Received 2 July 2022, accepted 15 July 2022, date of publication 21 July 2022, date of current version 29 July 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3193238

RESEARCH ARTICLE

DDMIA: Distributed Dynamic Mutual Identity Authentication for Referrals in Blockchain-Based Health Care Networks

MANJUNATH HEGDE[1], ROHINI R. RAO[1], AND B. M. NIKHIL[2]
[1]Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India
[2]Department of Electronics and Communication, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India
Corresponding author: Rohini R. Rao (rohini.rao@manipal.edu)

ABSTRACT Patients go to multiple healthcare providers for treatment, and their health data is generally distributed among providers. The distributed health data and the decentralized health care system structure make it ideal for blockchain-based health information systems. The authors consider the referral use case; for instance, a patient goes to his primary health Centre (PHC) for treatment and is referred to a hospital. Authentication is usually done using certificates or key cryptography, which could become cumbersome when multiple parties are involved in a healthcare interaction. The security requirements were defined, and a novel multi-party, mutual patient identity authentication scheme called ''Distributed Dynamic Mutual Identity Authentication (DDMIA)'' was proposed for the referral use case in a blockchain-based e-health network. The DDMIA enables the PHC to authenticate the patient to the referred hospital. The DDMIA scheme was designed using Elliptic Curve Cryptography. It was proven to be secure by assuming the hardness of the elliptic curve discrete log problem (ECDLP) and Elliptic curve computational Diffie–Hellman problem (ECDH) using CK-Model. The formal security analysis using BAN logic proved that the sessions are secure after authentication. The DDMIA scheme was simulated in the AVISPA tool and proven safe against all active attacks. The scheme allows a patient to be authenticated by multiple parties without registering with all parties. It eliminates the need for multiple registration centers as well as digital certificates. Hence, the DDMIA scheme can be implemented for similar multiparty authentication requirements in blockchain-based networks.

INDEX TERMS Blockchain, referral, e-health, health data exchange, distributed identity authentication, multi-party authentication.

I. INTRODUCTION

Patients visit various hospitals, private clinics, and public health centres for their health needs. Each of these healthcare providers generate and record health information about the patient [1]. There is a need to share the patient's health and medical history among healthcare providers, for informed medical decisions, which results in a better quality of healthcare. Technology adoption can improve the quality of healthcare as well as bring down the cost. The national e-health initiatives suggest the adoption of Electronic Health Records to create and maintain a longitudinal electronic record of the patient health information. Policy regulations for aspects like data exchange, data ownership, privacy protection, and security have been set. In general, the adoption of technology for the healthcare sector is low in most countries [2]. Some private hospitals have adopted Hospital Information Systems and electronic medical records, however, seamless data exchange and comprehensive patient health records are not available.

Blockchain technology has the potential to transform the healthcare industry [1]. The authors suggest the use of

The associate editor coordinating the review of this manuscript and approving it for publication was Diana Gratiela Berbecaru.

blockchain technology for integrating health data into EHRs as well as seamless data exchange [3]–[7]. Various private and public e-health providers can be connected on a network to enable data integration and sharing. The patient's interactions with e-health care providers can be recorded as transactions in a trusted network without the involvement of third parties. Blockchain technologies will record the distributed health interactions and enable integration of the data into a longitudinal EHR. Blockchain-based data exchange will ensure the completeness as well as the immutability of the patient's health interactions. Several organizations are using blockchain technology for health records. For instance, the Gem Health Network, OmniPHR deploy blockchain-based technology to share patient records in a seamless environment [5]. MedRec is a decentralized record management system to handle EMRs using blockchain technology [8].

Irrespective of the technology used, e-health providers are required to adopt administrative, physical, and technical safeguards to ensure the privacy, confidentiality, integrity, and availability of the e-health Data [9]. e-health providers need to implement access controls, authentication, and nonrepudiation of health records [5]. This article addresses the authentication in blockchain-based health data exchange. There is a need to verify the identity of the person or entity involved in the e-health interaction. Authentication services verify the person or entity seeking data access in the network. Identity confirmation is usually done using public-key cryptography or a DNS based authentication using an existing and widely accepted form of identification such as social security number. In some implementations, the provider nodes and patient nodes are authenticated using consensus methods facilitated by miners using the Ethereum technology stack.

An important technical barrier in blockchain-based health data exchange is the need for entity authentication to be robust, it must be repeated for every entity-to-entity relationship [6]. We consider a typical use case in Indian Public Health scenarios, which is the referral [10]. The patient approaches his/her primary e-health provider who may refer the patient to another healthcare provider. For instance, a patient approaches the local Public Health Centre (PHC) for treatment. In most cases, the patient is treated in the Public Health Centre by the nurses or midwives. Sometimes the patient cannot be treated at the PHC, he/she will be referred to a doctor at the nearest Government Hospital (GH). Our blockchain-based EHR implementation connects the three parties, the patient and the two e-health providers on a private permissioned blockchain network. There is a need to authenticate the user's legality while requesting or modifying the patient's health data. This article focusses on the aspect of authentication of the multiple parties involved in this particular e-health setting.

A. MOTIVATION AND RESEARCH CONTRIBUTION

In blockchain-based Health Information systems, the authentication is mostly done using digital certificates and signatures. When multiple e-health providers are involved in the authentication, then each organization has to set up a certificate authority to generate the certificates. In the multiparty authentication scenario, certificates and signatures are not ideal for authentication. Consider the scenario of a patient who has been referred from Hospital A to Hospital B. The medical or health history of the patient has to be sent from Hospital A to Hospital B. This would require the patient to be registered at both hospitals. The multiple certificates may cause a collision on identity and key management. Besides, if the certificate itself is corrupt, then the authentication fails, and the transaction is considered invalid. In the case of public key-based authentication methods, there are other challenges. The patient is required to manage the private keys for authentication among multiple e-health providers. Many key management solutions are unable to manage the patient's key pair while using various cryptographic mechanisms [11], [12]. Typically, key management issues occur: (1) When the system compromises its secret key and (2) When the number of patients is high. When the secret key is compromised, the security of the blockchain-based data is not guaranteed [8].

Based on the use case, the security requirements of the proposed multiparty authentication scheme were listed as follows:

- Mutual authentication: The entities involved in the transaction should be authenticated to each other before beginning the transaction.
- Quick credentials verification: When multiple entities are involved in the authentication, the communication overhead should be minimal. The patient credentials should be authenticated before sending the login request to the PHC server.
- Patient anonymity: The system should not reveal the identity of the patient. No potential attacker should be able to obtain the patient's original identity during authentication.
- Secret key management: Key management focuses on securing the secret key. The secret key will be involved in the credentials/authentication parameter encryption and decryption operations. The secret key is also used to calculate the session key, which is necessary to create a secure channel over an insecure communication environment. If the server's secret key is compromised then the whole session will be vulnerable.
- Session key agreement: The session key creates a secure channel over an insecure communication environment. All three parties should share a common session key among them to establish a secure channel between them. The shared session key should be confidential. If it is revealed, the entire session will be insecure.
- Perfect forward secrecy: This ensures that the session key will not be revealed to the adversary even if the secret key of the server has been compromised.
- Resilience against insider attack: The administrator or a person in the registration centre should not be able to access the password of the entities participating in the authentication

- Prevention of replay attack: In this attack, the adversary intercepts the previously successful login messages. The attacker resends or replay the obtained message and tries to enter into the system. Therefore the system should verify the freshness or validity of the message before authenticating it.

The objective of this article is to present a multi-party authentication scheme and key management system for referrals in a blockchain-based e-health network. A distributed, dynamic, mutual identity authentication (DDMIA) scheme for patients in the blockchain network has been designed and implemented. The authentication scheme is distributed among the parties involved in the referral. The authentication schemes preserve patient anonymity by computing a dynamic identity for every session. One party mutually authenticates the patient to another party without expecting the patient to register with multiple healthcare providers.

In general, the proposed blockchain-based authentication scheme is suitable for authentication among multiple parties involved in the communication. In this scenario, DDMIA avoids repeated registration with multiple healthcare providers. Also, the scheme is independent of the third party for authenticating all active participants involved in the communication. Hence, DDMIA reduces the overall time and computation required for traditional multi-party authentications. The proposed scheme security is formally proved in the CK-model. Using BAN logic we proved that the proposed scheme secures the sessions after authentication. Also, the proposed scheme is simulated in the AVISPA tool to prove that the scheme is safe against all active attacks. The security features of the proposed work for blockchain-based multi-party authentication has been proved by cryptanalysis.

B. RELATED WORK

Till date, several authentication schemes were proposed in the e-health sector. Chen *et al.* [14] proposed an authentication scheme for cloud-based electronic medical records which focusses on withstanding impersonation attack, replay attack, and man-in-middle attack. Later Chiou *et al.* [15] identified the weaknesses in Chen *et al.* scheme [14], they proved that it does not support patient anonymity, message authentication, and support telemedicine. Mohit *et al.* [16] reviewed Chiou *et al.* scheme [15] and identified that it does not support patients' anonymity and is vulnerable to stolen device attacks. Mohit *et al.* proposed an authentication scheme for cloud-based e-health systems, which overcomes the identified weaknesses. In the same year, Cheng *et al.* [17] reviewed Chiou *et al.* scheme [15] and identified the Key compromise impersonation and the forward secrecy issues in the scheme. Cheng *et al.* proposed an authentication scheme based on Bilinear pairing to achieve security. In 2018 Li *et al.* [18] reviewed Mohit *et al.* scheme and found that the scheme does not support patient anonymity and patient unlinkability. Also, it identified that the scheme is insecure against report revelation and report forgery attacks. Also, Li *et al.* proposed an authentication scheme to overcome the identified weaknesses.

Several blockchain-based authentication schemes have been proposed. In 2018, Wang *et al.* [19] proposed a blockchain-based mutual authentication scheme. In this scheme, they claimed that their scheme's authentication parameters would not be stored in the database. That makes the scheme independent from third parties during the authentication process. In 2019 Conti *et al.* [20] proposed a blockchain-based distributed authentication scheme to enable secure and efficient mobility management in information-centric networking. In 2019 Wang *et al.* [21] introduced a blockchain-based mutual authentication and key agreement scheme for smart grid infrastructure. This scheme elaborates on the conditional anonymity, active participation, and mutual authentication between the participants. In 2019, Liu *et al.* [22] proposed a Medibchain-based privacy-preserving mutual authentication scheme for the telecare medical information system. The scheme based on elliptic curve cryptography and focused on building a MediBchain-based system for mobile medical cloud architecture. Also, it provides security to sensitive data like patient identity. In 2020 Khalid *et al.* [23] proposed A decentralized lightweight blockchain-based authentication mechanism for IoT systems. This scheme was based on fog computing technology and built for a public blockchain.

Finally, the multi-party authentication schemes were reviewed. In 2017, Odelu *et al.* [24] proposed a multiparty authentication scheme using elliptic curve cryptography. In 2016 Park and Park [25] reviewed Chang *et al.*'s authentication and key agreement scheme proposed in 2015 and identified that Chang *et al.*'s scheme does not provide sufficient security and fails to provide accurate password updates. To overcome the identified weaknesses, Park-Park proposed a three-factor user authentication and key agreement scheme using the elliptic curve cryptosystem. In 2017 Amin *et al.* [26] proposed an anonymous and robust multi-server authentication protocol using multiple registration servers to manage a large number of users. Later, Qi *et al.* [27] proposed a biometrics-based authentication key exchange protocol for multi-server Telecare Medical Information System (TMIS) in 2018. The scheme aimed to secure the system's private key by not sharing it with the authentication process participants.

In 2020 Xiang *et al.* [28] proposed a permissioned blockchain-based identity management and user authentication scheme for e-Health systems. This scheme authenticates the users and medical servers through the registration center. In 2020, Li *et al.* [29] a blockchain-based data aggregation and group authentication scheme for the electronic medical system is proposed. Further, in 2020, Cui *et al.* [30] proposed a hybrid blockchain-based authentication scheme for multi wireless sensor networks. Here, the authors proposed an authentication scheme that performs between multiple wireless sensor networks and designed a hybrid blockchain for the network model. In 2021 Gao *et al.* [31] proposed a privacy-preserving identity authentication scheme based on the blockchain. In this scheme, users will generate their own identities and their publicly verifiable information. This public information is stored on the blockchain.

From the literature review, we observed that (i) most of the authentication schemes presented have security issues, and their improved schemes are also vulnerable to security attacks. (ii) Many schemes perform only two-party authentication and if the scheme performs multi-party authentication, then dependency over registration center(RC)/trusted third party is prevelant. (iii) Dependency on the third party is always a bottleneck for the system efficiency while handling large incoming requests [50]. Therefore, It is very much necessary to propose a distributed authentication scheme that can mutually authenticate patients and hospitals without the involvement of any third party. Based on the literature, we formulated the security requirements and proposed an appropriate multi-party authentication called Distributed Dynamic Mutual Identity Authentication (DDMIA).

