEE Transactions on Information Technology in Biomedicine), *IEEE Consumer Electronics Magazine*, IEEE Access, *IEEE Communications Magazine*, *Future Generation Computer Systems*, *Computers & Electrical Engineering*, *Computer Methods and Programs in Biomedicine*, *Computer Standards & Interfaces*, *Computer Networks*, *Expert Systems with Applications*, and *Journal of Network and Computer Applications*. He was a recipient of the Institute Silver Medal from IIT Kharagpur. He is on the editorial board of IEEE Systems Journal, *Computer Communications* (Elsevier), *IET Communications*, *KSII Transactions on Internet and Information Systems*, and *International Journal of Internet Technology and Secured Transactions* (Inderscience), is a Guest Editor of *Computers & Electrical Engineering* (Elsevier) for the special issue on Big data and IoT in e-healthcare and for *ICT Express* (Elsevier) for the special issue on Blockchain Technologies and Applications for 5G Enabled IoT, and has served as a Program Committee Member in many international conferences. He has also served as one of the Technical Program Committee Chairs of the first International Congress on Blockchain and Applications (BLOCKCHAIN'19), Avila, Spain, June 2019, and the second International Congress on Blockchain and Applications (BLOCKCHAIN'20), L'Aquila, Italy, October 2020.

YOUNGHO PARK (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyungpook National University, Daegu, South Korea, in 1989, 1991, and 1995, respectively. From 1996 to 2008, he was a Professor with the School of Electronics and Electrical Engineering, Sangju National University, South Korea. From 2003 to 2004, he was a Visiting Scholar with the School of Electrical Engineering and Computer Science, Oregon State University, USA. He is currently a Professor with the School of Electronic and Electrical Engineering, Kyungpook National University. His research interests include computer networks, multimedia, and information security.