Received: 20 February 2023 | *Revised:* 14 April 2023 | *Accepted:* 14 April 2023
DOI: 10.37917/ijeee.19.2.9 *Vol. 19 | Issue 2 | December 2023*

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access

A Privacy-Preserving Scheme for Managing Secure Data in Healthcare System

Naba M. Hamed*[1], Ali A Yassin[2]
[1]College of Computer Science and Information Technology, University of Basrah, Basrah, 61004, Iraq
[2]Department of Computer science, Education College for Pure Sciences, University of Basrah, Basrah, 61004, Iraq

Correspondance
*Naba M.Hamed
Department of Computer science,
College of Computer Science and Information Technology,
University of Basrah, Basrah, Iraq.
Email: nabawq12@gmail.com

Abstract
In the world of modern technology and the huge spread of its use, it has been combined with healthcare systems and the establishment of electronic health records (EHR) to follow up on patients. This merging of technology with healthcare has allowed for more accurate EHRs that follow a patient to different healthcare facilities. Timely exchange of electronic health information (EHR) between providers is critical for aiding medical research and providing fast patient treatment. As a result, security issues and privacy problems are viewed as significant difficulties in the healthcare system. Several remote user authentication methods have been suggested. In this research, we present a feasible patient EHR migration solution for each patient. finally, each patient may securely delegate their current hospital's information system to a hospital certification authority in order to receive migration proof that can be used to transfer their EHR to a different hospital. In addition, the proposed scheme is based on crypto-hash functions and asymmetric cryptosystems by using homomorphic cryptography. The proposed scheme carried out two exhaustive formal security proofs for the work that was provided. Using Scyther, a formal security tool, we present a secure user authentication technique in the proposed healthcare scheme that ensures security and informal analysis.

Keywords
Electronic Health Records, Scyther, Migration Data, Asymmetric encryption, Homomorphic Cryptography.

I. Introduction

The internet has become an indispensable part of everyday life. Thanks to the fast progress of internet technology, we can now deliver any service from anywhere and at any time [1]. Remote user authentication is becoming an increasingly significant component of gaining access to valuable services or resources in the healthcare system, cloud applications, multi-server configurations, and mobile devices. Remote user authentication is an essential component of any security strategy. In the absence of authentication, audit trails are opaque, and authorization grants identity-based privileges [2]. If we cannot distinguish between authorized and unauthorized parties, secrecy and privacy will be violated. In recent years, various study fields have evolved to improve human life.

An Electronic Health Record (EHR) is a personal medical record incorporated into health information systems [3]. Many countries create health information systems to help administer each patient's activities and health monitoring. Consider the following scenario: A patient (let's call her Alice) plans to see a doctor at a new hospital. If she visits a new hospital, she may be required to disclose her personal medical information again. Furthermore, if her doctor needs her medical treatment history from other institutions, she must decide how to securely communicate this information to her doctor. These issues are very pressing. Our proposed approach guarantees that data access and data transfer are simple and secure. Each

user must be granted the proper access rights [4]. One of the most straightforward and practical security solutions is password-based authentication. Password-based authentication mechanisms are used in the healthcare system, e-business, database management systems, and smart card applications. Our method presents a feasible and verifiable patient EHR fair exchange for health information systems. Patients must not only delegate the transfer of their personal EHR from their current hospital health information system to the hospital system of their choice but also retain their privacy [4]. Our system ensures safe data storage and the secure transmission of permitted information to a specified place. We propose a high-level, realistic, and demonstrable patient EHR fair exchange model with key agreements for health information systems. A patient can not only delegate the current hospital's health information systems to migrate their personal EHR to the chosen hospital system but also maintain their privacy [5].

In India, EHR guidelines advocate for the safe sharing of health information with minimal disclosure of personal identification. The majority of identity-related breaches are triggered by the leak of sensitive information associated with identifiers, as well as the vast data collection and tracking permitted by service providers [6]. The General Data Protection Regulation encourages entity-controlled identifiers and limited information collection to preserve privacy. Many countries have laws in place to protect patients' privacy, such as the Health Insurance Portability and Accountability Act (HIPAA)in the United States. Confidentiality in telecare services has become a key problem, especially how to ensure patient data security and privacy while transmitting over a public channel [7] [8]. User authentication is the first step in ensuring that only authorized users have access to protected data. Although password-based user authentication is the most convenient approach, it is prone to numerous attacks and may pose a threat to data security. Multifactor authentication is a recommended strategy in which any user is granted access to specified data after verifying two or more pieces of evidence [9] [10] perform poorly or have serious security flaws in the context of tele-health services. Our system stores data and securely transfers approved information to defined places. In this paper, we offer a safe technique for authenticating EHR patients and the Healthcare Center Server using real member IDs and verification codes. To provide robust security while maintaining good performance. The proposed work employs, based on multi-factor authentication, a lightweight crypto-hash function for the generation of One-Time Passwords (OTP) and symmetric key encryption (CTR mode) and an asymmetric key of homomorphic encryption Damgoard, Geisler and Kroigaard (DGK) to offer strong security performance. The primary goal of this research is to provide a robust authentication mechanism to address difficulties identified in previous studies. In practice, the Scyther security proof was utilized to show the strong security and resistance of our strategy against hostile attacks. The suggested technique strikes a good compromise between security complexity and performance, and it may be used in healthcare systems.

The remainder of the article is structured as follows. Section 2 reviews the related work. Section 3 focuses on the proposed scheme. Section 4 This section evaluates security analysis of the proposed scheme in terms of formal and informal security analysis. Section 5 presents the performance results. Finally, section 6 presents the conclusions.

II. Related Work

A. Centralized Identity Management for Entity Authentication

As part of a centralized identity management system, a centralized identity distribution point (IDP) will be responsible for issuing an identity (email ID, phone number, government identification code, patient identification code), as well as for maintaining the trust factor associated with those identities. Credential-based authentication is a well-established initial line of defense in any identification scheme. As the privacy and security of patient data records are critical in EHRs, credential-based protection is a relatively simple and adaptable first-hand solution that is able to identity theft, spoofing attacks, data loss, and other types of privacy violations. It is possible to add an extra layer of protection to the current credentials-based authentication by including secondary factors such as OTP, captchas, patterns, or biometrics in addition to the credentials-based authentication [11]. Many studies on two-factor authentication [12] and three-factor authentication [13] have been conducted for the purpose of validating medical records [12]. Although it adds an extra layer of protection, multifactor authentication is vulnerable to attacks such as identity theft, replay attacks, phishing attacks, and denial of service attacks, among others. The authentication of entities can be achieved by binding centralized identifiers to cryptographically generated keys, signatures, and certificates with the help of public key infrastructure (PKI).

B. Decentralized identity management for entity authentication

The whole concept of decentralization is founded on the basic assumption that a transaction for the transfer of a commodity or asset between two parties is accepted by the participating nodes through the use of a consensus mechanism. This transaction is recorded in an immutable distributed ledger as part of the transaction log. Blockchain is a practical distributed ledger technology, and it was this protocol that introduced the concept of decentralization into the settlement of financial transactions. Later, the framework was generalized in

the healthcare ecosystem by introducing programming capabilities using smart contracts. were the first to propose the use of blockchain technology in the design of healthcare or the purpose of decentralized identity management. MedRec [14], was the first functioning prototype of a blockchain-based system for accessing health records that is built on Ethereum smart contracts. Additionally, a solution for identity management and verification that uses blockchain technology was created [15]. The system aims to enable greater flexibility in health record access while simultaneously increasing patient data privacy. Furthermore, an efficient authentication mechanism for a hospital network based on blockchain was proposed [16] for the identification of distributed patients among others. Additionally, [17] presented a group authentication approach that would allow authorized group members to access sensitive health information in the context of a remote medical monitoring system. Moreover, using blockchain technology, [18] created a multi-identity verification system for a secure medical data sharing paradigm, preventing dependence on a third party [19] which allows signers to update their certificates without having to sign again. Furthermore, a decentralized, secure, and lightweight certificate-less signature protocol was proposed by transforming the logic of the key generation center (KGC) into smart contract code, which can withstand KGC compromised attacks and distributed denial of service attacks [20]. However, none of the above-mentioned approaches takes into account the integration of authentication with access control to increase the overall system efficiency. Consequently, the fundamental purpose of this research is to provide a robust authentication technique based on cryptosystem tools to solve issues highlighted in previous studies and provide an efficient, verifiable, and practical EHR fair exchange method, allowing each patient to safely transfer their own EHR from one institution to another. The proposed approach may also provide ease, speed, and integrity. We built a high-level, realistic, and verifiable EHR fair exchange plan with essential agreement for the health information system. A patient could not only delegate the current hospital's health information systems to move their personal EHR to the chosen hospital system but also maintain their privacy. We demonstrated the security of our protocol using security analysis and the Scyther tool in the security analysis discussed in the following section. The performance comparison and efficiency analysis findings show that the proposed approach delivers a greater level of security while maintaining computational economy.

