Received: 18 April 2024 | *Revised:* 3 July 2024 | *Accepted:* 12 July 2024
DOI: 10.37917/ijeee.21.1.7 *Vol. 21 | Issue 1 | June 2025*

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access
IJEEE University of Basrah College of Engineering

An Efficient EHR Secure Exchange Among Healthcare Servers Using Light Weight Scheme

Aqeel Adel Yaseen*[1], Kalyani Patel[2], Abdulla J. Aldarwish[1,3], Ali A. Yassin[3]
[1]Department of Computer Science, Science College, Gujarat University, Ahmedabad, 380009, India
[2]K. S. School of Business Management and Information, Gujarat University, Ahmedabad, 380009, India
[3]Department of Computer Science, Education College for Pure Sciences, University of Basrah, Basrah, 61004, Iraq

Correspondance
*Aqeel Adel Yaseen
Department of Computer Science,
Science College, Gujarat University,
Ahmedabad, 380009, India
Email: aay.ali80@gmail.com

Abstract
This work addresses the critical need for secure and patient-controlled Electronic Health Records (EHR) migration among healthcare hospitals' cloud servers (HHS). The relevant approaches often lack robust access control and leave data vulnerable during transfer. Our proposed scheme empowers patients to delegate EHR migration to a trusted Third-Party Hospital (TTPH); which is the Certification Authority (CA) while enforcing access control. The system leverages asymmetric encryption utilizing the Elliptic Curve Digital Signature Algorithm (ECDSA), EEC and ECDSA added robust security and lightness EHR sharing. Patient and user privacy is managed due to anonymity through cryptographic hashing for data protection and utilizes mutual authentication for secure communication. Formal security analysis using the Scyther tool and informal analysis was conducted to validate the system's robustness. The proposed scheme achieved EHR integrity due to the verification of the communicated HHS and ensuring the integrity of the HHS digital certificate during EHR migration. Ultimately, the result achieved in the proposed work demonstrated the scheme's high balance between data security and accuracy of communication, where the best result obtained represented 7.7/ ms as computational cost and 1248 /bits as communication cost compared with the relevant approaches.

Keywords
EHR, EHR Migration, ECDSA, Login Authentication, Security Analysis.

I. Introduction

With the acceleration of the development of the Internet and the rapid spread of means of data transmission and access to devices remotely via the Internet. In contrast, there is an incredible acceleration in monitoring and stealing that data which could be highly significant to its owners. Unauthorized attacks on the data are continuous and inevitable which makes it vulnerable to intruder compromise. However, this threatens the security and integrity of that data while exchanging it between the connected parties. Highlighting the critical need for robust security measures to safeguard sensitive information during its exchange between connected parties, is a necessity and plays the main vital to protect data and establishing secure communication media. Shared data and exchanging could be the electronic healthcare record (EHR) and the patient's significant data. The significance of electronic healthcare records lies in their ability to collect important and sensitive patient data in a unified structure. Previously, it is known as electronic medical records EMRs. However, the absence of the EHR might lead to the loss of a lot of important information, medical reports, prescriptions, etc. [1, 2]. Therefore, given the foregoing, the protection of EHR during exchange and migration holds utmost importance. User authentication is one of the most famous security techniques that ensures the user identity of the connected parties via communication media. A