C. METHODOLOGY

The proposed work appears in three stages. (i) the First stage of the work was to set up a blockchain for the healthcare network. (ii) Next stage is to propose a distributed dynamic user authentication scheme and related smart contract algorithms by considering the security requirements of the multi-party authentication scheme. (iii) the Last stage is to analyze the security and the performance of the proposed authentication scheme. The following subsections briefly illustrate how these stages are implemented.

1) SET UP A BLOCKCHAIN FOR HEALTHCARE NETWORK

In this work, the blockchain was set up using Hyperledger Fabric which is an open-source private permissioned blockchain network from the Linux foundation. In the proposed work, network consists of multiple parties such as a patient, PHCs, private clinics, private and government hospitals. Instances of stakeholders were created and the DDMIA scheme was plugged in for authentication. The DDMIA scheme is used for the referral cases between any two parties in the system. For illustration purposes, the authors considered a scenario where a patient is registered to the local Primary Health Centre (PHC_i) and visits the PHC_i for treatment. When the PHC_i is unable to treat the patient on its own, it refers the patient to the Government Hospital (GH_i). The DDMIA scheme does not expect the patient to register to the GH_i. Instead, the patient's identity is authenticated to the GH_i by the PHC_i.

2) PROPOSE AUTHENTICATION SCHEME AND RELATED SMART CONTRACT

The proposed DDMIA scheme uses the Elliptic Curve Cryptosystem (ECC) for random variable generation and message communication. An elliptic curve is a cubic equation of the form $y^2 + a_1xy + a_2y = x^3 + a_3x^2 + a_4x + a_5$ where a_1, a_2, a_3, a_4, a_5 are real numbers. In an elliptic curve cryptosystem (ECC), the equation is defined as $E_p(a, b)$: $y^2 = x^3 + ax + b(mod\ p)$ over a prime finite field F_p, where a, b, F_p and the point multiplication over $E_p(a, b) : s * P = P + P + P + \ldots.. + P$ [32].With respect to the operations performed during authentication, necessary smart contract algorithams are proposed.

3) DDMIA SECURITY AND PERFORMANCE ANALYSIS

In this stage, we made rigorous cryptanalysis and proved that the DDMIA scheme resists several active attacks. The proposed scheme security is formally proved in the CK - model. DDMIA is also simulated through the AVISPA tool. AVISPA provides different backends to verify the specified security. In AVISPA scheme can be implemented by the High-Level Protocol Specification Language (HLPSL). It is a role-based language where every participant in the system is considered as a role. Each role is independent and communicates through channels with other roles. The session security of DDMIA has been analyzed using a formal method called BAN logic [33]. Computation, communication, and functional analysis have been made and compared with other recently proposed schemes to verify the robustness of DDMIA. The DDMIA authentication scheme has been implemented in the GO language using crypto libraries to do the analysis. The GO-based DDMIA scheme was plugged-in to the hyperledger fabric for authentication [34]–[36].

The rest of the article is assembled as follows. Section II illustrates the Proposed DDMIA scheme in Blockchain-based e-health networks. This section includes the system design for DDMIA scheme (II.A), Proposed DDMIA scheme (II.B), and Algorithm design for smart contracts (II.C). Further, Section III discusses the security proofs of the DDMIA scheme namely cryptanalysis of DDMIA (III.A) Formal security proof using the CK-model (III.B), Formal security verification using the AVISPA tool (III.C), and Formal analysis using BAN Logic (III.D). Section IV presents the result analysis of DDMIA in terms of Performance analysis (IV.A), which includes the computation and communication costs analysis, and functional analysis between DDMIA and other related schemes. Finally, Section V presents the discussion about the security and the concluding remarks of this article.

II. PROPOSED DDMIA SCHEME

In this section, we proposed distributed dynamic authentication scheme for referrals in blockchain-based health care networks. Firstly the system model of the proposed authentication scheme has been presented. Further, every phase of the authentication scheme is explained. Finally the necessary smart contract algorithms are presented.

A. SYSTEM MODEL

For the demonstration purpose, we showcase a blockchain-based system model wherein the Patient (P_i), a Primary Health Centre (PHC_i), and the referred Government Hospital (GH_i) are involved in the communication. The system design for DDMIA is presented in Figure 1. The stakeholders are connected via a private permissioned blockchain network. In the designed system, a consortium of health care providers will function as a Registration Center (RC) and ensure the registration of all health care providers in the network. RC is responsible for generating the public and private keys and computing the registration parameters using selected credentials. The registration process of all participants is done through the secure channel where the communication messages are cannot be intercepted. The patient registers at the local PHC_i only In the case of a referral, P_i is referred from PHC_i to GH_i, the patient is authenticated by GH_i even though the patient is not registered in the GH_i system. This is because, the PHC_i mutually authenticates the P_i to the GH_i. The authentication process is done in an insecure channel where the communication parameters can be intercepted and modified. Hence there is a need for security of communication parameters. The system is based on the blockchain network and there are four smart contract algorithms SM 1 (REFAUTHInitialization), SM 2 (InsertREFAUTH), SM 3 (ModifyREFAUTH), and SM 4 (ReadREFAUTH) which are used for initialization of keys, insert parameters, update parameters and read parameters.

B. PROPOSED DDMIA SCHEME

The proposed scheme contains four phases (1)Initialization (2) Registration, (2) Login and authentication, and (4) Password change. The notations used throughout the proposed scheme is given in Table 1.

TABLE 1. Notations and descriptions.

Notations	Descriptions
RC	Registration Center
GH_i	Government Hospital
PHC_i	Primary Health Centre
P_i	Patient
ID_i	Patient Identity
x	Secret key maintained by RC
PW_i	Password
$\oplus$	Bitwise XOR operator
$\|$	Concatenation operator
$h_1(.), h_2(.), h_3(.)$	One-way hash functions
T_1, T_2, T_3	Timestamps generated by P_i, PHC_i, GH_i

1) THE INITIALIZATION PHASE

Before stepping into the registration, the system initializes some parameters. Registration Center (RC) selects an elliptic curve E_p over the finite field F_p with a large prime number ' p'. RC also chooses a one-way hash functions $h_1(.) \rightarrow Z^*, h_2(.) \rightarrow Z^*, h_3(.) \rightarrow Z^*$ and a point on the elliptic curve 'P' of order 'n'. Further RC selects 'x' as the master key and computes the public key $P_{pub} = x.P$ and publishes the parameters $\{E_p, P, F_p, h_1(.), h_2(.), h_3(.), P_{pub}\}$.

2) THE REGISTRATION PHASE

The registration phase contains the steps to register the communication participants. This phase includes the registration of Patient (P_i) and Primary Health Center (PHC_i) and the Hospitals (GH_i). The registration of PHC_i and GH_i has been done through a registation center. The patient register to Primary Health Center. The details of the participants registration have been presented below.

3) HOSPITALS (GH_i)/PRIMARY HEALTH CENTER (PHC_i) REGISTRATION PHASE

In the proposed scheme, GH_i/PHC_i registration is done with a registation center (RC). The registration procedure of GH_i and PHC_i is same. For the time being, we illustrate only PHC_i registration:

1) PHC_i selects an identity PID_i, a random value b_j and computes $A_i = h_1(PID_i \| b_j)$
2) PHC_i sends a registration request message $\{PID_i, A_i\}$ to the RC.
3) RC receives the request message, generates the random number 'e' and computes
$m_i = h_1(x \| e)$,
$Z_n = h_1(m_i \| A_i)$ and
$H_n = h_1(Z_n \| PID_i)$
4) RC stores e, m_i into the database and sends Z_n to the PHC_i. Primary Health Center receives Z_n and stores $\{Z_n b_j\}$ into its database. The registration phase of PHC_i has been presented in the Table 2.

In case of hospital registration GH_i selects identity HID_i, random value b_j and computes and computes A_j and sends the request message to RC. The computed parameters of RC are m_j, Z_m, and H_m.

TABLE 2. Registration phase of the proposed scheme.

Primary Health Center	Registation Center
Selects PID_i and b_j Computes $A_i = h_1(PID_i \| b_j)$ $\xrightarrow{\{PID_i, A_i\}}$	
	Receives message and computes Generates 'e' and computes $m_i = h_1(x \| e)$, $Z_n = h_1(m_i \| A_i)$ $H_n = h_1(Z_n \| PID_i)$ Stores e, m_i into the database $\xleftarrow{\{Z_n\}}$
Stores $\{Z_n b_j\}$ into its database.	

4) PATIENT (P_i) REGISTRATION PHASE

The registration the patient is done with PHC_i. The steps involved in this process are as follows:

1) Patient selects his/her ID_i, and PW_i and computes $RPW_i = h(ID_i \| PW_i)$
2) P_i sends the registration request message $\{ID_i, RPW_i\}$ to the PHC_i.
3) PHC_i receives the request message and computes $R_i = ID_i \oplus h_1(m_i \| RPW_i)$ $V_i = h_1(R_i \| ID_i \| RPW_i)$ and $GID_i = h_1(V_i).P_{pub}$.

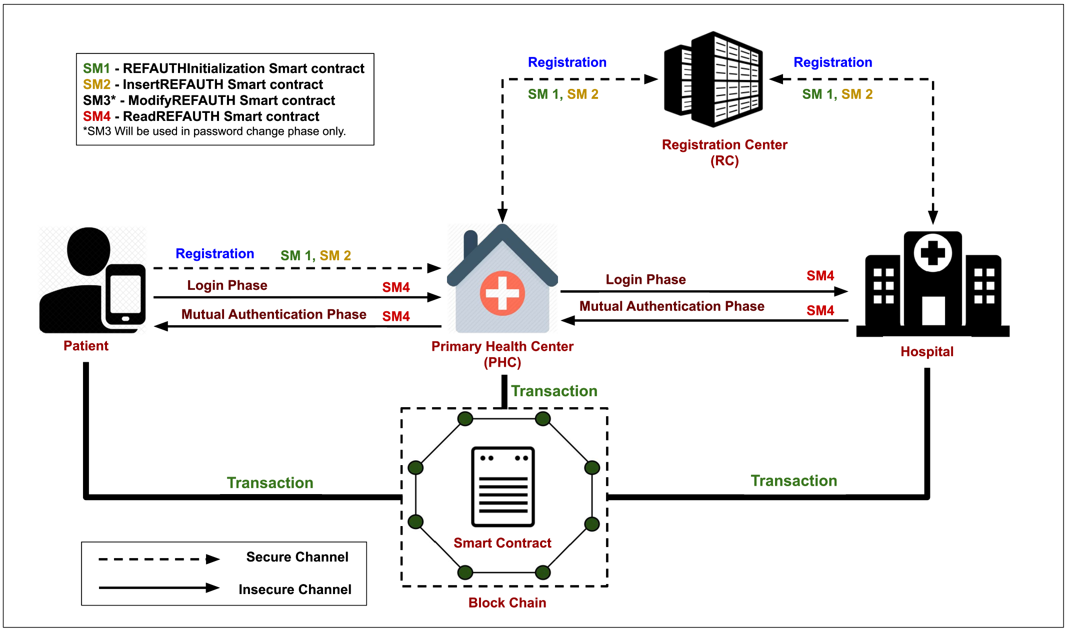

FIGURE 1. System design for proposed DDMIA.

TABLE 3. Registration phase of the proposed scheme.

Patient	Primary Health Center
Choose ID_i, PW_i Compute $RPW_i = h(ID_i \| PW_i)$ $\xrightarrow{ID_i, RPW_i}$	
	Receives message and computes $R_i = ID_i \oplus h_1(m_i \| RPW_i)$ $V_i = h_1(R_i \| ID_i \| RPW_i)$ and $GID_i = h_1(V_i).P_{pub}$ Uploads V_i, m_i into a smart contract $\xleftarrow{GID_i}$
Stores GID_i into its database.	

4) PHC_i uploads $\{V_i, m_i\}$ into a smart contract using the Algorithm **InsertREFAUTH** and sends GID_i to the patient.
5) The patient receives GID_i and stores it into its database.

The registration phase of the proposed scheme has been presented in Table 3.

5) THE LOGIN AND AUTHENTICATION PHASE

The login and authentication phase is done between Patient P_i, Primary Health Centre PHC_i, and the Hospital GH_i. Since the authentication is distributed, there is no involvement of the registration center in this phase. Table 3 presents the Login and authentication phase of the proposed scheme.