III. Proposed Scheme

The major purpose of the proposed scheme is to enable safe patient-centric EHR access while also providing efficient data security and administration. Users with access based on their professional duties, such as physicians, nurses, and medical researchers, make up the user data. In practice, user data might be assigned to a separate sector of society, such as healthcare. It also includes users who are intimately connected to a data owner (for example, family members or close friends), and have access to EHRs based on access privileges granted by the EHR's owner. The architecture is made up of four parts: EHR owner (EHR_{W_i}), EHR user (EHR_{U_j}), Cloud Health Server ($CHS(CHS_k)$), a hospital certification authority HCA assists a patient (EHR_{U_j}) in generating the patient's migration permit signature to another hospital or medical center in the public key infrastructure (PKI); where $(1 \leq i \leq N)$,$(1 \leq j \leq M)$,$(1 \leq k \leq Z)$; each of N, M, Z represent the number of patient EHR_W, users (EHR_U), healthcare centers (CHS), respectively. The EHR_{W_i} is the individual whose medical information is contained in the record, and he has full access to that data. The owner might share his information with friends, physicians, or nurses to seek clinical advice. The EHR_{U_j} may be in the public or private sectors, and their rights are determined by their roles with the EHR owner. A user can be a healthcare professional such as a doctor, a friend, a family member, or emergency personnel. A CHS_k is a storage facility that houses and manipulates sensitive health data. Maintaining data privacy and accuracy of patients necessitates a higher level of vigilance. The EHR owner relies on the cloud server for remote data storage and record maintenance, alleviating the burden of establishing and maintaining local storage infrastructure. Most cloud data storage services also offer benefits such as availability, scalability, low cost, and on-demand data sharing among a group of trusted users, such as physicians, insurance companies, emergency personnel, family and friends in a collaboration team, or employees in an enterprise organization. Because the data owner no longer has physical control over the data, it is vital to allow the data owner to check that his data is being saved and maintained appropriately in the cloud. The registration phase, the EHR migration phase, and the data exchange phase comprise the four steps of our proposed scheme.

A. Registration Phase

In this phase, hospital certification authority (HCA) is responsible to distributes the key parameters and certificate between main components.

1) Cloud Health Server Side

Each health establishment (Cloud Health Server (CHS_k)) should be identified as a health mother institution (hospital certification authority (HCA)) for the purpose of achieving, distributing, and exchanging data among different patients belonging to different establishments. HCA applies the following steps for each CHS_k.

- Step1. Compute a public key, $(PU_{(CHS_k)} = (N, g, h, u))$.
- Step2. Compute a private key, $(Pr_{(CHS_k)} = (p, q, v_p, v_q))$.
- Step3. Send the tuple $(ID_{(CHS_k)}, ID_{HCA}, PU_{(CHS_k)}, Pr_{(CHS_k)})$ to CHS_k and declare the public key $PU_{(CHS_k)}$ to other healthcare institutions $CHS_1, CHS_2, \ldots CHS_n$ (see Figure 1).

2) Patient Side

A patient (W_i) sends request to (CHS_k) for registering and getting his electronic healthcare record EHR owner$(EHR_{(W_i)})$ that consists of sensitive information such as $(ID_{(W_i)}, PW_{(W_i)}, Address_{(W_i)}, Email_{(W_i)}, \ldots$ etc.). However, CHS_k first prepares hash function that is H, where $H : Z_n^* \rightarrow \{0,1\}^1$. CHS_k prepares anomaly parameters $ID_{AW_i} = H(ID_{W_i})$, $PW_{AW_i} = H(PW_{W_i})$. Then, CHS_k forwards patient's request based on his anomaly parameters to the HCA to help W_i obtaining the permission parameters from HCA that implementing the following steps:

- Step1. Generate Shared key $(SK_{(W_i)})$ and certificate $(Cert_{(W_i)})$.
- Step2. Send the tuple $(SK_{(W_i)}, ID_{(AW_i)}, PW_{(AW_i)}, Cert_{(W_i)})$ to W_i via CHS_k.
- Step3. CHS_k upgrades the main information of $EHR_{(W_i)}$ such as $ID_{(AW_i)}, PW_{(AW_i)}$ and keeps the shared key for using it in the next phases.

Finally, the $EHR_{(W_i)}$ is active to use in the healthcare system and applied key operations (update, insert, delete) on it (see Fig. 1).

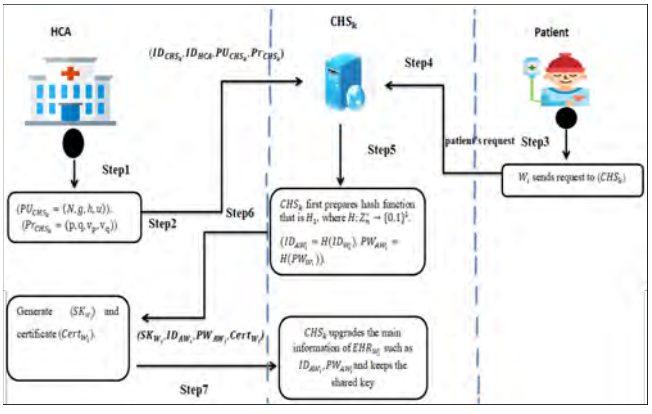

Fig. 1. Registration phase of cloud health server side and patient side.

3) User Side

In the healthcare system, there is important part represented by users like employees, doctors, administrator. The user (U_i) sends his request to (CHS_k) for registering and getting his electronic healthcare record (EHR_{U_i}) that consists of sensitive information such as $(ID_{U_i}, PW_{U_i}, Address_{U_i}, Email_{U_i}, \ldots etc.)$. However, CHS_k prepares anomaly parameters $ID_{AU_i} = H(ID_{U_i})$, $PW_{AU_i} = H(PW_{U_i})$ and forwards user's request (ID_{AU_i}, PW_{AU_i}) to HCA. The following steps performed by HCA to generate main keys.

- Step1. Generate Shared key (SK_{U_i}) and certificate $(Cert_{U_i})$.
- Step2. Send the tuple $(SK_{U_i}, ID_{AU_i}, PW_{AU_i}, Cert_{U_i})$ to U_i via CHS_k.
- Step3. CHS_k upgrades the main information of $EHR_{E_i} < SK_{U_i}, ID_{AU_i}, PW_{AU_i}, Cert_{U_i}, \ldots >$ for using it in the next phases (see Fig. 2).