user "identifies himself by sending a query (q) to the system; the system authenticates his identity by calculation F(q) and ensuring that it equals the stored value w" [3, 4]. Despite of presenting a lot of approaches to secure the connection between users and healthcare servers such as session keys, but they still struggle with keeping anonymity [5]. Consecutively, access control is essential for the successful implementation of EHR in healthcare establishments, sometime healthcare establishments implement its EHR with no role-based access control. Where, it has been discovered in such institutions that all the system entities (doctors, nurses, admins, ...etc.) have access to all EHR. Resulting, controlling data access authorizations is not really possible [6, 7]. Although different encryption methods are used for EHR exchange over different healthcare institutions and their significant success in protecting transferred records. It still encounters various breaches of such records [8]. Relatedly, EHR migration between different servers of various healthcare institutions faces greater risks of attacks. It requires stronger data protection and security measures to mitigate potential vulnerabilities [9, 10]. In this paper, we present a secure technique for authenticating EHR patients in his healthcare institution server. Considering a scenario in which server A wants to migrate and share secured data with server B, the biggest challenge is the security of data migration. The data breaches by the attackers intercept data and the different types of well-known attacks (such as MIM, DoS, insider threats, and replies) [11]. However, our proposed scheme has been employed to overcome such challenges using ECC-encrypted/ decrypted EHR, additionally, it is faster compared with the other related schemes. Unauthorized access to data, another challenge faced the EHR sharing among different servers, as well as the verification of the identity of communicated servers using the digital certificate that has been issued by a trusted third-party healthcare server (TTPH). We relied on the asymmetric cipher, specifically the elliptical curve cipher (EEC), in the process of generating the key pair (public and private). Also using Elliptic Curve Digital Signature (ECDS) to issue and verify digital certificates for the servers that intend to share their EHR. For more security complexes, we use modern encryption and symmetric cryptography to generate shared keys among the entities of healthcare institutions that are utilized during the authentication phase. One of the important challenges is data corruption or loss during the migration process. The proposed scheme is conducted according to 4 phases, the setting phase is the first to manage and generate the keys and then to distribute them to the different servers, the registration phase is divided into two parts: The first part of the registration process focuses on enrolling healthcare servers with a trusted third party. The third party then issues digital identities to the servers for verification purposes during the EHR migration, while the second part involves registering the components of the healthcare system (patients and users) and assigning roles to them to maintain user privacy and prevent unauthorized access. This process is particularly useful for the login authentication phase. The third phase is the login authentication to preserve system entities and their privacy against unauthorized parties. The fourth phase is responsible for the EHR migration among healthcare servers securely. Anyhow, our scheme performs a comprehensive backup of data before migration to preserve data in case of loss. Another robustness technique used in our scheme is data validation and ensuring data integrity. Monitor server activity, and access logs for anomalies and suspicious behavior by checking who accesses data. When? And what did he/she do? That increased the security process to achieve a strong security level. The proposed scheme demonstrates strong resistance against known attack types (MITM, insider, linkability, DoS, phishing attack, replay, impersonal, and spoofing). This was proven through the use of formal security analysis, represented by the Scyther formal analysis tool, as well as informal analysis. The rest of the paper is organized as per followings: part II. reviews the related work. Part III. presents the proposed healthcare-secured schemes and is justified by security analysis in part IV. . While part V. represents comparing the relevant approaches and our contribution. Finally, part VI. presents the conclusion.

II. Related Work

To fulfill the demands of the EHR development environment, particularly concerning the security and integrity of data during sharing and transfer, it was imperative to explore alternative security measures that eliminate the reliance on outdated traditional methods, which may compromise data integrity. [12]. In the recent past, the password was relied upon to secure privacy. However, it has become easy to attack such approaches and violate the privacy [2, 13]. Therefore, the focus has been on alternative methods of protection in the past few years. For the reasons mentioned, many approaches presented to protect privacy and preserve data integrity in EHR systems. However, recent approaches focused on extending the factors of authentication such as two-factor authentication (2-FA), three-factor authentication (3-FA), and multi-factor–authentication (MFA) to increase the security measurements. The authentication factors could be mobile numbers, national identity cards (NIC), or biometrics (fingerprint, eye-print, and facial) [14–16]. We focused on some security techniques such as (cryptography type, digital signature, encryption/decryption algorithm, ...). Sudhakar et al. [17] presented a lightweight multi-server user authentication system based on a one-way hash function, It has been disclosed that their protocol lacks resistance against impersonation, insider, and Man-in-the-Middle (MITM) attacks. Manohara Pai