1) Patient P_i inputs ID_i and PW_i. Further, P_i system retrieves the stored parameters in the smart contract using the Algorithm 4 **ReadREFAUTH** and Computes $RPW_i^* = h(ID_i \| PW_i)$,
$R_i^* = ID_i \oplus h_1(m_i \| RPW_i)$
$V_i^* = h_1(R_i^* \| ID_i \| RPW_i^*)$
$GID_i^* = h_1(V_i^*).P_{pub}$
2) Verifies the condition $GID_i^* = GID_i^*$ or not. If both are not equal then entered ID_i and PW_i are incorrect and the system terminates the session.
3) If the input credentials are correct then the patient system generates random number ‘ w’and computes $C_u = w.P_{pub} \oplus h_2(GID_i \| m_i \| T_1)$,
$CID_i = h_2(w.P_{pub} \| T1) \oplus RPW_i$,
$C_1 = h_2(CID_i \| m_i \| w.P_{pub})$
4) Patient system sends a login request message $M_1 = \{C_u, CID_i, C_1, T_1\}$ to the PHC_i.
5) On the other side PHC_i receives the request message M_1 from P_i and verifies the freshness of the received message. PHC_i takes its present system time T_2 and verifies the validity of the received message. First PHC_i checks the condition $T_2 - T_1 \leq \Delta T$. Also confirms that there is no other request with the same parameter within the period of $(T_1 + \Delta T)$ and $(T_1 - \Delta T)$. If the above conditions are true then the PHC_i performs the further calculation, else it rejects the request message M_1 and drops the session.
6) After accepting the request message M_1 from the patient P_i, PHC_i retrieves the parameters from the smart contract using the Algorithm 4 **ReadREFAUTH** and computes $w.P_{pub}$’$= h_2(GID_i \| m_i \| T_1) \oplus C_u$
$C_p = C_u \oplus h_2(Z_n \| m_j \| T_2)$
$CID_j = h_2(C_u \| C_p \| w.P_{pub}’)$ and
$C_2 = h_2(w.P_{pub}’ \| C_1 \| CID_i \| CID_j \| T_1 \| T_2)$.
7) PHC_i creates the login request message $M_2 = \{C_p, C_2, T_2\}$ and sends it to the GH_i along with M_1.

8) GH_i receives the request message $\{M_1, M_2\}$ from PHC_i and verifies its freshness. GH_i takes the present time T_3 and checks the condition $T_3 - T_2 \leq \Delta T$. If the condition does not satisfy then GH_i drops the session.
9) If the condition is true then GH_i retrieves the parameters in the smart contract using the Algorithm 4 **ReadREFAUTH** and computes $w.P_{pub}$'$= h_2(GID_i \| m_i \| T_1) \oplus C_u$
CID_j'$= h_2(C_u \| C_p \| w.P_{pub}$'$)$
C_2'$= h(w.P_{pub}$'$\| C_1 \| CID_i \| CID_j$'$\| T_1 \| T_2)$.
10) Compares C_2'$= C_2$. If the condition is true, then P_i and PHC_i are authenticated by GH_i. Else the system drops the session.
11) After authentication of P_i and PHC_i, GH_i starts mutual authentication. Here, system generates random number 'y' and computes $C_k = y.P_{pub} \oplus h_3(m_j \| T_3)$
$SK_{gh} = h_3(w.P_{pub}$'$\| y.P_{pub} \| m_i \| m_j)$ and
$C_3 = h(SK_{gh} \| T_3 \| y.P_{pub})$.
GH_i sends $M_3 = \{C_3, C_k, T_3\}$ to the PHC_i for mutual authentication.
12) PHC_i receives M_3 and computes
$y.P_{pub}$'$= h_3(m_j \| T_3) \oplus C_s$
$SK_{phc} = h_3(w.P_{pub}$'$\| y.P_{pub}$'$\| m_i \| m_j)$ and
$C_4 = h(SK_{phc} \| C_3 \| T_3 \| w.P_{pub}$'$\| y.P_{pub}$'$)$
13) PHC_i creates mutual authentication message $M_4 = \{C_4\}$ and sends to Patient P_i along with M_2.
14) P_i receives the mutual authentication message from PHC_i and computes $y.P_{pub}$'$= h_3(m_j \| T_3) \oplus C_s$, $SK_p = h_3(w.P_{pub} \| y.P_{pub}$'$\| m_i \| m_j)$ and
C_4'$= h(SK_p \| C_3 \| T_3 \| w.P_{pub} \| y.P_{pub}$'$)$
15) Patient system verifies whether C_4'$= C_4$ If both are equal then P_i, PHC_i and GHS is authenticated. Else drops the session.

Further communications will be done through the shared session keys. The session keys are For patient, $SK_p = h_3(w.P_{pub} \| y.P_{pub}$'$\| m_i \| m_j)$, For PHC_i $SK_{phc} = h_3(w.P_{pub}$'$\| y.P_{pub}$'$\| m_i \| m_j)$, For GH_i $SK_{ghc} = h_3(w.P_{pub}$'$\| y.P_{pub} \| m_i \| m_j)$

6) THE PASSWORD CHANGE PHASE

In this phase, P_i changes the password PW_i to PW^{new}. Procedure to change the password is given as follows:

1) P_i inputs ID_i and PW_i to the system. The system Retrieves the parameters in the smart contract using the Algorithm 4 **ReadREFAUTH** and Computes $RPW_i^* = h_1(ID_i \| PW_i)$, $R_i^* = ID_i \oplus h_1(m_i \| RPW_i)$, $V_i^* = h_1(R_i^* \| ID_i \| RPW_i^*)$ and $GID_i^* = h_1(V_i^*).P_{pub}$
2) Verifies the condition $GID_i^* = GID_i^*$ or not. If both are equal then input ID_i and PW_i is correct and system asks for new password PW^{new}. Further, System computes
$RPW_i^{new} = h_1(ID_i \| PW^{new})$
$R_i^{new} = ID_i \oplus h_1(m_i \| RPW_i^{new})$
$V_i^{new} = h_1(R_i^{new} \| ID_i \| RPW_i^{new})$,
$GID_i^{new} = h_1(V_i^{new}).P_{pub}$
3) Updates the parameters $(V_i)^{new}$ into the smart contract using the algorithm 3 **ModifyREFAUTH** and GID_i^{new} in the patient system.

C. ALGORITHM DESIGN FOR SMART CONTRACTS

The smart contract algorithm design for phases of DDMIA is presented below:

- **SM 1 - REFAUTHInitialization:** This algorithm is used to initialize the parameters space for GID, V, m. Here, the system creates RAUTH [] to store the parameter and the type of the parameter by creating. The parameter initialization has been presented in Algorithm 1.

Algorithm 1 REFAUTHInitialization

```
contract REFAUTH {
address patient;
struct RAUTH {
byte32GID;
uint256[2]V;
uint256[4]m; }
RAUTH[] public REFAUTH;
constructor REFAUTH() {
patient = msg.sender;
len = 0;
return1; }
}
```

- **SM 2 - InsertREFAUTH:** To insert the registration parameters into the system, we need the InsertREFAUTH algorithm, which is presented in Algorithm 2. Here, the system first checks the sender details. If the sender already exists, then smart-contract returns zero else stores the registration parameters through $RAUTH[i]$.

Algorithm 2 InsertREFAUTH

```
function insertREFAUTH (GID, V, m){
if patient ≠ msg.sender then
return 0;
else {
if Exist(RAUTH[i].GID == GID) then {
RAUTH[i].GID = GID;
RAUTH[i].V = V;
RAUTH[i].m = m;
return1; }
else{
Return 0;}
}
```

- **SM 3 - ModifyREFAUTH:** Suppose the patient wishes to change his/her password; new parameters will be computed by the system. In this case, to update the new parameters into the system, the ModifyREFAUTH algorithm will be used. Here, the system checks whether

TABLE 4. Login and authentication phase of the proposed scheme.

Patient	PHC_i	GH_i
Inputs ID_i and PW_i Computes $RPW_i^* = h(ID_i \| PW_i)$, $R_i^* = ID_i \oplus h_1(m_i \| RPW_i)$ $V_i^* = h_1(R_i^* \| ID_i \| RPW_i^*)$ $GID_i^* = h_1(V_i^*).P_{pub}$ Verifies the condition $GID_i^* = GID_i^*$ Generates 'w' Computes $C_u = w.P_{pub} \oplus h_2(GID_i \| m_i \| T_1)$ $CID_i = h_2(w.P_{pub} \| T1) \oplus RPW_i$, $C_1 = h_2(CID_i \| m_i \| w.P_{pub})$ $\xrightarrow{\{M_1 = C_u, CID_i, C_1, T_1\}}$		
	Takes the present time T_2 verifies $T_2 - T_1 \leq \Delta T$ Retrieves stored parameters computes $w.P_{pub}$'$= h_2(GID_i \| m_i \| T_1) \oplus C_u$ $C_p = C_u \oplus h_2(Z_n \| m_j \| T_2)$ $CID_j = h_2(C_u \| C_p \| w.P_{pub}$') and $C_2 = h_2(w.P_{pub}$'$\| C_1 \| CID_i \| CID_j \| T_1 \| T_2)$ Creates the login request message $M_2 = \{C_p, C_2, T_2\}$ $\xrightarrow{\{M_1, M_2\}}$	
		Receives the request message takes the present time T_3 checks the condition $T_3 - T_2 \leq \Delta T$ Retrieves the stored parameters $w.P_{pub}$'$= h_2(GID_i \| m_i \| T_1) \oplus C_u$ CID_j'$= h_2(C_u \| C_p \| w.P_{pub}$') C_2'$= h(w.P_{pub}$'$\| C_1 \| CID_i \| CID_j$'$\| T_1 \| T_2)$ Compares C_2'$= C_2$. Starts mutual authentication. Generates random number y and computes $C_k = y.P_{pub} \oplus h_3(m_j \| T_3)$ $SK_{gh} = h_3(w.P_{pub}$'$\| y.P_{pub} \| m_i \| m_j)$ and $C_3 = h(SK_{gh} \| T_3 \| y.P_{pub})$. Sends $M_3 = \{C_3, C_k, T_3\}$ to the PHC server $\xleftarrow{M_3 = C_3, C_k, T_3}$
	Receives M_3 and computes $y.P_{pub}$'$= h_3(m_j \| T_3) \oplus C_s$ $SK_{phc} = h_3(w.P_{pub}$'$\| y.P_{pub}$'$\| m_i \| m_j)$ $C_4 = h(SK_{phc} \| C_3 \| T_3 \| w.P_{pub}$'$\| y.P_{pub}$') Creates mutual authentication message $M_4 = \{C_4\}$ Sends $\{M_3, M_4\}$ to Patient P_i $\xleftarrow{\{M_3, M_4\}}$	
Receives $\{M_3, M_4\}$ and computes $y.P_{pub}$'$= h_3(m_j \| T_3) \oplus C_s$ $SK_p = h_3(w.P_{pub} \| y.P_{pub}$'$\| m_i \| m_j)$ C_4'$= h(SK_p \| C_3 \| T_3 \| w.P_{pub} \| y.P_{pub}$') Verifies whether C_4'$= C_4$ or not.		
$SK_p = h_3(w.P_{pub} \| y.P_{pub}$'$\| m_i \| m_j)$	$SK_{phc} = h_3(w.P_{pub}$'$\| y.P_{pub}$'$\| m_i \| m_j)$	$SK_{ghc} = h_3(w.P_{pub}$'$\| y.P_{pub} \| m_i \| m_j)$

the sender sent the same message or not. If yes, then smart-contract returns zero else updates the new parameters through *RAUTH*[*i*]. The ModifyREFAUTH has been presented in Algorithm 3.

- **SM 4 - ReadREFAUTH:** Whenever stored parameters are required; it will be retrieved using the ReadREFAUTH algorithm. *GID* can identify uniqueness in stored parameters. The algorithm checks whether the respective *GID* is exists or not. If *GID* is exist, it will be retrieved by *RAUTH*[*i*]. Algorithm 4 presents ReadREFAUTH.

III. SECURITY PROOF OF DDMIA SCHEME

In this section, we discuss the cryptanalysis of DDMIA scheme, formal security proof using CK-mode followed by the results of the security verification using the AVISPA tool. This section also presents the formal analysis of the proposed scheme using BAN logic.

A. CRYPTANALYSIS OF DDMIA SCHEME

The cryptanalysis of DDMIA is presented in this section. This analysis mainly focused on checking whether DDMIA meets all the security requirements illustrated in section 1.1 or not.

Algorithm 3 ModifyREFAUTH

```
function modifyREFAUTH (GID, V, m){
if patient ≠ msg.sender then
return 0;
else {
if Exist(RAUTH[i].GID == GID) then {
RAUTH[i].GID = GID;
RAUTH[i].V = V;
RAUTH[i].m = m;
return1; }
else{
len + +;
RAUTH[i].GID = GID;
RAUTH[i].V = V;
RAUTH[i].m = m;
return1; }
}
```

Algorithm 4 ReadREFAUTH

```
function readREFAUTH (GID){
if Exist(RAUTH[i] : GID == GID) then
return RAUTH;
Else;
return 0;
}
```

Considered threat model for the cryptanalysis of DDMIA is proposed by [37].

1) QUICK WRONG CREDENTIALS DETECTION

The proposed scheme detects the correctness of the login credentials (ID_i and PW_i) before login. When user inputs the ID_i and PW_i, the authentication scheme computes $RPW_i^* = h_1(ID_i \| PW_i$, $R_i^* = ID_i \oplus h_1(m_i \| RPW_i)$, and Verifies the equation $V_i.P = h_1(R_i^* \| ID_i \| RPW_i^*).Ppub$. This is to check the correctness of entered ID_i and PW_i. If both LHS and RHS are equal then the entered credentials are correct. This verification will be done before interacting with the PHC_i or GH_i, hence the proposed scheme verifies the credentials quickly.