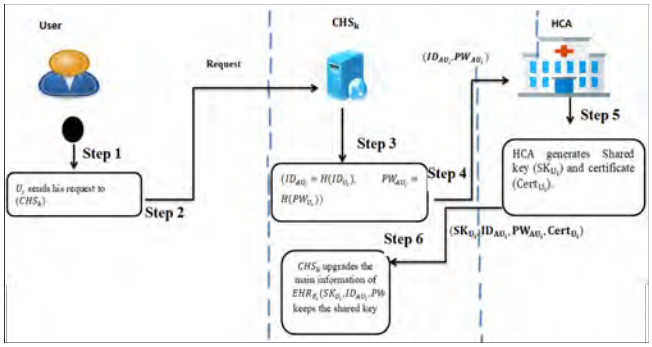

Fig. 2. Registration phase of user phase.

B. Login and Authentication Phase

In this phase, the patient and employee want to login the healthcare system , receiving report from his doctor or sending quires to his doctor for accessing EHR_{W_i} existed CHS_k.

1) Patient Side (The owner of EHR)

The patient (W_i) desires to access the system in order to view his electronic health record (EHR_{W_i}), gets a report from his doctor, or sends some queries to his doctors. There are main steps to allow W_i from accessing the system as follows:

- Step1. The W_i enters his ID_{W_i}, PW_{W_i} and then generates an integer random number $r_i \in Z_n^*$. Finally, he calculates an anonymity of identity and one-time password $ID'_{AW_i} = H(ID_{W_i})$, $PW'_{AW_i} = H(H(PW_{W_i}) \oplus r_i))$, respectively.
- Step2. W_i encrypts $E_{W_i} = Enc_{SK_{W_i}}(r_i)$ using symmetric key and $E_{HW_i} = HEnc_{SK_{W_i}}(PW'_{AW_i}) = g^{PW'_{AW_i}} h^{r_i} mod N$, which is based on homomorphic encryption.

- Step3. W_i sends his login request $< ID'_{AW_i}, E_{H(W_i)}, E_{W_i} >$ to CHS_k.

- Step4. In the Cloud Healthcare server side, CHS_k verifies patient's login request as follows.

  (a) $ID_{W_i} \stackrel{?}{=} ID'_{AW_i}$; if so, CHS_k restores random number by decrypting $r_{i'} = Dec_{SK_{W_i}}(E_{W_i})$.

  (b) CHS_k computes $PW''_{AW_i} = H(H(PW_{W_i}) \oplus r_{i'}))$ and compares between $E_{HW_i} \stackrel{?}{=} g^{PW''}_{AW_i} h^{r_{i'}} modN$. If so, he accepts; CHS_k sends challenge as verification code (VC) to W_i. Where, VC represents SMS message that is sent via mobile communication channel.

- Step5. As a result, W_i retrieves verification code (VC') via his mobile phone number and computes $CH_{W_i} = H(Cert_{W_i} \oplus VC')$. Then, he replies CH_{W_i} to CHS_k.

- Step6. CHS_k computes $CH'_{W_i} = H(Cert_{W_i} \oplus VC')$ and compares between $CH_{W_i} \stackrel{?}{=} CH'_{W_i}$. If so, CHS_k accepts the user's login request and allows him to use the resources and services od system based on his privileges. Otherwise, he rejects the login phase (see Fig. 3).

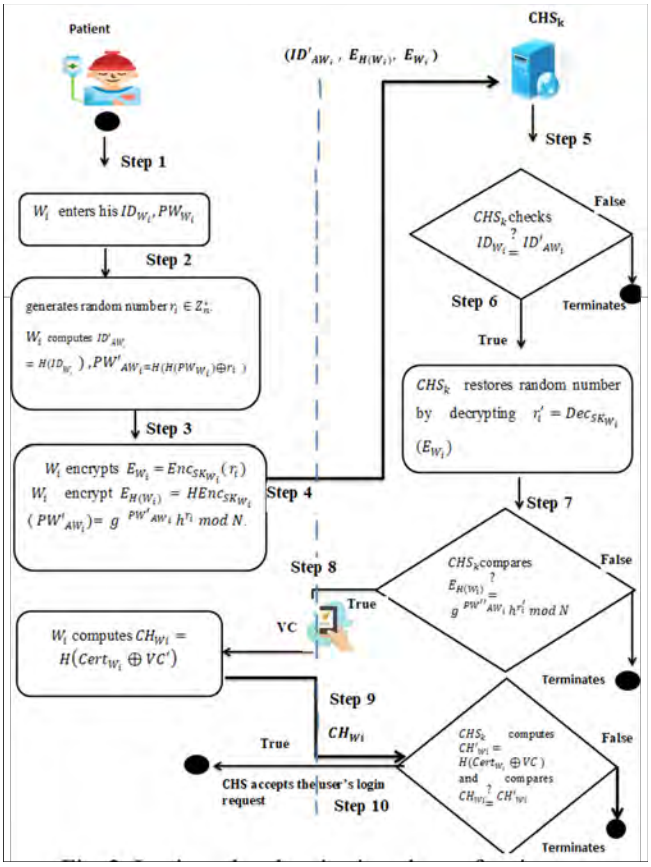

Fig. 3. Login and authentication phase of patient.

2) User Side

The U_i wishes to login system for checking EHR_{W_i} of patient based on his role and privileges. The details of main steps are viewed as follows:

- Step1. $U_i \rightarrow CHS_k : ID'_{U_i}, E_{HU_i}, E_{U_i}$. U_i performs the following computations:

  (a) U_i enters his ID_{U_i}, PW_{U_i} and then generates random number $r_i \in Z_{n^*}$. Finally, U_i computes $ID'_{AU_i} = H(ID_{U_i})$, $PW'_{AU_i} = H(H(PW_{U_i}) \oplus r_i)$, respectively.

  (b) U_i encrypts $E_{U_i} = Enc_{SK_{U_i}}(r_i)$ using symmetric key and $E_{HU_i} = HEnc_{SK_{U_i}}(PW'_{AU_i}) = g^{PW'_{AU_i}} h^{r_i} modN$, which is based on homomorphic encryption.

  (c) U_i sends his login request $< ID'_{AU_i}, E_{HU_i}, E_{U_i} >$ to CHS_k as a first factor.

- Step2. $CHS_k \rightarrow U_i : QR_{U_i}$. CHS_k verifies patient's login request as follows:

  (a) CHS_k checks $ID_{U_i} \stackrel{?}{=} ID'_{AU_i}$; if the verification of $ID_{U_i} \stackrel{?}{=} ID'_{AU_i}$ is successful, CHS_k restores random number by decrypting $r'_i = Dec_{SK_{U_i}}(E_{U_i})$.

  (b) CHS_k computes $PW''_{AU_i} = H(H(PW_{U_i}) \oplus r'_i))$ and compares between $E_(H(U_i)) \stackrel{?}{=} g^{PW''_{AU_i}} h^{r'_i} modN$. If so, CHS_k generates and encrypts verification code $(VC_{U_i})E_{U_i} = Enc_{SK_{U_i}}(VC_{U_i})$ and generates Quick Response Code QR_{U_i} that contains encrypted verification code (VC_{U_i}) (see Fig. 2. Then, CHS_k sends (QR_{U_i}) to U_i. Where, VC_{U_i} represents SMS message that is sent by CHS_k via mobile communication channel.

- Step3 . $U_i \rightarrow CHS_k : CH_{U_i}$. Upon receiving this information in Step 2, U_i computes:

  (a) He works on reading (QR_{U_i}) using QR & Barcode Scanner. After scanning step, he will get (E_{U_i}) and decrypts $VC'_{U_i} = Dec_{SK_{U_i}}(E_{U_i})$.

  (b) He computes $CH_{U_i} = H(Cert_{U_i} \oplus VC'_{U_i})$. After that, U_i computes $SK_{U_i} = SK_{U_i} \oplus VC'_{U_i}$. Then, computes $E_{Cert_{U_i}} = Enc_{SK'_{U_i}}(Cert_{U_i})$.

  (c) He sends the tuple $< E_{Cert_{U_i}}, CH_{U_i} >$ to CHS_k.