et al. [18] proposed a standard EHR framework using RSA signature, it can be used to improve the quality of healthcare services and reduce the cost of healthcare delivery. However, their scheme suffers from slowness during data transfer. Although, S. Shukla and S. Patel [19] suggested a new elliptic curve cryptography (ECC) based multi-factor authentication protocol for cloud computing to overcome weaknesses the proposed scheme by Y. Chen and J. Chen [20]. However, the paper identifies that Sah Shukla et al.'s protocol is prone to user linkability, replay, and denial-of-service (DoS) attacks. Also, Chen et al.'s protocol is vulnerable to user linkability and known session-specific temporary information (KSSTI) attacks. N. M. Hamed and A. A. Yassin [16] presented a consistent scheme of authentication to preserve the privacy of EHR reflecting the usage of modern security measurements. Anyway, their scheme uses a digital chameleon signature which is usually suffers from the limitation of usage and complexity a little. I. A. Obiri et al, presented a new scheme for sharing personal health records. Some advantages gained from their proposed scheme include enhancing privacy due to user privilege that includes special attributes. Anyhow, the scheme suffers from some drawbacks such as complexity due to the use of ABSC signature, and collusion risks colluding users combine their attributes [21]. F. Lalem et al proposed a scheme against various attacks, such as private key recovery, forgery, and replay attacks as well as its robustness against private key recovery and forgery attacks. Their approach depends on the huge prime number p and the difficulty of the discrete logarithm issue, it demonstrates that it is resilient and safe against these kinds of assaults. In contrast, the scheme lacks a balance between the complexity of security and the speed of communication during sharing of the EHR, due to using RSA and El Gamal digital signatures [22]. Table I shows the taxonomy comparison between our proposed scheme and some relevant ones depending on the digital signature type, security efficiency, complexity, and communication speed.

According to table I, we noticed that our proposed scheme is interesting in the balance between security and communication

III. Proposed Scheme

Our proposed scheme leverages the integration of health data to introduce a groundbreaking approach for privacy-preserving data exchange, authentication, and verification for the authorized entities over the healthcare institutions. The primary focus of our scheme is to ensure the secure sharing and collaboration of these data. This scheme comprises various components, including the trusted third-party healthcare institution $TTPH$ as a certificate authority CA, the health cloud which is called a hospital healthcare server HHS, and the patient P, who is the owner of the electronic health records EHR. Other important entities in EHR include users such as doctors, administrators, employees, etc.

Many patients or users need to access their servers to maintain their $EHRs$. For the aforementioned purpose, the previous symbols are followed by the letter i. For example, we refer to the different patients that are registered in the healthcare institution by P_i, and so on.

Following are the key preliminaries:

TTPH: is the Trusted Third-Party Healthcare used to generate certificates and keys for different hospital healthcare servers (HHS).

HHS: The Hospital Healthcare Server is the healthcare server of a specific hospital, used to generate encryption keys, manage keys, facilitate key exchange, establish key sessions, authenticate patients, and manage communication among the system components and other hospital servers.

The proposed approach revolves around four phases, as follows: (A) setting phase, (B) registration phase, (C) authentication phase, and (D) migration and verification phase.

A. Setting Phase

In our proposed scheme, the setting phase is a crucial level. The $TTPH$ is responsible for the key generation for the HHS. Depending on Elliptic Curve Cryptography ECC, the $TTPH$ achieves it through the steps below:

1. Choosing a and b that satisfy the desired security and efficiency properties, and a large prime number p, over $\mathbb{F}_p$ (e.g. $\mathbb{F}_p = \{0,1,2,\ldots,p-1\}$).

2. Using the equation of the elliptic curve:

$$E_p : y^2 = x^3 + ax + b \mod p,$$

where constants a and $b \in \mathbb{F}_p$. Additionally, it satisfies the equation $4a^3 + 27b^2 \neq 0$ (to ensure that the discriminant of the elliptic curve for a and b is non-zero, indicating a non-singular curve; for example, $a = 5376$ and $b = 2438$). The values of a and b specify the behavior of the curve, such as the number of points and symmetry. The pair (x,y) represents the coordinates of the curve that satisfy the equation.