2) PATIENT ANONYMITY

The anonymity of the patient identity has been preserved in each stage of communication. In the proposed scheme, the patient's IDi will not be communicated in plain text format to either PHCi or GHi. Instead of that, a dynamic ID $CID_i = h_2(w.P_{pub} \| T1) \oplus RPW_i$, will be computed in every login and authentication session. The dynamic ID is a temporary user identity computed using the patient's ID_i and it is different at every login attempt. Hence the proposed scheme provides anonymity of the patient identity.

3) SECRET KEY MANAGEMENT

It is essential to develop a scheme that protects the secret key from both the legal user and the adversary. In the DDMIA, the secret key is not used in plain text format for any operation. Whenever the RC receives the registration request message, it generates a random number 'e' and computes $m_i = h_1(x \| e).P$. But random number 'e'will not be stored either in P_i, PHC_i, and GH_i. Hence, DDMIA provides security to the secret key.

4) SESSION KEY WITH PERFECT FORWARD SECRECY

The DDMIA scheme ensures that all three parties should share a common session key among them to establish a secure channel between them. The session key forms a secure channel over a public channel. In the authentication mechanism, perfect forward secrecy is a feature that assures the confidentiality of the session key even after compromising the private/secret key. Let us assume adversary $\mathcal{A}$ attempts to compute the session key SK using the equation $SK = h_3(w.P_{pub} \| y.P_{pub} \| m_i \| m_j)$. Even though the adversary intercepts all the communicated parameters, he/she can get only m_i and m_j where $w.P$, $y.P$ are still unknown to the $\mathcal{A}$. Hence, the DDMIA scheme ensures perfect forward secrecy with session keys.

5) RESISTS INSIDER ATTACK

In the proposed scheme, password PW_i will not be submitted to the Registration Centre (RC) in a plain text format. Before sending the registration request, the client system computes $RPW_i = h_1(ID_i \| PW_i)$ and then sends $\{RPW, ID_i\}$ to RC. To obtain password PW_i from RPW_i adversary $\mathcal{A}$ should know both ID_i and PW_i. Hence the proposed scheme gives complete security against insider attack.

6) PROVIDES SECURITY AGAINST REPLAY ATTACK

To avoid the replay attack, the DDMIA uses the time stamp to verify the freshness of the received message. In the login and authentication phase of the proposed scheme, while sending the login request P_i system generates the timestamp T_1, includes it with the request message, and sends it to PHC Server. On the other side. PHC server takes the request message and generates its present time T_2 and verifies the validity of the time T_1. First PHC_i verifies the condition $T_2 - T_1 \leq \Delta T$ and also confirms, there is no other request with the same parameter within the period of $(T_1 + \Delta T)$ and $(T_1 - \Delta T)$. These conditions will be true if and only if the received message is fresh. Similarly, T_2 of the PHC server will be attached to the request message, which would be sent to the GH_i server. GH_i generates the timestamp T_3 and verifies the validity of T_2. The verification steps are the same as the procedure followed by the PHC server. Hence the DDMIA scheme resists the replay attack.

7) RESIST IMPERSONATION ATTACK

In this attack, An adversary $\mathcal{A}$ tries to impersonate valid P_i through the registration and communication parameters. In DDMIA scheme $\mathcal{A}$ does not get any patient information since the registration performed through the secure channel. Also, the parameters of the messages M_1, M_2, M_3 and M_4 are computed with atleast two unknown parameters which is not possible to guess by $\mathcal{A}$. Finally, the authentication of P_i done by PHC_i and GH_i which means, adversary must impersonate two entities involved in the process of authentication which is not possible. Thus, the proposed DDMIA scheme has the ability to resist the impersonation attack.

8) SECURITY AGAINST MAN-IN-THE-MIDDLE ATTACK

In this attack, an attacker $\mathcal{A}$ may try to impersonate a valid patient during the time of authentication. Since the authentication of P_i is done by PHC_i and GH_i, impersonating two entities involved in the authentication process is impossible. Hence, our scheme is secure against a man-in-the-middle attack.

9) PASSWORD SELECTION IS DONE BY USER

Many authentication schemes do not provide the feature of selecting their password by the user. If the system generates the password, then it is difficult to remember especially if the patient does not use the system frequently. Hence the DDMIA scheme allows the patient to choose a strong and memorable password.

B. FORMAL SECURITY PROOF USING CK-MODEL

This section presents the formal security analysis to prove that the proposed scheme is secure against the adversary modeled in [38] which is proposed by [39]. In this model, adversary $\mathcal{A}$ has complete control over the transmission channel. Therefore $\mathcal{A}$ can eavesdrop, intercept, alter the communication messages. Also, $\mathcal{A}$ knows all the public parameters. The adversary cannot access the secret parameter directly but can construct queries to capture the information leakage.

1) PARTICIPANTS

A participant in the entity takes part in the authentication process. In DDMIA scheme, there are three participants performing the authentication named as Patient (P_i), a Primary Health Centre (PHC_i), and the referred Government Hospital (GH_i). Each participant have multiple instances to run the scheme parallelly. The instances are represented as P^i, PHC^i, and GH^i, where 'i' is the i^th instance of the participants [40].

- $Execute(P^i, PHC^i, GH^i)$: This query forms the eavesdropping attack. Using this query, $\mathcal{A}$ simulates the login and authentication phase. In other words, $\mathcal{A}$ gets the transcript of the communication messages done between the instances P_i, PHC_i, and GH_i.
- $Send(P^i/PHC^i/GH^i, M)$: Adversary $\mathcal{A}$ models this query to perform active attacks. With this query, $\mathcal{A}$ intercept the message M communicated between the instances $P^i/PHC^i/GH^i$. Also, $\mathcal{A}$ tries modify the intercepted message. In other words, the query outputs a message M sent by participant $P^i/PHC^i/GH^i$.
- $EKeyReveal(P^i/GH^i)$: This query allows adversary to obtain the session state ephemeral secret key information held by the instance $P^i/PHC^i/GH^i$.
- $SKReveal(P^i/GH^i)$: This query allows adversary to get the session key held by the instance $P^i/PHC^i/GH^i$.
- $Corrupt(P^i/PHC^i/GH^i)$: This query express the notion of perfect forward secrecy where long term secret key can be compromise with $\mathcal{A}$ to get the session key on the oracle
- $Test(P^i/PHC^i/GH^i)$: This single query can be constructed by the adversary at most once. It models the semantic security of the session. Here, $\mathcal{A}$ returns the session key of $P^i/PHC^i/GH^i$ or a random string with an equal bit length of the session key. This result is depending upon tossing a coin b. If b = 1, the adversary gets the original session key. Else $\mathcal{A}$ gets a random string with the same length as the real session key.

We need to describe some definitions before proving the security of the proposed scheme.

- Partnering: When two entities are said to be partners if and only if they are accepted and shared a commen session key. In other words, If P_i, PHC_i and GH_i are partners only if $SK_p = SK_{phc} = SK_{gh}$.
- Freshness: The freshness is related to the session key. Here, oracle constructs the session key. We can say that the constructed session key is fresh if the instance meets the following conditions.
  1) When there is no *Reveal* query is done by P_i, PHC_i and GH_i, sission key SK_i should not be null.
  2) $Send(P^i/PHC^i/GH^i, M)$, should be asked after modelling the *Currupt* query
- Semantic Security: The goal of semantic security is to guess the bit $'b'$, which is involved in the $Test(P^i/PHC^i/GH^i)$ query. Consider an event $S()$ that the adversary $\mathcal{A}$ guess the bit b correctly. Let P^i, PHC^i and GH^i oracles are considered as partners when authenticating each other and share a common session key. The adversary's goal is to differentiate the session key from a random key. $\mathcal{A}$ can model many Test queries for $P^i/PHC^i/GH^i$. Consider queries, for instance, P^i. Further, P^i toss a coin b. If b=1, the adversary gets the original session key. Else A gets a random string with the same length as the real session key.

  Let $Pr[S]$ denotes the game-winning probability of $\mathcal{A}$. The advantage of the Adversary $\mathcal{A}$ against breaking the semantic security of the proposed scheme is $Adv_P^{AKE}(A) = |2Pr[Succ] - 1|$.

2) SECURITY PROOF

The security proof is based on the following computational problems:

- Elliptic curve computational Diffie–Hellman problem (ECDH): Let $P, xP, yP \in E_p$ where $a, b \in Z_q^*$, then

TABLE 5. Simulation of execute, revel and test query.

For a hash oracle $h(i,q)$ where $i=1,2,3$ if $(i,q,h)\in L_h$ Return h Else, Choose h and add to L_h as (i,q,h)
For $Execute(P^i, PHC^i, GHC^i)$ query $(CID_i, C_u, C_1, T_1, C_2, C_p, T_2) \leftarrow Send(CID_i, C_u, C_1, T_1)$ $(C_3, C_k, T_3) \leftarrow Send(CID_i, C_u, C_1, T_1, C_2, C_p, T_2)$ and $C_4 \leftarrow$ $Send(C_3, C_k, T_3)$ $Send(C_4)$ $Return(CID_i, C_u, C_1, T_1), (CID_i, C_u, C_1, T_1, C_2, C_p, T_2),$ $(C_3, C_k, T_3), (C_4)$
For $Revel(P^i/PHC^i/GH^i)$ query $ReturnSK_p$
For $Test(P^i/PHC^i/GH^i)$ query $SK_p \leftarrow Revel(P)$ $b \leftarrow \{0,1\}$ $SK_p \leftarrow \{0,1\}^{k'}$
For $Corrupt()$ query If $P = P_i$ Retutn RPW_i or A_i Else if $P = S$ Return A_i

it is hard to compute xyP in polynomial time without knowledge of x or y.

- Elliptic curve discrete logarithm problem (ECDLP): It says that when $G \in E_p(x,y)$ of order n and $G = kP \in E_p(x,y)$, it is computationally infeasible to compute k in polynomial-time.
- Reversing One way Hash function: Let $H(.)$ is a one way hash function, then it is computationally hard to get x from $H(x)$. Also it is hard to find x'where $H(x) = H(x')$

Theorem 1: Let E_p over the finite field F_p with a large prime number 'p' and $\mathcal{D}$ be the finite set of password. Consider $\mathcal{A}$ is a adversary running in a polynomial time and perfom security attack on the proposed scheme SC. Consider Adv_{SC}^{AKE} is the advantage of the $\mathcal{A}$ against the proposed scheme SC and also the advantage of $\mathcal{A}$ that solves CDH in E_p. If the adversary wants to break the protocol SC, then $\mathcal{A}$ can make q_s Send queries, q_h hash oracles, and q_e Execute queries within the time t. The advantage of $\mathcal{A}$ will be

$$Adv_{SC}^{AKE} \leq \frac{(q_s+q_e)^2}{2n} + \frac{(q_h)^2+(q_s)^2}{2^{k+1}} + q_h .Adv_{EC}^{ECDH}(t+(q_{exe}+q_{send})T_{EC}) \quad (1)$$

Proof: The sequence of games from G_0 to G_4 defines the proposed authentication scheme's proof. The queries constructed by $\mathcal{A}$ has been presented in Table 5 and 6. Based on the queries build by $\mathcal{A}$, the proof is presented. Let S_n denotes the event that occurs after the adversary's Test query while guessing the bit b correctly.

Game G_0 This game corresponds to the real game in the model. By definition, we have

$$Adv_{SC}^{AKE} \leq 2\,Pr[S_0] - 1$$

Game G_1 This game simulates the two hash oracles for each query. All queries manage two hash list L_h and L_h'.

TABLE 6. Simulation of execute, revel and test query.