- Step4. Upon receiving the information in Step 3, CHS_k computes $CH'_{U_i} = H(Cert_{U_i} \oplus VC_{U_i})$ and compare between $CH_{U_i} \stackrel{?}{=} CH'_{U_i}$. If so, CHS_k accepts the user's login request and allows him to use the resources and services of system based on his privileges. Then, CHS_k computes $SK_{U_i} = SK_{U_i} \oplus VC_{U_i}$ and decrypts $Cert'_{U_i} = Dec_{SK''_{U_i}}(E_{Cert_{U_i}})$. Otherwise, he rejects the current phase.

Note: Now the user can work according to the his privileges granted to him from administor (doctor, administrator).

C. EHR Migration Phase

In this phase, the patient wishes to receive medical treatment in a certain institution CHS'_K, which does not necessarily be the same institution that registered her/him previously.

- Step1. W_i computes a random value rW_i with a random number $r_i \in Z_n^*$, where $rW_i = r_i \oplus H_{W_i}$. After correctly calculating the foregoing, he sends his request to the CHS_k in an anomaly and freshness message style. The message request includes $(Cert_{W_i}, ID'_{AW_i}, E_{AW_i})$, which is computed from $(ID'_{AW_i} = ID_{AW_i} \oplus rW_i)$ and encrypted main parameter via his shared key $E_{AW_i} = Enc_{SK_{W_i}}(rW_i)$.
  $W_i \overset{\left(\text{Cert}_{W_i}, D'_{AW_i}, E_{AW_i}\right)}{\longrightarrow} CHS_k$
- Step2. Following the receipt of this message by CHS_k, it can check the $Cert_{W_i}$ with his index file; if it is found then go to step 3. Otherwise, go to the Step4.
- Step3. W_i performs the main medical treatment in his institution, the results report (RR_{W_i}) should be added to the EHR_{W_i} by EHR_{U_j} directly and apply the same functions in the upgrading phase.
- Step4. This case means that the patient wishes to do some medical treatments outside of his healthcare center. The new institution CHS'_K is used public key of HCA to encrypt $AE_{pk1} = AEnc_{PU_{HCA}}((\text{Cert}_{W_i}, ID'_{AW_i}, E_{AW_i}))$, and sends $(ID_{CHS'_K}, AE_pk1)$ to HCA for ensuring from the validity of the patient and his institution.
- Step5. This message tuple $ID_{CHS'_K}, AE_pk1$ is delivered to HCA. When HCA has received this message with $ID_{CHS'_K}$, it can decrypt AE_pk1 based on Pr_HCA in order to restore all parameters using $ADec_{Pr_HCA}(AE_pk1)$. First, it can fetch the random value $rW'_i = Dec_{SK_{W_i}}(E_{AW_i})$, we notice this step also verifies certificate of W_i relied on his shared key SK_{W_i} and $Cert_{W_i}$. Second, it compares between ID'_{AW_i} and $ID_{AW_i} \oplus rW'_i$, if they are matched, it ensures from the authority of W_i and saves current parameters for usage in the next steps. Finally, HCA returns the result R to the server CHS'_K using the following function.

$$R = \begin{cases} (H(rW'_i \bigoplus \text{Cert}_{W_i})) & \text{if } W_i \text{ is registered} \\ (H(rW'_i \bigoplus 0)) & \text{if } W_i \text{ isnot registered} \end{cases}$$

  As a result, HCA detects W_i's institution CHS_k based on his certificate $Cert_{W_i}$. It sends (R, ID_{CHS_k}) to CHS'_K.
- Step6. CHS'_K receives this message challenge, it can verify the patient by comparing $HrW_i \oplus Cert_{W_i}$ with R. When the above parameters are not valid, CHS'_K notifies the patient to register at a public healthcare center or checks his authority with his medical institution CHS_k (see Fig. 4).

D. Treatment and Exchanging Phase

In this phase, W_i can do many medical treatments such as tests of blood diseases, blood pressure, diabetes, Covid-19 infection, CT-Scan, MRI in the CHS'_K.

(a) The results report (RR_{W_i}) should be added to the EHR_{W_i} existing in the original patient's institution CHS_k where he belongs in the registration phase. Therefore, CHS'_K computes $AE_pk2 = AEnc_{PU_{CHS_k}}(Cert_{W_i}, RR_{W_i})$ based on the identification of patient's institution detected previously in Step3.2. Finally, CHS'_K sends message tuple $(ID_{CHS'_K}, ID_{CHS_k}, AE_pk2)$ to HCA.

(b) The server HCA will behave according to the delegated message tuple $(ID_{CHS'_K}, ID_{CHS_k}, AE_pk2)$, and will exchange secure data of medical institutions (CHS_k, CHS'_K) by forwarding patient's data (ID_HCA, AE_pk2) to CHS_k.

E. Upgrading Phase

When CHS_k receives (ID_HCA, AE_pk2), it decrypts AE_pk2 with Pr_{CHS_k}. If it is valid, it can obtain RR_{W_i}, $Cert_{W_i}$ and upgrade the information of EHR_{W_i} by adding the new status of the patient W_i relied on RR_{W_i}. The upgrade process will be performed by EHR user (EHR_{U_j}) working as an employee who has privileges that allow him to upgrade to the EHR_{W_i}. Additionally, these privileges gained by the Administrator (ADM), represent the role of U_j. Now, the EHR_{W_i} contains the last update of the patient's case. In an emergency patient's case, EHR_{U_j} can tell the family member about the patient's case by sending SMS-Emergency to the patient's family member (see Fig. 5).

IV. Security Analysis

This section evaluates security analysis of the proposed scheme in terms of formal and informal security analysis as the follows:

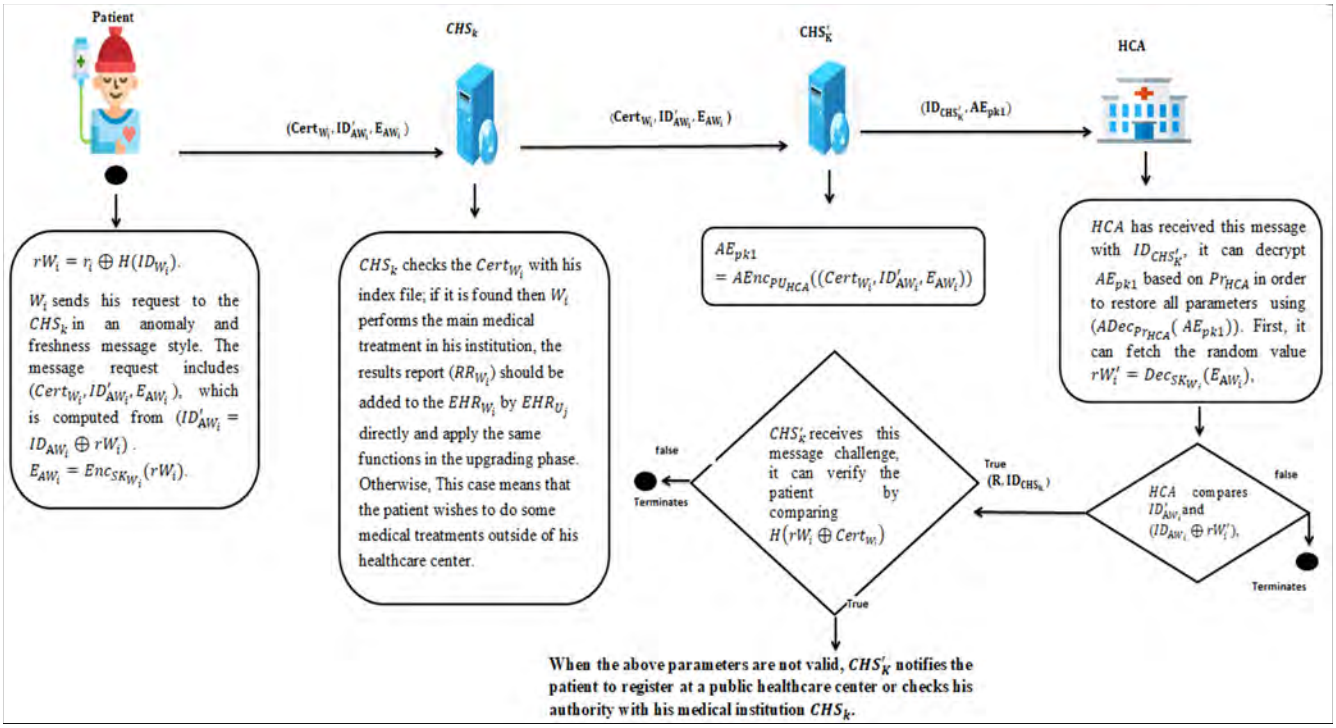

Fig. 4. EHR migration phase.

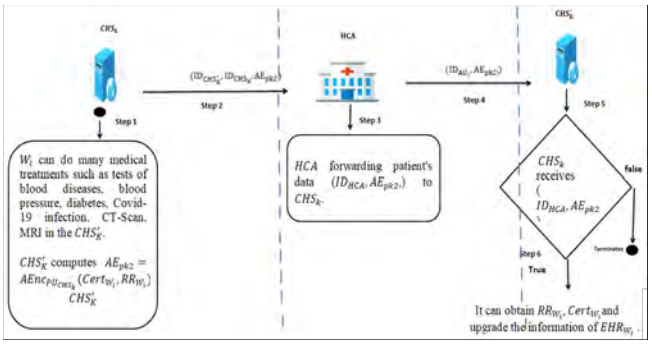

Fig. 5. Explains the treatment and exchange.

A. *Formal Security Analysis*

Scyther, which is based on the Security Protocol Description Language (SPDL) proposed in [21], is a formal verification tool for security protocols. Many security protocols have applied the Scyther tool for verification. Our protocol is verified using the "verification claims" and "automatic claims" schemes in the Scyther tool. Currently, the proposed scheme has been written in SPDL, and the results are viewed as Automatic Claim and Verification Claim. Based on the Scyther tool, our approach resists harmful attacks such as MITM attack, insider attack, replay attack, spoofing, and impersonation The login and authentication phases are depicted in Fig. 6 and Fig. 7.

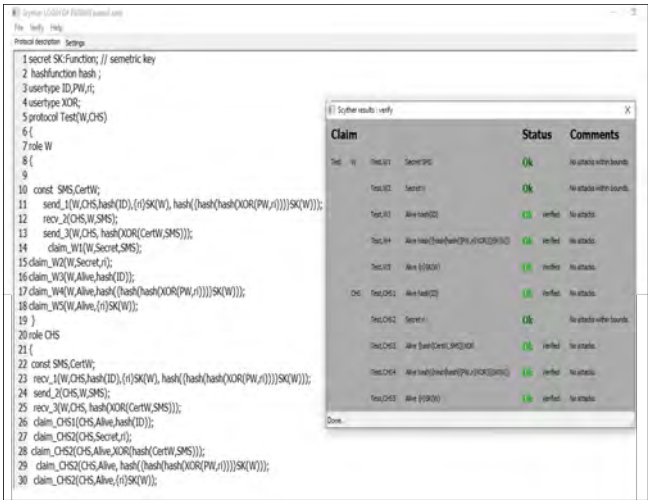

Fig. 6. Login and Authentication phase that cannot be attacked of patient.

Our investigation has revealed that the proposed solution provides security against malicious attacks as previously stated. Because of this, SPDL is capable of performing a number of critical cryptographic activities, such as sending and receiving messages between components, and it also dis-

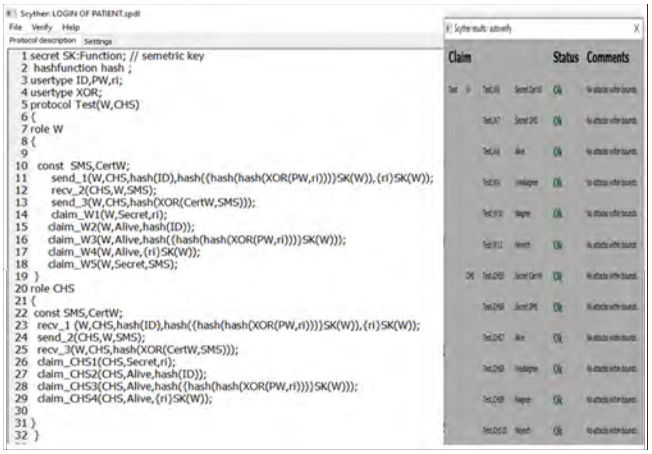

Fig. 7. Login and Authentication phase that cannot be attacked of user.

tinguishes between the obligations that each component bears. After removing the security components of the proposed system, such as crypto hashing and encryption, we will be able to observe the system's apparent vulnerability. As a result, the system becomes unsafe as a result of this, making it more vulnerable to assault by malicious entities (see Fig. 8. Figure 9 demonstrates the safety and security of the Login and Authentication phase that cannot be attacked of user.

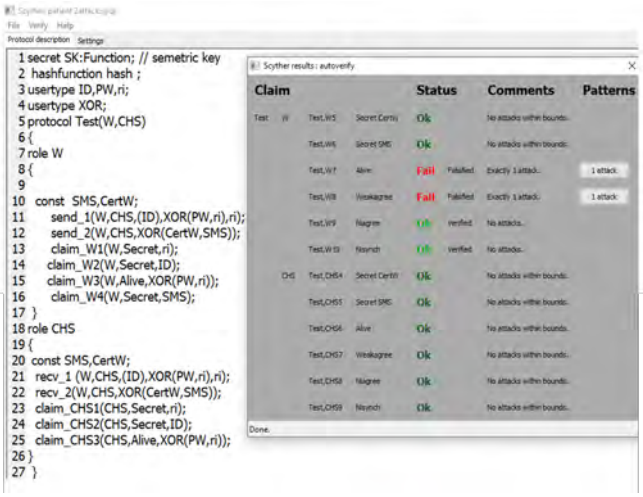

Fig. 8. Login and authentication phase that can be attacked.

B. Informal Security Analysis

In this section, the proposed scheme is proved using an informal method. We aim to resist well-known attacks such as MITM attack, replay attack, and insider attack according to the proposed scheme. Furthermore, the proposed scheme possesses several merits, including user anonymity, mutual

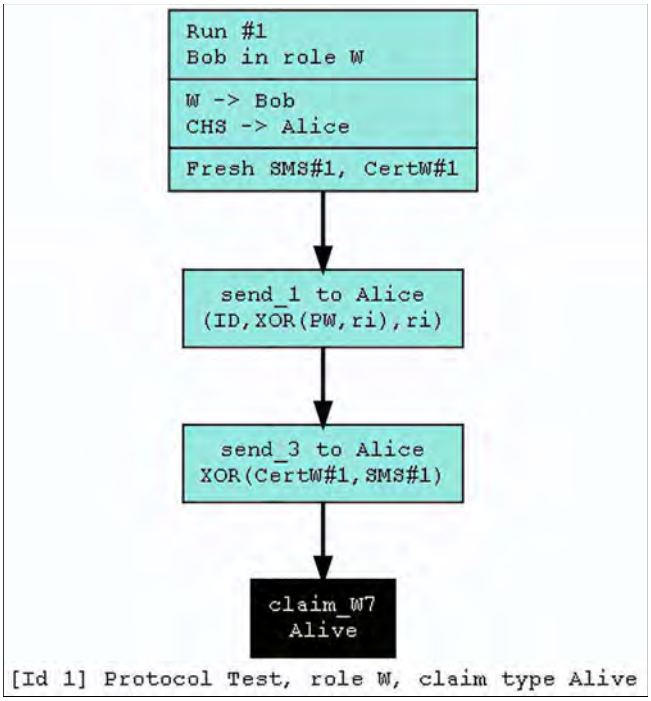

Fig. 9. Model checking of the login and authentication phase of patient.

authentication, and session key agreement.