For $Send(P^i, Start)$ query generates random number $w \in$ [1, n-1] and computes $C_u = w.P_{pub} \oplus h_2(GID_i \Vert m_i \Vert T_1), CID_i = h_2(w.P_{pub} \Vert T_1) \oplus RPW_i$ and $C_1 = h_2(CID_i \Vert m_i) \oplus RPW_i$ Return (CID_i, C_u, C_1, T_1)
For $Send(CID_i, C_u, C_1, T_1)$ query $w.P_{pub}$'$= h_2(GID_i \Vert m_i \Vert T_1) \oplus C_u$, $C_p = w.P_{pub}$'$\oplus h_2(Z_n \Vert m_j \Vert T_2)$ $CID_j = h_2(C_u \Vert C_p \Vert w.P_{pub}$'$)$ and $C_2 =$ $h(w.P_{pub}$'$\Vert C_1 \Vert CID_i \Vert CID_j \Vert T_1 \Vert T_2)$. Return $(CID_i, C_u, C_1, T_1, C_2, C_p, T_2)$
For $Send(CID_i, C_u, C_1, T_1, C_2, C_p, T_2)$ query $w.P_{pub}$'$= h_2(GID_i \Vert m_i \Vert T_1) \oplus C_u$, CID_j'$= h_2(C_u \Vert C_p \Vert w.P_{pub}$'$)$ C_2'$= h(w.P_{pub}$'$\Vert C_1 \Vert CID_i \Vert CID_j$'$\Vert T_1 \Vert T_2)$. If C_2'$= C_2$ P_i and PHC_i is authenticated Generates $y \in [1, n-1]$ Computes $C_k = y.P_{pub} \oplus h_3(m_j \Vert T_3)$, $SK_{gh} =$ $h_3(w.P_{pub}$'$\Vert y.P_{pub} \Vert m_i \Vert m_j)$ and $C_3 = h(SK_{gh} \Vert T_3 \Vert y.P_{pub})$ Return (C_3, C_k, T_3) Else Terminated
For $Send(C_3, C_k, T_3)$ query $y.P_{pub}$'$= h_3(m_j \Vert T_3) \oplus C_s$, $SK_{phc} = h_3(w.P_{pub}$'$\Vert y.P_{pub}$'$\Vert m_i \Vert m_j)$ and $C_4 = h(SK_{phc} \Vert C_3 \Vert T_3 \Vert w.P_{pub}$'$\Vert y.P_{pub}$'$)$ Return (C_3, C_k, T_3, C_4)
For $Send(C_3, C_k, T_3, C_4)$ query $y.P_{pub}$'$= h_3(m_j \Vert T_3) \oplus C_s$, $SK_p = h_3(w.P_{pub} \Vert y.P_{pub}$'$\Vert m_i \Vert m_j)$ and C_4'$= h(SK_p \Vert C_3 \Vert T_3 \Vert w.P_{pub} \Vert y.P_{pub}$'$)$ If C_4'$= C_4$ PHC_i and GH_i is authenticated Else Terminated

The simulation shows that the transcript distribution of the game is indistinguishable in the model. Hence we have

$$Pr[S_0] = Pr[S_1]$$

Game G_2 This game is to avoid the occurrence of collision in the transcript $(CID_i, C_u, C_1, T_1), (CID_i, C_u, C_1, T_1, C_2, C_p, T_2), (C_3, C_k, T_3), (C_3, C_k, T_3, C_4)$ and in the hash queries. In the proposed scheme, SC, w, and y are chosen randomly. According to the birthday paradox, the collision probability that occurred in the transcript's transmit is at most $(q_s+q_e)^2/2n$. Also, the probability of the occurrence of the collision in the output of the hash oracle is at most $(q_h)^2/2^{k+1}$. Hence we have

$$|Pr[S_2] - Pr[S_1]| \leq \frac{(q_s+q_e)^2}{2n} + \frac{(q_h)^2}{2^{k+1}} \quad (2)$$

Game G_3 In this game, the adversary could guess the authentication value C_3 and C_4 without making the hash query. Since the games G_3 and G_2 are indistinguishable unless government hospital server GH_i or patient P_i rejects a valid authentic value. Hence we have

$$|Pr[S_3] - Pr[S_2]| \leq \frac{(q_s)^2}{2^{k+1}} \quad (3)$$

Game G_4 In this game A compute the session key using a private oracle h_3'instead of h_3. Hence the session key SK

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL

C:\progra~1\SPAN\testsuite\results\blockchain_authenticati
on.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 4.39s
  visitedNodes: 512 nodes
  depth: 9 plies
```

FIGURE 2. OFMC result in DDMIA.

is independent of h_3, w, y, and P. The difference between G_3 and G_4 is negligible as long as the ECDH assumption holds because of the Diffie-Hellman problem's random self-reducibility. Hence we have

$$|Pr[S_4] - Pr[S_3]| = q_h.Adv_{EC}^{ECDH}(t + (q_{exe} + q_{send})T_{EC}) \tag{4}$$

C. RESULT OF FORMAL SECURITY VERIFICATION USING AVISPA TOOL

This section presents the results of the security verification using the AVISPA tool. The schemes in AVISPA can be Implemented by the HLPSL. In HLPSL, the Doley-Yao model [37] has been used to build the intruder. During the execution of schemes, the HLPSL code is converted into an Intermediate Format(IF) through hlpsl2if. Further, the backend reads the IF and analys the security goals. There are four backends are used in AVISPA used for security analysis known as On-the-fly Model-Checker (OFMC) [41], CL-AtSe (Constraint Logic-based Attack Searcher) [42], SAT-based Model checker [43] and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) [44]. If the protocol achieves all defined goals, then the output will be given as SAFE else the output will be UNSAFE.

The result of security verification using the AVISPA tool is presented in Figure 2 and Figure 3. The results are obtained through the back ends OFMC and CLAtSe as SAFE. The other two backends SATMC and TA4SP, do not support the XOR feature. Hence the results are received as "Inconclusive." From the obtained result, we can clearly say that the DDMIA scheme achieves all the specified goals and resists all active attacks.

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL

C:\progra~1\SPAN\testsuite\results\blockchain_authenticat
ion.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.28 seconds
  Computation: 0.00 seconds
```

FIGURE 3. ATSE result in DDMIA.

D. FORMAL ANALYSIS OF DDMIA USING BAN LOGIC

This section presents the formal analysis of DDMIA using BAN-logic proposed by Burrows *et al.* [33]. The analysis aims to prove the security of the scheme's session key shared between the P_i, PHC_i, and GH_i. Before beginning the analysis, we illustrate the notations and the logical postulates related to BAN-logic.

1) NOTATIONS IN THE BAN STATEMENTS [33]

This section presents the syntax and semantics of the BAN-logic necessary to prove the security goals. The logic model has several objects which are named as principals, encryption keys, and formulas. To better understand the notations, we represented U and S as principals, K is the shared key, SK is the shared session key between the principals, and X denotes the statements. The logical notations and its descriptions are given below:

$U \overset{SK}{\leftrightarrow} S$: U_i and S share the session key SK to communicate.

$U \mid\equiv X$: U believes the statement X and take as true.

$U \triangleleft X$: U receives statement X and is capable of reading it.

$U \mid\sim X$: U sents the statement X.

$U \Rightarrow X$: U is an authority on X and should be trusted on this matter.

$\#(X)$: Statement X is *fresh* i.e., X has not been sent first time while running the protocol.

$\overset{K}{\mapsto} P$: P has K as a public key.

$(X.P)$: P is a point on elliptic curve mulplied by the number X

$\{X, Y\}_K$ or $[X, Y]_K$: This represents that X, Y encrypted with key K.

$(X, Y)_K$: This represents that X, Y are exclusive-ORed with key K.
$\langle X, Y\rangle_K$: This represents X, Y are hashed with by K.

2) LOGICAL POSTULATES

In this section, we describe the postulates which are applied during the formal analysis.

1) *Message meaning rules:* This postulate presents the interpretation of communicated messages. This rule forms the beliefs about the origin of messages to the principal. According to the postulate, If principal U believes that the key K is shared with S and message containing X is encrypted under K, then U believes that S is capable of reading X. We can represent *message meaning rules:* as follows:

$$\frac{U \mid\equiv S \overset{K}{\leftrightarrow} U, U \triangleleft [X]_K}{U \mid\equiv S \mid\sim X}$$

2) *The nonce-verification:* This rule states that the message sent by the principal U/S is recent, and it is in the same session. Also, the sender believes in the freshness of the message. It can be represented as follows:

$$\frac{U \mid\equiv \#(X), U \mid\equiv S \mid\sim X}{U \mid\equiv S \mid\equiv X}$$

3) *The jurisdiction rule:* The rule states that if principal U believes that S has authority and trust over the statement X, then U also trusts S about the truth of statement X. The *jurisdiction rule:* is represented as follows:

$$\frac{U \mid\equiv S \Rightarrow X, U \mid\equiv S \mid\equiv X}{U \mid\equiv X}$$

4) *Fresh conjuncatenation rule* If principal U believes about the freshness of X, then it U also believes (X, Y) are fresh. This postulate can be represented as follows:

$$\frac{U \mid\equiv \#(X)}{U \mid\equiv \#(X, Y)}.$$

3) METHOD

There are three main steps involved in the analysis of DDMIA using BAN logic. The first step is to set the goals and assumptions necessary to prove the session security. It is in the form of formulas represented using the symbolic notations. The second step is to convert the communicated messages of the proposed into the formulas using symbolic notations called as idealized form. Finally, apply the logical postulates to the communicated message's idealized form. In the DDMIA analysis, we have taken P_i, PHC_i, and GH_i as principals, $H(x\|e)$ is the shared key, sk is the session key, and the communicated messages are the statements.

4) GOALS OF DDMIA

This section sets the goal (G) to achieve from the analysis. Our goal is to secure the communication session by protecting the session key (SK) between the communication participants. To prove this, P_i and GH_i should trust each other. Hence we set mainly two goals named as G_1 and G_2.

$Goal1{:}GH_i \mid\equiv GH_i \overset{sk}{\leftrightarrow} PHC_i$
$Goal2{:}P_i \mid\equiv P_i \overset{sk}{\leftrightarrow} PHC_i$

5) ASSUMPTIONS IN DDMIA

In BAN logic, assumptions assure the success of the protocol. The assumptions mainly state the initially shared keys, fresh nonce, and trusted principals. In DDMIA, there are six assumptions named from A_1 to A_6. In the given assumptions, A_1 and A_2 are the shared key 'x'. Assumptions A_3 and A_4 are session key 'sk' shared between P_i, PHC_i and GH_i. The generated timestamp and fresh nonce are presented in A_5 and A_6.

A_1: $P_i \mid\equiv P_i \overset{H(x\|e)}{\leftrightarrow} PHC_i$
A_2: $PHC_i \mid\equiv GH_i \overset{H(x\|e)}{\leftrightarrow} PHC_i$
A_3: $GH_i \mid\equiv PHC_i \Rightarrow PHC_i \overset{sk}{\leftrightarrow} GH_i$
A_4: $PHC_i \mid\equiv P_i \Rightarrow PHC_i \overset{sk}{\leftrightarrow} P_i$
A_5: $PHC_i \mid\equiv \#T_3$
A_6: $P_i \mid\equiv \#y$

6) COMMUNICATED MESSAGES

An analysis of DDMIA uses the BAN logic model to prove that the scheme mutually authenticates and shares a common session key between P_i, PHC_i, and GH_i. We use the communication messages sent and received between the principals. Since the BAN logic is used to verify the session key security, we used only mutual authentication messages, which are given below:

Message 1: $\{C_3, C_k, T_3\}$
Message 2: $\{C_4\}$

7) IDEALIZED FORM OF THE PROPOSED SCHEME

The scheme messages should be changed to the idealized forms to describe the BAN logic model. A message in the idealized protocol is a formula. Idealized represents which parameter shares key or nonce between the principals. Also, it includes the other parameters communicated between the principals. The Idealized form of DDMIA messages are given below:

C_3: $(GH_i \overset{sk}{\leftrightarrow} PHC_i, y.P_{pub}, T_3)$
C_k: $(GH_i \overset{H(e\|x)}{\leftrightarrow} PHC_i, y.P_{pub}, T_3)$
C_4: $(PHC_i \overset{sk}{\leftrightarrow} P_i, GH_i \overset{sk}{\leftrightarrow} PHC_i, w.P_{pub}, y.P_{pub}, T_3)$

8) SECURITY ANALYSIS PROOF

The formal analysis of DDMIA using BAN-logic is presented in this section. The study of the communicated message idealized form helps us to explain the proof. The detailed proof is shown below:

PHC_i receives Message 1, then we have

$$PHC_i \triangleleft \{(GH_i \overset{sk}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), (GH_i \overset{H(e\|x)}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), T_3\} \quad (5)$$

From jurisdiction rule we can prove that

$$PHC_i \triangleleft \{(GH_i \overset{sk}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), (GH_i \overset{H(e\|x)}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), T_3\} \quad (6)$$

According to $Assumption A_2$ and equation (6) apply message meaning rule and we get

$$PHC_i| \equiv GH_i \,|\sim\{(GH_i \overset{sk}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), (GH_i \overset{H(e\|x)}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), T_3\} \quad (7)$$

According to A_4 and (7) apply the freshness rule and we get

$$PHC_i| \equiv \#\{(GH_i \overset{sk}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), (GH_i \overset{H(e\|x)}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), T_3\} \quad (8)$$

According to (7) and (8) We apply nonce verification rule and we get

$$PHC_i| \equiv GH_i| \equiv \{(GH_i \overset{sk}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), (GH_i \overset{H(e\|x)}{\leftrightarrow} PHC_i, y.P_{pub}, T_3), T_3\} \quad (9)$$

We can also write equation (8) as

$$PHC_i| \equiv GH_i| \equiv GH_i \overset{sk}{\leftrightarrow} PHC_i \quad (10)$$

According to Assumption A_3 and (10) apply jurisdiction rule we get

$$PHC_i| \equiv GH_i \overset{sk}{\leftrightarrow} PHC_i$$

Which satisfies the Goal 1.
P_i receives Message 2, then we have

$$P_i \triangleleft \{(PHC_i \overset{sk}{\leftrightarrow} P_i, GH_i \overset{sk}{\leftrightarrow} PHC_i, w.P_{pub}, y.P_{pub}, T_3)\} \quad (11)$$

From jurisdiction rule we can prove that

$$P_i \triangleleft \{(PHC_i \overset{sk}{\leftrightarrow} P_i, GH_i \overset{sk}{\leftrightarrow} PHC_i, w.P_{pub}, y.P_{pub}, T_3)\} \quad (12)$$

According to $Assumption A_1$ and equation (12) apply message meaning rule and we get

$$P_i| \equiv PHC_i \,|\sim\{(PHC_i \overset{sk}{\leftrightarrow} P_i, GH_i \overset{sk}{\leftrightarrow} PHC_i, w.P_{pub}, y.P_{pub}, T_3)\} \quad (13)$$

According to A_6 and (13) apply the freshness rule and we get

$$PHC_i| \equiv \#\{(PHC_i \overset{sk}{\leftrightarrow} P_i, GH_i \overset{sk}{\leftrightarrow} PHC_i, w.P_{pub}, y.P_{pub}, T_3)\} \quad (14)$$

According to (13) and (14) We apply nonce verification rule and we get

$$P_i| \equiv PHC_i| \equiv \{(PHC_i \overset{sk}{\leftrightarrow} P_i, GH_i \overset{sk}{\leftrightarrow} PHC_i, w.P_{pub}, y.P_{pub}, T_3)\} \quad (15)$$

We can also write equation (14) as

$$P_i| \equiv PHC_i| \equiv PHC_i \overset{sk}{\leftrightarrow} P_i \quad (16)$$

According to Assumption A_4 and (16) apply jurisdiction rule we get

$$P_i \,| \equiv P_i \overset{sk}{\leftrightarrow} PHC_i$$

Which satisfies the Goal 2.