Proposition 1. Our proposed scheme provides mutual authentication.

Proof. This security feature denotes that an attacker should fail to impersonate the legal system's components (W_i, D_i, ADM, E_i) to CHS_K, and vice versa. In this paper, authentication of U_i to CHS_K has used the following four steps:

- User (U_i), who possesses the secret factors, can successfully bring the factors $(ID'_{AU_i}, E_{HU_i}, E_{U_i})$ to CHS_k as a first factor.
- CHS_k compares $ID_{U_i} \stackrel{?}{=} ID'_{AU_i}$; if the verification of $ID_{U_i} \stackrel{?}{=} ID'_{AU_i}$ is successful, it computes $r'_i = Dec_{SK_{U_i}}(E_{U_i})$. Then, it computes $PW''_{AU_i} = H(H(PW_{U_i}) \oplus r'_i)$ and compares $E_{H(U_i)} \stackrel{?}{=} g^{PW''_{AU_i}} h^{r'_i} mod N$. If so, CHS_k generates and encrypts verification code (VC_{U_i}) $EUi = Enc_{SK_{U_i}}$ (VC_{U_i}) and generates the Quick Response code $(QR_U i)$ that contains the encrypted verification code (VC_{U_i}). Then, CHS_k sends (QR_{U_i}) to U_i.
- Upon receiving this information, U_i scans $(QR_U i)$ using a QR scanner. Subsequently, U_i will get $(E_U i)$ and decrypt $VC'_{U_i} = Dec_{SK_{U_i}}(E_U i)$. Then, it computes $CH_U i = H(Cert_{U_i} \oplus VC'_{U_i})$. Next, U_i computes $SK_{U_i} = SK_{U_i} \oplus$

VC'_{U_i} and then computes $E_{Cert_{U_i}} = Enc_{SK'_{U_i}}(Cert_{U_i})$ and sends $(E_{Cert_{U_i}}, CH_U i)$ to CHS_k as a second factor.

- CHS_k computes $CH'_U i = H(Cert_{U_i} \oplus VC_{U_i})$ and compares $CH_U i \stackrel{?}{=} CH'_U i$. If so, a user is authenticated at the same time. Then, CHS_k computes $SK_{U_i} = SK_{U_i} \oplus VC_{U_i}$ and decrypts $Cert'_{U_i} = Dec_{SK''_{U_i}}(E_{Cert_{U_i}})$. Therefore, our proposed scheme achieves mutual authentication between the two entities (U_i, CHS_K). Otherwise, it rejects the current phase.

Proposition 2. Our proposed scheme can support user anonymity.

Proof. If an attacker tries to eavesdrop on the user's login request, he cannot obtain the user's identity from the crypto hash function since it is embedded with r_i, which is not identified to the attacker. Additionally, r_i generates once for each user's login request. In the login and authentication phase, U_i sends $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i})$ to CHS_K. Thus, it has been encrypted by shared key SK_{U_i} that was known by U_i and CHS_K. Therefore, it is difficult for an attacker to reveal the user's identity, and he cannot restore the shared key that is generated once for each user's login request. This indicates that our proposed scheme can support user anonymity.

Proposition 3. Our proposed scheme can ensure forward secrecy.

Proof. The popular session key relies on SK_{U_i} used in the login and authentication phase. Our proposed scheme protects the password even when the shared key SK_{U_i} is disclosed or leaked. If the shared key SK_{U_i} is revealed by the adversary, the authentication of the system is not impressed to affection of attackers' behaviors, and he cannot use this key in the next login phase since the shared key is generated once based on VC_{U_i}. Furthermore, it is extremely difficult for an adversary to derive PW'_{AU_i} and random number r_i, as well as the attribute of the crypto one-way hash function $PW'_{AU_i} = H(H(PW_{U_i}) \oplus r_i)$. Additionally, if an adversary can eavesdrop all transmitted messages $(ID'_{AU_i}, E_{(HU_i)}, E_{U_i})$, he will be unable to use these parameters again for logging into the system, as these parameters are generated once for each user's login request. Therefore, our proposed scheme ensures perfect forward secrecy.

Proposition 4. Our proposed scheme can provide unlinkability.

Proof. This feature verifies that a user can attempt several logins to the CHS_K to consume resources/services without others being able to connect the logins together to identify the person. In the proposed scheme, each time U_i wants to log into the system, he submits $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i}))$ to CHS_K. Thus, the primitive components of $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i})$ are generated once for each login phase by using the following points:

- CHS_k checks $ID_{U_i} \stackrel{?}{=} ID'_{AU_i}$; if the verification of $ID_{U_i} \stackrel{?}{=} ID'_{AU_i}$ is successful, CHS_k restores random number by decrypting $r'_i = Dec_{SK_{U_i}}(E_{U_i})$.

- CHS_k computes $PW''_{AU_i} = H(H(PW_{U_i}) \oplus r'_i)$ and compares $E_{(}H(U_i)) \stackrel{?}{=} g^{PW'_{AU_i}} h^{r'_i} mod N$. If so, CHS_k generates and encrypts verification code $(VC_{U_i})E_U i = Enc_{SK_{U_i}}(VC_{U_i})$ and generates $QR_U i$ that contains the encrypted verification code (VC_{U_i}). Then, CHS_k sends $QR_U i$ to U_i.

As a result, the primitive parameters of $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i})$ generate once, and CHS_k cannot link many logins with the same U_i. Therefore, the proposed scheme can provide unlinkability.

Proposition 5. Our proposed scheme is resistant to replay attacks.

Proof. In a replay attack, an adversary intercepts the login message delivered by a legitimate user to the CHS_k and replays it back to the attacker. Then, the adversary reuses this message to impersonate the user when logging into the system in the next session. In our proposed scheme, each new login request should be identical to $CHS'_k s$ parameters $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i}, E_{Cert_{U_i}}, CH_U i)$, and he will be unable to use these parameters again for logging into the system, as these parameters are generated once based on r_i for each user's login request and he will be unable to get r_i. Therefore, an adversary cannot pass any replayed message to the CHS_k verification. Moreover, our approach can resist this attack without synchronization clocks. Therefore, an adversary will fail to apply this type of attack .

Proposition 6. Our proposed scheme can resist MITM attacks.

Proof. An MITM attack intercepts a conversation between the parties to the communication. The conversation appears normal for both parties; however, all the information exchanged passes through the attacker, and he can eavesdrop or modify and re-send. We assume that the attacker has obtained $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i})$ and modified it as $(ID'^{*}_{AU_i}, E_{H(U_i)}*, E_{U_i}*)$; the modified parameters do not work, as CHS_k verifies the $ID'*_{AU_i}$ that was sent by the U_i, and finds that $(ID'_{AU_i} \neq ID'*_{AU_i})$. Additionally, the message $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i})$ is generated once for each login phase. Thus, the proposed scheme does not allow MITM attacks .

Proposition 7. Our proposed scheme is resistant to eavesdropping.

Proof. This is the process of intercepting and examining messages to extract information from them. All parameters exchanged between U_i and CHS_k are the parameters used only once $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i}, r_i, SK_{U_i} and VC_{U_i})$; therefore, if

eavesdropping these parameters, the attacker will fail to enter the system.

- U_i sends $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i})$ to CHS_k.
- CHS_k sends $QR_U i$ to U_i.
- U_i sends $(E_{Cert_{U_i}}, CH_U itoCHS_k)$.

Note: parameters are generated once for each admin's login request. Accordingly, the proposed scheme is resistant to eavesdropping.

Proposition 8. Our proposed scheme can withstand an insider attack.

Proof. In our proposed scheme, when U_i wishes to register with a cloud health server, he sends $ID'_{AU_i}, E_{H(U_i)}$ instead of ID_{U_i}, PW_{U_i}. Due to the utilization of the one-way hash function h(), it is difficult for the attacker to extract the password of the user from the hashed value. In addition, when the attacker wants to impersonate the valid user, he needs to forge a legal login request parameter $(ID'_{AU_i}, E_{H(U_i)}, E_{U_i})$, in which $ID'_{AU_i} = H(ID_{U_i})$, $E_{HU_i} = HEnc_{SK_{U_i}}(PW'_{AU_i}) = g^{(}PW'_{AU_i})h^{r_i} modN$, $E_U i = Enc_{SK_{U_i}}(r_i)$. However, the attacker will be unable to obtain the SK_{U_i} of the user and will fail to forge such parameters.

Proposition 9. Our proposed scheme provides key management.

Proof. The primary parties have agreed to produce a shared key for each login request based on (SK_{U_i}, r_i). When the patient successfully logs in, the primary parties (U_i, CHS_k) execute the following steps to implement this phase:

- The user (U_i) computes $SK_{U_i} = SK_{U_i} \oplus r_i$.
- The (CHS_k) side computes $SK_{U_i} = SK_{U_i} \oplus r'_i$.

Therefore, we notice that our work has key management metric.

Proposition 10. Our proposed scheme provides EHR migration phase in secure manner.

Proof. In this phase, the patient wishes to obtain medical care in a given institution CHS'_K, which will not necessarily be the same institution that enrolled them earlier. In this paper as following steps:

- W_i who possesses the secret factors can successfully bring the factors sends $(Cert_{W_i}, ID'_{AW_i}, E_{AW_i})$ to CHS_k, where $rW_i = r_i \oplus H(ID_{W_i}), (ID'_{AW_i} = ID_{AW_i} \oplus rW_i)$ and the encrypted main parameter via his shared key $E_{AW_i} = Enc_{SK_{W_i}}(rW_i)$.
- CHS_k checks the $Cert_{W_i}$ with his index file; if it is found, W_i performs the main medical treatment in his institution, the results report (RR_{W_i}) should be added to the EHR_{W_i} by EHR_{U_j} directly.
- If the patient wishes to have medical treatments outside of his healthcare center, the new institution CHS'_K sends $(ID_{CHS'_K}, AE_pk1)$ to HCA , where $AE_{pk1} = AEnc_{PU_{HCA}}(Cert_{W_i}), ID_{AW'_i}, E_{AW_i}$.
- When HCA has received this message with $ID_{CHS'_K}$, it can decrypt AE_pk1 based on Pr_HCA to restore all parameters using $(ADec_{Pr_HCA}AE_pk1)$. First, it can fetch the random value $rW'_i = Dec_{SK_{W_i}}(E_{AW_i})$. Second, it compares ID'_{AW_i} and $(ID_{AW_i} \oplus rW'_i)$, and if they are matched, it ensures from the authority of W_i and saves the current parameters for usage in the next steps. Finally, HCA sends (R, ID_{CHS_k}) to CHS'_K, where

$$R = \begin{cases} \left(H\left(rW'_i \bigoplus Cert_{W_i}\right)\right) & \text{if } W_i \text{ is registered} \\ \left(H\left(rW'_i \oplus 0\right)\right) & \text{if } W_i \text{ is not registered} \end{cases}$$

- CHS'_K receives this message challenge, and it can verify the patient by comparing $H(rW_i \oplus Cert_{W_i})$ with R. When the above parameters are not valid, CHS'_K notifies the patient to register at a public healthcare center or checks his authority with his medical institution CHS_k.

V. Performance Analysis

A. Computational Cost

The computational cost is used to calculate the proposed scheme's temporal complexity. Table I compares the computational costs of the most significant similar schemes with that of our technique and compares our technique with other relevant research. Table II compares important security features of the proposed approach with earlier efforts. Furthermore, depending on [22], the processing times for the fundamental functions are roughly as follows applying the following rules(see Fig. 10).

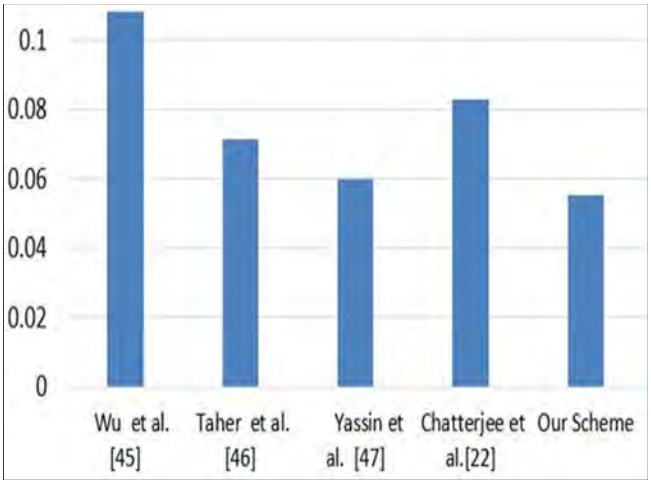

Fig. 10. Computation cost comparison.

TABLE I.
COMPUTATION COST COMPARISON WITH OTHER RELATED WORKS.

Term	Meaning	Time needed
T_h	The time allotted to the crypto hash function.	0.0023 ms
$T_{\oplus}$	The processing time for the XOR operation.	Negligible
T_Enc	The processing time for a symmetric encryption function.	0.0046 ms
$T_{\|\|}$	The processing time for the Concatenation operation.	Negligible

TABLE II.
COMPARING OF THE COMPUTATIONAL COST.

Scheme	Registration Phase	Login and Authentication Phases	Total Cost
Wu et al. [21]	$8T_h + 3T_{\oplus} + 7T_{II}$	$35T_h + 11T_{\oplus} + 30T_{II} + 1T_{Dec} + 1T_{Enc}$	$43T_h + 14T_{\oplus} + 37T_{II} + 1T_{Dec} + 1T_{Enc} \approx 0.1081$
Taher et al. [22]	$10T_h + 10T_{\oplus} + 9T_{II}$	$21T_h + 32T_{\oplus} + 19T_{II}$	$31T_h + 42T_{\oplus} + 28T_{II} \approx 0.0713$
Yassin et al. [23]	$5T_h + 2T_{\oplus} + 1T_{II}$	$13T_h + 12T_{\oplus} + 6T_{II} + 2T_{Dec} + 2T_{Enc}$	$18T_h + 14T_{\oplus} + 7T_{II} + 2T_{Dec} + 2T_{Enc} \approx 0.0598$
Chatterjee et al. [24]	$6T_h + 3T_{\oplus} + 15T_{II}$	$2T_{Dec} + 2T_{Enc} + 22T_h + 5T_{\oplus} + 88T_{II}$	$28T_h + 8T_{\oplus} + 103T_{II} + 2T_{Dec} + 2T_{Enc} \approx 0.0828$
Our Scheme	$2T_h$	$8T_h + 4T_{Enc} + 3T_{Dec} + 6T_{\oplus}$	$10T_h + 4T_{Enc} + 3T_{Dec} + 6T_{\oplus} \approx 0.0552$

TABLE III.
COMPARISON WITH OTHER RELATED WORKS.

Security Features	[16]	[17]	[18]	[19]	Our Scheme
Mutual Authentication	YES	YES	NO	YES	YES
Anonymous & Untraceable	YES	YES	YES	YES	YES
Forward Secrecy	YES	YES	NO	YES	YES
Key Agreement	NO	NO	NO	NO	YES
key management	NO	NO	NO	NO	YES
MITM Attack	YES	NO	NO	NO	YES
Replay Attack	YES	YES	YES	NO	YES
Eavesdropping Attack	NO	NO	NO	NO	YES
Unlinkability	YES	NO	NO	NO	YES
EHR Migration phase	NO	NO	NO	NO	YES
Insider attacks	YES	NO	YES	NO	YES

TABLE IV.
COMPARISON WITH OTHER RELATED WORKS.