From the proof the goals G_1 and G_2 are achieved. Hence, we conclude that the principals P_i, PHC_i, and GH_i believes that session key SK is shared securely.

IV. PERFORMANCE ANALYSIS OF DDMIA SCHEME

This section focuses on the performance analysis of the DDMIA scheme. The analysis mainly focuses on calculating the computation and communication costs and comparing the result with related authentication schemes. For the analysis, the DDMIA scheme has been compared with multi party authentication schemes which are Odelu *et al.* [24], Park and Park [25], Amin *et al.* [26], Qi *et al.* [27]. The remaining Irshad *et al.* [45], Chaudhary *et al.* [46], Xu *et al.* [47], and Lei and Chuang [48] schemes are traditional two parties authentication schemes.

A. COMPUTATION AND COMMUNICATION COSTS ANALYSIS

The computation cost of the DDMIA has been compared with the other schemes and presented in Table 7. This analysis has been made considering the schemes total computation cost and expected execution tiome. To measure the expected execution time, we implemented the DDMIA using the GO language in the environment of Ubuntu 18.04.4 LTS 64-bit PC, Intel Core-i5 6200U CPU of 2.80 GHz, 4GB RAM, and Intel®HD Graphics 520. In the implementation, 'crypto/sha256'was used to perform the hash operations, 'crypto/elliptic'was used for the elliptic curve cryptographic operations, and ' crypto/rand' was employed for generating the random numbers.

In Table 7, computational parameters are defined as follows: T_{mul} - Execution time of elliptic curve scalar multiplication, T_h - Execution time of one-way hash function, T_{sym} - Execution time of one symmetric encryption/decryption function, T_{kdf} - the time for performing one-way key derivation function, T_c - The time for executing the Chaotic polynomial mapping. According to the simulation result, the required execution time for one T_h operation is 0.008 ms (millisecond), and one T_{mul} operation requires 0.101 ms. The execution time for T_{sym}, T_{kdf}, T_c is taken from [24], [27], which is 0.0046ms, 0.008ms, 0.02104, respectively.

The computation cost of DDMIA is the sum of the costs of P_i; PHC_i; and GH_i. The cost has been computed based on the number of operations performed during the authentication phase. We are not considering the cost of xor and

TABLE 7. Computation and communication cost analysis.

Schemes	Patient/ User	PHC/RC	GH/ Server	Total Computational Cost	Estimated Time	Communication cost
Odelu et al. [24]*	$1T_{sym} + 3T_{mul} + 7T_h$	$2T_{sym} + T_{mul} + 10T_h$	$3T_{sym} + 2T_{mul} + 6T_h$	$6T_{sym} + 6T_{mul} + 23T_h$	$0.817ms$	$2944bits$
Park-Park [25]*	$2T_{mul} + 10T_h$	$11T_h$	$3T_{mul} + 4T_h$	$5T_{mul} + 25T_h$	$0.705ms$	$3360bits$
Amin et al. [26]*	$9T_h$	$7T_h$	$6T_h$	$22T_h$	$0.176ms$	$2980bits$
Qi et al. [27]*	$3T_{mul} + 6T_h$	$1T_{sym} + 1T_{kdf} + T_{mul} + 5T_h$	$1T_{sym} + 1T_{kdf} + 4T_{mul} + 4T_h$	$2T_{sym} + 2T_{kdf} + 8T_{mul} + 15T_h$	$0.946ms$	$2846bits$
Irshad et al. [45]	$4T_c + 7T_h$	-	$4T_c + 4T_h$	$8T_c + 11T_h$	$0.256ms$	$1088bits$
Chaudhary et al. [46]	$1T_{sym} + 15T_h$	-	$1T_{sym} + 12T_h$	$2T_{sym} + 27T_h$	$0.225ms$	$1344bits$
Xu et al. [47]	$3T_{mul} + 10T_h$	-	$3T_{mul} + 6T_h$	$6T_{mul} + 16T_h$	$0.734ms$	$1696bits$
Lei et al. [48]	$4T_{mul} + 6T_h$	-	$3T_{mul} + 4T_h$	$7T_{mul} + 10T_h$	$0.787ms$	$1600bits$
DDMIA Scheme*	$2T_{mul} + 14T_h$	$11T_h$	$1T_{mul} + 10T_h$	$3T_{mul} + 35T_h$	$0.583ms$	$3360bits$

* multi-party authentication schemes

concatenation operations since the execution time is negligible. P_i needs two T_{mul} operations and 14 hash operations to compute parameters therefore the computation cost of P_i is $2T_{mul} + 14T_h$. Similarly PHC_i and GH_i needs $12T_h$, and $1T_{mul} + 10T_h$, respectively to compute the communication parameters. Hence, the total computation cost of the proposed DDMIA scheme is $3T_{mul} + 35T_h$.

On comparison of the DDMIA authentication scheme with multiparty schemes, we observe that the computation cost of DDMIA is better than all, except for the Amin *et al.* scheme [26]. This difference is because Amin *et al.* scheme is proposed using hash functions only whereas DDMIA uses elliptic curve cryptography. According to Wang and Wang [49] the usage of hash function only in the scheme, will result in loss of user anonymity and public-key techniques should be used instead. On comparing the proposed DDMIA scheme with other traditional schemes presented in Table 7, we observed that the computation cost of DDMIA is lesser than Xu *et al.* and Lei *et al.* schemes and slightly higher than Irshad *et al.* and Chaudhary *et al.* schemes. However, these traditional schemes having architectural limitations wherein the patient directly communicates with the hospital. The estimated execution time of the DDMIA scheme is (3 * 0.101 ms) + (35 * 0.008 ms) = 0.583ms. Similarly, on comparing the results of DDMIA with the other multi-party authentication schemes (Table 7), we observed that the estimated execution of the DDMIA is slightly higher than Amin *et al.* [26] scheme but lesser than all other multi-party schemes.

The communication cost was also analysed against the schemes listed in table 7. The communication cost calculation includes the estimated cost of the communication parameters in the login and authentication phase of one complete session. For consistency purpose, we assume that the length of the identity $ID_i/PID_i/HID_i$ is 32 bits, the output size of hash function $h_1(.)$, $h_2(.)$, and $h_3(.)$ is 160 bits, size of an elliptic curve point is 320 bits, the block size of symmetric encryption/decryption is 128 bits, and a random number/Timestamp is 128 bits. The login phase, and authentication and key agreement phase, DDMIA requires a total of 320 + 160 + 160 + 128 = 768 bits, 768 + 160 + 160 + 128 = 1216 bits, 160 + 320 + 128 = 608 bits, and 608 + 160 = 768 bits, for the messages $M_1 = \{C_u, CID_i, C_1, T_1\}$, $M_2 = \{C_p, C_2, T_2\}$, $M_3 = \{C_3, C_k, T_3\}$ and $M_4 = \{C_4\}$. Hence, the total communication cost required to achieve the one session of DDMIA is 3360 bits.

TABLE 8. Functional analysis.

Schemes	F1	F2	F3	F4	F5	F6	F7	F8
Odelu et al. [24]*	✓	✓	✓	✓	✓	✓	✓	×
Park-Park [25]*	✓	✓	✓	✓	✓	×	×	✓
Amin et al. [26]*	✓	✓	×	✓	✓	✓	✓	✓
Qi et al. [27]*	✓	✓	✓	✓	✓	✓	×	✓
Irshad et al. [45]	✓	×	×	✓	✓	✓	×	✓
Chaudhary et al. [46]	✓	✓	×	✓	✓	✓	×	×
Xu et al. [47]	✓	✓	✓	✓	✓	✓	✓	✓
Lei et al. [48]	✓	✓	×	✓	✓	×	×	✓
DDNIA Scheme*	✓	✓	✓	✓	✓	✓	✓	✓

* multi-party authentication schemes

Compared to the other schemes presented in Table 7, the overall communication cost of DDMIA is equal to the Park and Park [25] scheme and higher than all other schemes. But, it is still acceptable because the DDMIA scheme is a distributed and dynamic, multi-party authentication scheme where the participants can mutually authenticate without depending upon the registration center. In DDMIA, a registration center is essential only for the registration of healthcare providers. While other multi-party authentication schemes completely depend upon a third party to perform the authentication. Also, DDMIA meets the mentioned security criteria whereas the other schemes are vulnerable to several attacks which are presented in the next section. Hence, we claim that the DDMIA scheme is still efficient and robust in the blockchain based distributed, multi-party architecture.

B. FUNCTIONAL ANALYSIS

Table 8 presents the feature-wise functional analysis of the DDMIA scheme with other schemes. The functionalities represented in table 8 are as follows: F1 - Mutual authentication, F2 - Quick credentials verification, F3 - Patient anonymity, F4 - Secret key management, F5 - Session key agreement, F6 - Perfect forward secrecy, F7 - Resilience against insider attack, F8 - Prevention of a replay attack. The functions considered for comparision are based on the security requirements mentioned in section I A.

From Table 8, it is clear that two schemes, [47] and the proposed DDMIA scheme meet all the functional requirements. But, the [47] scheme may not be suitable for the implemention in a blockchain network. In addition, the DDMIA achieves distributed multi-party authentication where one health provider can authenticate the patient through another provider. Multiparty schemes like [24] authentication scheme uses elliptic curve cryptography, but the scheme is vulnerable to replay attack. Reference [25] authentication scheme is a three-factor user authentication that uses the elliptic curve cryptosystem. However, the scheme does not support forward secrecy and is vulnerable to insider attack. Reference [26] authentication protocol uses multiple registration servers to manage users and does not provide patient anonymity. Registration in multiple servers may result in identity collision and affect key management. Reference [27] scheme is based on biometrics authentication for multi-server TMIS, But the scheme is vulnerable to insider attack. Therefore, the authors conclude that the DDMIA scheme has all the required security features which makes it the most robust.

The proposed multi-party DDMIA scheme ensures that the authentication process is scalable in comparison with a centralized approach. DDMIA decreases the dependency on registration centers for authentication, a one-time registration is done by health care providers in the blockchain network. During the referral itself, authentication is de-centralized, it will be done on the server of the hospital to which the patient has been referred. In case of multiple simultaneous authentication requests on the same hospital server, the requests will be queued and the running time and latency are expected to increase linearly.

V. CONCLUSION

The proposed DDMIA authentication scheme was designed keeping in mind the security requirements of a decentralized, multiparty authentication scheme for blockchain-based networks. The patient registers with a e-health provider ‘A’ and can initiate a consultation. The e-health provider ‘ A’ can refer the patient to another e-health provider ‘B’ for appropriate treatment. The patient is not required to register with ‘B’ and ‘A’ dynamically and mutually authenticates the patient with ‘B’. The security requirements of a blockchain-based e-health network were defined, the DDMIA scheme was proven in theory and with cryptanalysis. The formal security verification was done using the AVISPA tool and BAN Logic proved that the session is secure. In terms of computation cost, the DDMIA scheme is more efficient than all other multi-party schemes. The distributed nature of the multiparty authentication is achieved by using more communication bits, however, this cost will not affect the network. The DDMIA scheme has an execution time better than the other multi-party schemes and its distributed nature will ensure scalability. The overall communication cost of DDMIA is higher than all other schemes but it is acceptable because it is a distributed, dynamic and multi-party authentication scheme independent of the registration center. Functional analysis proves that in comparison with other multi-party schemes, the DDMIA is the most robust among them because it achieves all the specified security requirements. Furthermore, the DDMIA scheme is a plug-in over an existing blockchain technology that performs authentication separately from the primary blockchain based transaction system, it is not affected by the writing of transactions and block generation. Hence DDMIA authentication is distributed among healthcare providers and it can scale and perform authentication for simultaneous requests efficiently.

In most countries, e-health networks consist of many healthcare providers, and the patient interacts with different providers for every health problem. Each health care provider generates data about the patient and records the same in their own isolated repositories. In the absence of nationwide identifiers, it is challenging to integrate the health data or medical history of a patient into a comprehensive EHR. The features of blockchain-based technologies make them ideal for adoption in e-health environments. The network could be a country-wide network of health providers connected in a permissioned private network. Authentication could become cumbersome because the patient must register with multiple health care providers. This article presents a novel, robust, distributed multi-party authentication scheme for referrals in blockchain-based e-health networks. The DDMIA scheme ensures that healthcare providers can mutually authenticate a patient without registrations faster and securely. In the future, the proposed scheme can be improved in terms of communication cost reduction, which will enhance the throughput and decrease latency. Since the DDMIA scheme has a plug-in architecture, it can also be developed as a cloud-based ‘Authentication-as-a-service.’ The registration center can also be replaced with a novel consensus algorithm, ensuring a fully decentralized blockchain network. In the future, artificial intelligence-based approaches and intelligent blockchain technologies could play a game-changing role in blockchain-based authentication schemes.

APPENDIX

The registration phase, login and authentication phase of DDMIA has been implemented among the four roles named as Patient (Pi), primary Health center (PHCi), and govhosp (GHj) and regcen(RCi). The role of each participant is presented in Figure 5, Figure 6, Figure 7, and Figure 8. The patient role of the proposed scheme is shown in Figure 5. Here the scheme begins by receiving a start signal. There are three symmetric keys SKpiphc, SKphcgh, and SKghpi are used to communicate the messages between the participants. Snd() and RCV() functions are the channels created for message communication. Similarly, hospital, govhosp and *RC* roles are implemented and presented in Figures 6, 7, and 8.

Figure 8 and Figure 9 present the session and environment roles. The session role includes the primary roles for composition and the channels of all roles involved

```
role patient (Pi, PHCi, GHj, RCi : agent,
      SKpiphc, SKphcgh, SKghpi : symmetric_key,
      Snd,RCV :channel(dy))

played_by Pi
def=
 local State : nat,
     Ei, Ej, Mi, Mj, Ri, GIDi, Vi,Zn,Zm,Hn,Hm, X, Xi:text,
     IDi, PIDi, HIDi, PWi, RPWi, Bi, Bj,Ai, Aj, Cu, Cp, Ck, CIDi, CIDj:text,
     C1, C2, C3, C4, W, Y, T1, T2, T3, SKphc, SKghc, SKp, Qi, Qs: text,
     Ec, H1, H2, H3 : hash_func
     const subs1, subs2, subs3, pi_ghj_w, ghj_pi_y, pi_ghj_T1, ghj_pi_T2 : protocol_id

init State:=0
          transition
1.        State=0 /\ RCV(start)=|>
          State':=1 /\ RPWi' := H1(IDi'.PWi')
                /\ Snd({IDi'.RPWi'}_SKpiphc)

          /\ secret({IDi, PWi}, subs1, {PHCi, GHj, RCi})
 2. State = 1 /\RCV({Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi'))))}_SKpiphc) =|>

 State':= 2
                /\RPWi' := H1(IDi'.PWi')
                /\ Ri':= xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi')))
                /\ Vi':= H1(Ri'.IDi'.H1(IDi'.PWi'))
                /\ GIDi':= Ec(H1(Vi'))
                /\W' := new()
                /\T1' := new()
                /\Qi':= Ec(W')
                /\Cu' := xor(Ec(W'), H2(GIDi'.Ec(H1(X.Ei')).T1'))
                /\CIDi' := xor(H2(Ec(W').T1'), RPWi')
                /\C1' := H2(CIDi'.H1(X.Ei').Ec(W'))
                /\ Snd ({Cu'.CIDi'.C1'.T1'}_SKpiphc)
                /\ secret({W}, subs3, {Pi, PHCi, GHj})
                /\witness(Pi, GHj, pi_ghj_w, W')
                /\witness(Pi, GHj, pi_ghj_T1, T1')
3. State = 2 /\ RCV
({H3(H3(Ec(W').Ec(Y').H1(X'.Ej').H1(X.Ei')).T3.Ec(Y')).xor(Ec(Y'),H3(H1(X'.Ei').T3)).T3'.H3(H3(xor(H2(Ec(H1(H1(xor(IDi'
, H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).H1(X'.Ej').T1),xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1'))).xor(H3(H1(X'.Ei').T3),xor(H1(X'.Ei'),H3(H1(X'.Ei').T
3))).H1(X'.Ej').H1(X.Ei')).H3(H3(xor(H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).H1(X'.Ej').
T1),xor(Ec(W'), H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1'))).
xor(H3(H1(X'.Ei').T3),xor(H1(X'.Ei'),H3(H1(X'.Ei').T3))).H1(X'.Ej').H1(X.Ei')).T3'.Ec(Y')).T3'.xor(H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).H1(X'.Ej').T1),xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1'))).Ec(Y'))}_SKpiphc) =|>
 State' := 3 /\Qs' := xor(H3(H1(X'.Ei').T3),xor(H1(X'.Ei'),H3(H1(X'.Ei').T3)))
          /\SKp' := H3(Qi'.Qs'.H1(X'.Ej').H1(X.Ei'))
          /\C4' := H3(SKp'.H3(SKp.T3.Qs').T3.Qi'.Qs')
          /\request(Pi, GHj, ghj_pi_y, Y')
          /\request(Pi, GHj, ghj_pi_T2, T2')
      end role
```

FIGURE 4. Role specification for the P_i in DDMIA.

```
role healthcare (Pi, PHCi, GHj, RCi : agent,
      SKpiphc, SKphcgh, SKghpi : symmetric_key,
      Snd,RCV :channel(dy))
played_by PHCi
def=
 local State : nat,
     Ei, Ej, Mi, Mj, Ri, GIDi, Vi,Zn,Zm,Hn,Hm, X, Xi:text,
     IDi, PIDi, HIDi, PWi, RPWi, Bi, Bj,Ai, Aj, Cu, Cp, Ck, CIDi, CIDj:text,
     C1, C2, C3, C4, W, Y, T1, T2, T3, SKphc, SKghc, SKp, Qi, Qs: text,
     Ec, H1, H2, H3 : hash_func
     const subs2 : protocol_id
init State := 0
     transition
1.        State=0 /\ RCV(start)=|>
          State':=1 /\ Bj':=new()
                /\ Ai' := H1(PIDi'.Bj')
                /\ Snd({PIDi'.Ai'}_SKphcgh)
2.   State=1 /\ RCV({IDi'.H1(IDi'.PWi')}_SKphcgh)=|>
          State':=2 /\ Ri':= xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi')))
          /\ Vi':= H1(Ri'.IDi'.H1(IDi'.PWi'))
          /\ GIDi':= Ec(H1(Vi'))
          /\ Snd({GIDi'}_SKphcgh)
          /\secret({IDi, RPWi}, subs2, {Pi, PHCi, GHj, RCi})
3.   State=2 /\ RCV({xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')).xor(H2(Ec(W').T1'), H1(IDi'.PWi')).H2(xor(H2(Ec(W').
T1'),H1(X.Ei')).H1(X.Ei').Ec(W')).T1'}_SKphcgh)=|>
          State':=3 /\Qi' := xor(H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).
H1(X'.Ej').T1),xor(Ec(W'), H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')))
                    /\T2' := new()
                    /\Cp' := xor(xor(Ec(W'), H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).
Ec(H1(X.Ei')).T1')),H2(H1(H1(X'.Ej').H1(PIDi'.Bj')).H1(X'.Ej').T2'))
                    /\CIDj' := H2(xor(Ec(W'),H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).
Ec(H1(X.Ei')).T1')).Cp'.Qi')
                    /\C2' := H2(Qi'.H2(xor(H2(Ec(W').T1'), H1(IDi'.PWi')).H1(X.Ei').Ec(W')).xor(H2(Ec(W').T1'),
H1(IDi'.PWi')).CIDj'.T1'.T2')

                    /\ Snd({xor(Ec(W'), H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).
Ec(H1(X.Ei')).T1')).xor(H2(Ec(W').T1'), H1(IDi'.PWi')).H2(xor(H2(Ec(W').T1'), H1(X.Ei')).H1(X.Ei').Ec(W')).T1'.
Cp'.C2'.T2}_SKphcgh)
                    /\ secret({PIDi, Qi}, subs2, {Pi, PHCi, GHj, RCi})
4.        State=3 /\ RCV({H3(H3(Ec(W').Ec(Y').H1(X'.Ej').H1(X.Ei')).T3.Ec(Y')).xor(Ec(Y'),H3(H1(X'.Ei').T3)).
T3'}_SKphcgh)=|>
          State':=4 /\Qs' := xor(H3(H1(X'.Ei').T3),xor(H1(X'.Ei'),H3(H1(X'.Ei').T3)))
                    /\SKphc':= H3(xor(H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).H1(X'.Ej').T1),
xor(Ec(W'), H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1'))).
xor(H3(H1(X'.Ei').T3),xor(H1(X'.Ei'),H3(H1(X'.Ei').T3))).H1(X'.Ej').H1(X.Ei'))
                    /\C4':= H3(SKphc'.H3(SKphc'.T3'.Qs').T3'.xor(H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).
IDi'.H1(IDi'.PWi')))).H1(X'.Ej').T1),xor(Ec(W'), H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).
Ec(H1(X.Ei')).T1'))).Qs')
               /\ Snd({H3(H3(Ec(W').Ec(Y').H1(X'.Ej').H1(X.Ei')).T3.Ec(Y')).xor(Ec(Y'),H3(H1(X'.Ei').T3)).T3'.C4'}_SKphcgh)

end role
```

FIGURE 5. Role specification for the PHC_i in DDMIA.

```
role govhosp (Pi, PHCi, GHj, RCi : agent,
      SKpiphc, SKphcgh, SKghpi : symmetric_key,
      Snd,RCV :channel(dy))
played_by GHj
def=
 local State : nat,
     Ei, Ej, Mi, Mj, Ri, GIDi, Vi,Zn,Zm,Hn,Hm, X, Xi:text,
     IDi, PIDi, HIDi, PWi, RPWi, Bi, Bj,Ai, Aj, Cu, Cp, Ck, CIDi, CIDj:text,
     C1, C2, C3, C4, W, Y, T1, T2, T3, SKphc, SKghc, SKp, Qi, Qs: text,
     Ec, H1, H2, H3 : hash_func
     const subs4, subs5, pi_ghj_w, ghj_pi_y, pi_ghj_T1, ghj_pi_T2 : protocol_id
init State:=0
          transition
1.        State=0 /\ RCV(start)=|>
          State':=1 /\ Bi':=new()
                /\ Aj' := H1(HIDi'.Bi')
                /\ Snd({HIDi'.Aj'}_SKghpi)
 2.       State = 1 /\RCV({H1(H1(X'.Ei').H1(HIDi'.Bi'))}_SKghpi) =|>
                State':= 2
                     /\ secret({HIDi', Bi'}, subs4, {Pi, PHCi, RCi})

 3.       State=2 /\ RCV({xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')).xor(H2(Ec(W').T1'),
H1(IDi'.PWi')).H2(xor(H2(Ec(W').T1'), H1(X.Ei')).H1(X.Ei').Ec(W')).T1'.xor(xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')),H2(H1(H1(X'.Ej').H1(PIDi'.Bj')).H1(X'.Ej').T2')).H2(xor(H2
(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).H1(X'.Ej').T1),xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1'))).H2(xor(H2(Ec(W').T1'),
H1(IDi'.PWi')).H1(X.Ei').Ec(W')).xor(H2(Ec(W').T1'), H1(IDi'.PWi')).H2(xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')).xor(xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')),H2(H1(H1(X'.Ej').H1(PIDi'.Bj')).H1(X'.Ej').T2')).xor(H2(Ec
(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).H1(X'.Ej').T1),xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')))).T1'.T2').T2}_SKghpi)=|>
          State':=3
          /\T3' := new()
          /\Qi' := xor(H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).H1(X'.Ej').T1),xor(Ec(W'),
H2(Ec(H1(H1(xor(IDi', H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')))
          /\CIDj' := H2(xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')).xor(xor(Ec(W'), H2(Ec(H1(H1(xor(IDi',
H1(H1(X'.Ej').H1(IDi'.PWi'))).IDi'.H1(IDi'.PWi')))).Ec(H1(X.Ei')).T1')),H2(H1(H1(X'.Ej').H1(PIDi'.Bj')).H1(X'.Ej').T2')).Qi')
          /\C2':= H2(Qi'.H2(xor(H2(Ec(W').T1'), H1(IDi'.PWi')).H1(X.Ei').Ec(W')).xor(H2(Ec(W').T1'),
H1(IDi'.PWi')).CIDj'.T1'.T2')
%Starts mutual authentication.
          /\Y' := new()
          /\Qs':= Ec(Y')
          /\Ck' := xor(Qs',H3(H1(X'.Ei').T3))
          /\SKghc' := H3(Qi'.Qs'.H1(X'.Ej').H1(X.Ei'))
          /\C3' := H3(SKghc'.T3.Qs')
          /\ Snd ({C3'.Ck'.T3'}_SKghpi)
          /\ secret({Y}, subs5, {Pi, PHCi, GHj})
          /\witness(Pi, GHj, ghj_pi_y, Y')
          /\witness(Pi, GHj, ghj_pi_T2, T2')
          /\request(Pi, GHj, pi_ghj_w, W')
          /\request(Pi, GHj, pi_ghj_T1, T1')
      end role
```