Authors	No of bits	No of messages
Chatterjee et al. [24]	1280	2
Xiong et al. [19]	1120	3
Tahe et al. [26]	1660	3
Wu et al. [21]	1600	3
Our Scheme	736	3

According to the above-mentioned comparisons, the suggested system has a lower time complexity $(10T_h + 4T_{Enc} + 3T_{Dec} + 6T_{\oplus} \approx 0.0552)$ than those in previous relevant studies. We can see that the proposed system has a fair mix of performance and security aspects (see Table III).

B. Communication Cost

The cost of transmitted messages is assessed during the login and authentication process. We assumed the identity size is 32 bits, the hash value size is 160 bits [25], the cipher text value size is 128 bits, and the cipher text value size is homomorphic 32 bits. Table IV compares our proposed approach with those in previous relevant research.

VI. CONCLUSIONS

EHRs allow authorized health stakeholders to communicate organized medical data to enhance the quality of healthcare delivery. Since the patient's situation may become exceedingly perilous if personal information becomes public, privacy and security are of the utmost importance inside these systems. It is commonly accepted that concerns around safety and secrecy pose substantial challenges to the functioning of the healthcare system. We offer a safe user authentication approach for patients in the healthcare system that uses Scyther, a formal security tool, to validate the proposed scheme's security. Our proposed approach clearly ensures ease, speed, and integrity. Our technique ensures safe data storage and approved information flow to defined sites. To ensure strong security while maintaining appropriate speed, the proposed scheme employs a lightweight crypto hash function for the generation of OTPs and DGK . The major purpose of this research is to provide a trustworthy authentication technique based on cryptosystem tools to solve the issues highlighted in the previous studies. The suggested system will be able to defend against attacks such as MITM, insider, and replay attacks, among others. It is safe to employ features such as mutual authentication, anomalies, key management, and other secure features, and it strives to achieve a mix of speed and security.

TABLE V.
NOTATION USED IN THE PROPOSED SCHEME.

Symbol	Description
U_i	User
CHS	Cloud Healthcare Server
KGC	Key Generator Center
$\oplus$	XOR operation
MITM	Man-In the middle attack
EHR_i	Electronic healthcare record
PU_{CHS_k}	Public key of cloud health server
Pr_{CHS_k}	Private key of cloud health server
ID_{W_i}	Identity of patient W_i.
PW_{W_i}	Password of patient W_i
CHS_k	The current medical establishment
SK_{U_i}	Shared key of user
QR_{Ui}	QRcode of user
$E_{H(W_i)}$	Homomorphic encryption of W_i
$h(.)$	One-way hash function
r_i	The one-time random number generated by user

CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

REFERENCES

[1] P. D. Singh, G. Dhiman, and R. Sharma, "Internet of things for sustaining a smart and secure healthcare system," *Sustainable computing: informatics and systems*, vol. 33, p. 100622, 2022.

[2] M. Hartmann, U. S. Hashmi, and A. Imran, "Edge computing in smart health care systems: Review, challenges, and research directions," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, 2022.

[3] R. Fazal, M. A. Shah, H. A. Khattak, H. T. Rauf, and F. A. Turjman, "Achieving data privacy for decision support systems in times of massive data sharing," *Cluster Computing*, pp. 1–13, 2022.

[4] B. K. Rai, A. Tyagi, B. Arora, and S. Sharma, "Blockchain based electronic healthcare record (ehr)," in *in ICCCE 2021: Springer*, pp. 185–193, 2022.

[5] M. T. Chen and T. H. Lin, "A provable and secure patient electronic health record fair exchange scheme for health information systems," *Applied Sciences*, vol. 11, no. 5, 2021.

[6] T. Manoj, K. Makkithaya, and V. Narendra, "A blockchain based decentralized identifiers for entity authentication in electronic health records," *Cogent Engineering*, vol. 9, no. 1, 2022.

[7] H. A. Younis, I. M. Hayder, I. S. Seger, and H. A. Younis, "Design and implementation of a system that preserves the confidentiality of stream cipher in non-linear flow coding," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, no. 7, pp. 1409–1419, 2020.

[8] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, "Psspr: a source location privacy protection scheme based on sector phantom routing in wsns," *International Journal of Intelligent Systems*, vol. 37, no. 2, pp. 1204–1221, 2022.

[9] N. C. Basjaruddin, S. Ramadhan, F. Adinugraha, and K. Kuspriyanto, "Baggage tracing at airports using near field communication," in *in 2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, pp. 109–113, 2019.

[10] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Bedgehealth: A decentralized architecture for edge-based iomt networks using blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11743–11757, 2021.

[11] I. Indu, P. R. Anand, and V. Bhaskar, "Identity and access management in cloud environment: Mechanisms and challenges," *Engineering science and technology, an international journal*, vol. 21, no. 4, pp. 574–588, 2018.

[12] A. Chaturvedi, D. Mishra, and S. Mukhopadhyay, "An enhanced dynamic id-based authentication scheme for telecare medical information systems," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 1, pp. 54–62, 2017.

[13] K. Renuka, S. Kumari, and X. Li, "Design of a secure three-factor authentication scheme for smart healthcare," *Journal of medical systems*, vol. 43, no. 5, pp. 1–12, 2019.

[14] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *in 2016 2nd international conference on open and big data (OBD)*, pp. 25–30, 2016.

[15] Y. Liang, "Identity verification and management of electronic health records with blockchain technology," in *In 2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 1–3, 2019.

[16] A. Yazdinejad, G. Srivastava, R. M. Parizi, A. Dehghantanha, K. R. Choo, and M. Aledhari, "Decentralized authentication of distributed patients in hospital networks using blockchain," *IEEE journal of biomedical and health informatics*, vol. 24, no. 8, pp. 2146–2156, 2020.

[17] 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.

[18] X. Cheng, F. Chen, D. Xie, H. Sun, and C. Huang, "Design of a secure medical data sharing scheme based on blockchain," *Journal of medical systems*, vol. 44, no. 2, pp. 1–11, 2020.

[19] C. Lin, X. H. D. He, M. K. Khan, and K. K. R. Choo, "A new transitively closed undirected graph authentication scheme for blockchain-based identity management systems," *IEEE Access*, vol. 6, pp. 28203–28212, 2018.

[20] L. Xiong, F. Li, M. He, Z. Liu, and T. Peng, "An efficient privacy-aware authentication scheme with hierarchical access control for mobile cloud computing services," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2309–2323, 2020.

[21] O. Siedlecka-Lamch, "Probabilistic and timed analysis of security protocols," in *In Computational Intelligence in Security for Information Systems Conference*, pp. 142–151, 2019.

[22] M. Kompara, S. H. Islam, and M. Hölbl, "A robust and efficient mutual authentication and key agreement scheme with untraceability for wbans," *Computer Networks*, vol. 148, pp. 196–213, 2019.

[23] T. Y. Wu, L. Yang, Z. Lee, C. M. Chen, J. S. Pan, and S. Islam, "Improved ecc-based three-factor multiserver authentication scheme," *Security and Communication Networks*, vol. 2021, 2021.

[24] B. H. Taher, F. A. H. Liu, H. L. A. A. Yassin, and A. J. Mohammed, "A secure and lightweight three-factor remote user authentication protocol for future iot applications," *Journal of Sensors*, vol. 2021, 2021.

[25] M. H. Alzuwaini and A. A. Yassin, "An efficient mechanism to prevent the phishing attacks," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 17, no. 1, 2021.

[26] A. A. Yassin, J. Yao, and S. Han, "Strong authentication scheme based on hand geometry and smart card factors," *Computers*, vol. 5, no. 3, 2016.