FIGURE 6. Role specification for the GH_i in DDMIA.

```
role regcen (Pi, PHCi, GHj, RCi : agent,
      SKpiphc, SKphcgh, SKghpi : symmetric_key,
      Snd,RCV :channel(dy))
played_by RCi
def=
 local State : nat,
     Ei, Ej, Mi, Mj, Ri, GIDi, Vi,Zn,Zm,Hn,Hm, X, Xi:text,
     IDi, PIDi, HIDi, PWi, RPWi, Bi, Bj,Ai, Aj, Cu, Cp, Ck, CIDi, CIDj:text,
     C1, C2, C3, C4, W, Y, T1, T2, T3, SKphc, SKghc, SKp, Qi, Qs: text,
     Ec, H1, H2, H3 : hash_func

init State := 0
     transition
% Registration phase
1.        State=0 /\RCV({PIDi'.H1(PIDi'.Bj')}_SKpiphc) =|>
          State':=1 /\Ej' := new()
                    /\Mi' := H1(X'.Ej')
                    /\Zn' := H1(Mj'.H1(PIDi'.Bj'))
                    /\Hn' := H1(Zn'.PIDi)
                    /\ Snd({Zn'}_SKpiphc)

2.        State=1 /\RCV({HIDi'.H1(HIDi'.Bi')}_SKphcgh) =|>
          State':=2 /\Ei' := new()
                    /\Mj' := H1(X'.Ei')
                    /\Zm' := H1(Mi'.H1(HIDi'.Bi'))
                    /\Hm' := H1(Zm'.HIDi)
                    /\ Snd({Zm'}_SKphcgh)

end role
```

FIGURE 7. Role specification for the RC in DDMIA.

in communication. The environment role specifies the global constants and sessions for an adversary to play a legitimate role. It also defines the goals of DDMIA.

```
role session(Pi, PHCi, GHj, RCi : agent,
        SKpiphc, SKphcgh, SKghpi : symmetric_key)
def=
local Send1, Send2, Send3, Recv1, Recv2, Recv3:
channel (dy)
composition
        patient(Pi, PHCi, GHj, RCi, SKpiphc, SKphcgh,
SKghpi, Send1, Recv1)
        /\healthcare(Pi, PHCi, GHj, RCi, SKpiphc,
SKphcgh, SKghpi, Send2, Recv2)
        /\govhosp(Pi, PHCi, GHj, RCi, SKpiphc, SKphcgh,
SKghpi, Send3, Recv3)
        %/\regcen(Pi, PHCi, GHj, RCi, SKpiphc, SKphcgh,
SKghpi, Send3, Recv3)
end role
```

FIGURE 8. Role specification for the session in DDMIA.

```
role environment()
def=
const pi, phci, ghj, rci: agent,
skpiphc, skphcgh, skghpi : symmetric_key,
ei, ej, mi, mj, ri, gidi, vi,zn,zm,hn,hm, x, xi:text,
idi, pidi, hidi, pwi, rpwi, bi, bj,ai,ak, cu, cp, ck, cidi, cidj:text,
c1, c2, c3, c4, w, y, t1, t2, t3, skphc, skghc, skp, qi, qs: text,
ec, h1, h2, h3 : hash_func,
subs1, subs2, subs3, subs4, subs5, pi_ghj_w, ghj_pi_y,
pi_ghj_T1, ghj_pi_T2: protocol_id

intruder_knowledge = {pi, phci, ghj, rci, ec, h1, h2, h3, c1, c2, c3,
c4, t1, t2, t3, cidi, cidj}
composition
session(pi, phci, ghj, rci, skpiphc, skphcgh, skghpi)
/\session(pi, phci, ghj, rci, skpiphc, skphcgh, skghpi)
/\session(pi, phci, ghj, rci, skpiphc, skphcgh, skghpi)
end role
goal
secrecy_of subs1, subs2, subs3, subs4, subs5
authentication_on pi_ghj_w, ghj_pi_y, pi_ghj_T1, ghj_pi_T2
end goal
environment()
```

FIGURE 9. Role specification for the environment in DDMIA.

REFERENCES

[1] R. Mabiyan. (2020). *Knuth: Computers and Typesetting*. [Online]. Available: https://bit.ly/3Ktav9g

[2] S. K. Srivastava, "Adoption of electronic health records: A roadmap for India," *Healthcare Inform. Res.*, vol. 22, no. 4, pp. 261–269, 2016.

[3] B. K. Mohanta, D. Jena, S. S. Panda, and S. Sobhanayak, "Blockchain technology: A survey on applications and security privacy challenges," *Internet Things*, vol. 8, Dec. 2019, Art. no. 100107.

[4] G. Drosatos and E. Kaldoudi, "Blockchain applications in the biomedical domain: A scoping review," *Comput. Struct. Biotechnol. J.*, vol. 17, pp. 229–240, Jan. 2019.

[5] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 135, pp. 62–75, Jun. 2019.

[6] W. J. Gordon and C. Catalini, "Blockchain technology for healthcare: Facilitating the transition to patient-driven interoperability," *Comput. Struct. Biotechnol. J.*, vol. 16, pp. 224–230, Jan. 2018.

[7] J. A. Alzubi, "Blockchain-based Lamport Merkle digital signature: Authentication tool in IoT healthcare," *Comput. Commun.*, vol. 170, pp. 200–208, Mar. 2021.

[8] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, Aug. 2016, pp. 25–30.

[9] NHP I. (2016). *Electronic Health Record Standards of India, 2016*. [Online]. Available: http://nhp.org.in/en/e-health-intiatives/1146-ehr-standards

[10] K. Makkithaya, V. G. Kamath, and R. Cordeiro, "A minimal e-referral for meaningful share of maternal health information in public health scenarios," *Int. J. Electron. Healthcare*, vol. 8, nos. 2–4, pp. 142–162, 2015.

[11] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, and S. Liu, "Blockchain-based data preservation system for medical data," *J. Med. Syst.*, vol. 42, no. 8, p. 141, Aug. 2018.

[12] L. Zhou, L. Wang, and Y. Sun, "MIStore: A blockchain-based medical insurance storage system," *J. Med. Syst.*, vol. 42, no. 8, p. 149, 2018.

[13] *Health Informatics—Security and Privacy Requirements of EHR Systems for Use in Conformity Assessment*, Standard ISO/TS14441:2013, 2020. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:ts:14441:ed-1:v1:en

[14] 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.

[15] 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.

[16] P. Mohit, R. Amin, A. Karati, G. 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, 2017.

[17] Q. Cheng, X. Zhang, and J. Ma, "ICASME: An improved cloud-based authentication scheme for medical environment," *J. Med. Syst.*, vol. 41, no. 3, p. 44, Mar. 2017.

[18] 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.

[19] S. Wang, S. Zhu, and Y. Zhang, "Blockchain-based mutual authentication security protocol for distributed RFID systems," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 74–77.

[20] M. Conti, M. Hassan, and C. Lal, "BlockAuth: BlockChain based distributed producer authentication in ICN," *Comput. Netw.*, vol. 164, Dec. 2019, Art. no. 106888.

[21] J. Wang, L. Wu, K.-K.-R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1984–1992, Mar. 2020.

[22] X. Liu, W. Ma, and H. Cao, "MBPA: A MediBchain-based privacy-preserving mutual authentication in TMIS for mobile medical cloud architecture," *IEEE Access*, vol. 7, pp. 149282–149298, 2019.

[23] U. Khalid, M. Asim, T. Baker, P. C. K. Hung, M. A. Tariq, and L. Rafferty, "A decentralized lightweight blockchain-based authentication mechanism for IoT systems," *Cluster Comput.*, vol. 23, no. 3, pp. 2067–2087, Sep. 2020.

[24] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[25] 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, 2016.

[26] R. Amin, S. H. Islam, M. S. Obaidat, G. P. Biswas, and K.-F. Hsiao, "An anonymous and robust multi-server authentication protocol using multiple registration servers," *Int. J. Commun. Syst.*, vol. 30, no. 18, p. e3457, Dec. 2017.

[27] M. Qi, J. Chen, and Y. Chen, "A secure biometrics-based authentication key exchange protocol for multi-server TMIS using ECC," *Comput. Methods Programs Biomed.*, vol. 164, pp. 101–109, Oct. 2018.

[28] X. Xiang, M. Wang, and W. Fan, "A permissioned blockchain-based identity management and user authentication scheme for e-health systems," *IEEE Access*, vol. 8, pp. 171771–171783, 2020.

[29] C.-T. Li, D.-H. Shih, C.-C. Wang, C.-L. Chen, and C.-C. Lee, "A blockchain based data aggregation and group authentication scheme for electronic medical system," *IEEE Access*, vol. 8, pp. 173904–173917, 2020.

[30] Z. Cui, F. Xue, S. Zhang, X. Cai, Y. Cao, W. Zhang, and J. Chen, "A hybrid BlockChain-based identity authentication scheme for multi-WSN," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 241–251, Apr. 2020.

[31] S. Gao, Q. Su, R. Zhang, J. Zhu, Z. Sui, and J. Wang, "A privacy-preserving identity authentication scheme based on the blockchain," *Secur. Commun. Netw.*, vol. 2021, pp. 1–10, Jun. 2021.

[32] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[33] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 426, no. 1871, pp. 233–271, 1989.

[34] P. Yabo. (2020). *Comparison of Cryptocurrency Developments. Key Metrics of Blockchain Platforms*. CoinfaBrik Blog. [Online]. Available: https://bit.ly/3vSDVZ0

[35] T.-T. Kuo, H. Z. Rojas, and L. Ohno-Machado, "Comparison of blockchain platforms: A systematic review and healthcare examples," *J. Amer. Med. Inform. Assoc.*, vol. 26, no. 5, pp. 462–478, 2019.

[36] M. Macdonald, L. Liu-Thorrold, and R. Julien, "The blockchain: A comparison of platforms and their uses beyond bitcoin," *Work. Papers*, pp. 1–18, May 2017.

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

[38] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably secure authenticated key agreement scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 3, pp. 1900–1910, May 2018.

[39] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer 2001, pp. 453–474.

[40] J.-L. Tsai and N.-W. Lo, "Secure anonymous key distribution scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 906–914, Mar. 2016.

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

[42] M. Turuani, "The CL-Atse protocol analyser," in *Proc. Int. Conf. Rewriting Techn. Appl.* Berlin, Germany: Springer, 2006, pp. 277–286.

[43] A. Armando and L. Compagna, "SATMC: A SAT-based model checker for security protocols," in *Proc. Eur. Workshop Logics Artif. Intell.* Berlin, Germany: Springer, 2004, pp. 730–733.

[44] Y. Boichut, P. C. Héam, O. Kouchnarenko, and F. Oehl, "Improvements on the Genet and Klay technique to automatically verify security protocols," in *Proc. AVIS*, vol. 4, 2004, p. 84.

[45] A. Irshad, M. Sher, S. A. Chaudhary, H. Naqvi, and M. S. Farash, "An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging registration centre," *J. Supercomput.*, vol. 72, no. 4, pp. 1623–1644, Apr. 2016.

[46] S. A. Chaudhry, H. Naqvi, M. S. Farash, T. Shon, and M. Sher, "An improved and robust biometrics-based three factor authentication scheme for multiserver environments," *J. Supercomput.*, vol. 74, no. 8, pp. 3504–3520, Aug. 2018.

[47] D. Xu, J. Chen, and Q. Liu, "Provably secure anonymous three-factor authentication scheme for multi-server environments," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 2, pp. 611–627, 2019.

[48] C.-L. Lei and Y.-H. Chuang, "Privacy protection for telecare medicine information systems with multiple servers using a biometric-based authenticated key agreement scheme," *IEEE Access*, vol. 7, pp. 186480–186490, 2019.

[49] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Comput. Netw.*, vol. 73, pp. 41–57, Jul. 2014.

[50] V. Odelu, A. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Gener. Comput. Syst.*, vol. 68, pp. 74–88, Mar. 2017.

MANJUNATH HEGDE received the master's degree in computer science from Mangalore University Karnataka, India, in 2014, and the Ph.D. degree in mathematical and computational sciences from the National Institute of Technology Karnataka, India, in 2019. He is currently working as an Assistant Professor with the Department of Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. His research interests include network security, information security, secure authentication, cryptography, and blockchain technology.

ROHINI R. RAO received the Ph.D. degree in electronic health records with respect to interoperability and privacy. She is currently a Faculty with the Department of Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. She is a Researcher, working on cost-effective IT solutions and technology-based interventions for public health. Her current research interests include data science, public health informatics, blockchain technologies, and health care analytics.

B. M. NIKHIL received the bachelor's degree in electronics and communication from the Manipal Institute of Technology, Manipal, Karnataka, India, in 2020. His research interests include machine learning, image processing, the IoT, secure authentication, and blockchain technology.