3. Select a random private key $TTPH_{\mathrm{SK}_i}$ within the range of the curve's order n, where n represents the number of points on the curve, e.g., $\mathrm{TTPH}_{\mathrm{SK}} = 1234567890$. Selecting the private key randomly provides several benefits, it increases the security where it could be attacked by some techniques such as brute-force attacks or key guessing in case of selecting non-random. Another powerful feature is forward secrecy, which means in case of a compromised exact key, it does not compromise the security communication either in the past or the future.

TABLE I.
TAXONOMY OF THE AUTHENTICATION SCHEME

Schemes	Signature Type	Security	Efficiency	Complexity / Communication Speed
N. M. Hamed and A. A. Yassin [16]	Chameleon	Moderate (collision-prone)	High	Variable (depends on chameleon hash)
Sudhakar et al. [17]	-	Moderate (didn't meet the security measurements)	Moderate	High
Manohara Pai et al. [18]	RSA	High	Moderate	Moderate
Shukla and S. Patel [19]	ECC	Moderate (prone to user linkability)	Moderate	High
Y. Chen and J. Chen [20]	-	Moderate (prone to user linkability and DoS)	Moderate	High
I. A. Obiri et al. [21]	ABS	Moderate (attribute-based collusion attacks)	High	Low
F. Lalem et al. [22]	RSA & El Gamal	High	High	Low
Our Proposed Scheme	ECDSA	High	Moderate	High

4. Compute the corresponding public key $TTPH_{\mathrm{PK}_i}$ as follows:

$$TTPH_{\mathrm{PK}} = TTPH_{\mathrm{SK}} \times G,$$

where $G = G(X,Y)$ is the base point on the curve, and $X,Y \in G$.

Upon finalizing key generation, the $TTPH$ is ready to receive the request of HHS_i to register and obtain the digital certificates *DCs*. *ECC* key pairs play a vital role in providing security measures. The public and secret keys ensure secure communication. In addition, they perform the authentication process to prove the identity and ownership among the communicating parties.

B. Registration Phase:

This phase is divided into two parts. The first part is the registration of different healthcare establishments, HHS_i, in the $TTPH$. The second is the registration of the healthcare entities (patients, users, etc.) in their healthcare hospital server. Registering the HHS_i in the $TTPH$ and the entities to their servers are indispensable for secure communication. This phase ensures the access to the crucial data of patients is obtained by authorized parties. The certificate issuing provides the establishing trust and verifying identity among the communicating. HHS_i. Meanwhile, registering the entities in HHS_i ensures privacy preservation and prevents unauthorized parties to get connected to the exact healthcare institution.

1) Registration of the HHS_i into the $TTPH$

For secure data exchange among different healthcare establishments, it is necessary that the HHS_i of these establishments register in a trusted third-party healthcare institution $TTPH$. To manage this phase, we propose that HHS_i should have a digital certificate (DC_i) issued by a certificate authority *CA*. The *CA* in our proposed scheme is $TTPH$. The steps below summarize the registration and issuance of the digital certificate *DC*.

HHS_i Side

1. Depending on *ECC*, the HHS_i generates its key pair (private key $HHS_{i_{\mathrm{SK}}}$) and the corresponding public key ($HHS_{i_{\mathrm{PK}}}$).

2. Creating a certificate signature request (CSR_{HHS_i}), this includes the important information which is typically called *CSR* data, such as ID (ID_{HHS_i}), Password (PW_{HHS_i}), email, server's domain name, organization details, etc. The CSR_{HHS_i} is signed using $HHS_{i_{SK}}$. We denote the *CSR* data by $HHS_{i_{CSR\_D}}$. The following steps outline the process of obtaining the signature:

   (a) H_{CSR_D}=$hash(HHS_{i_{CSR\_D}})$; using a cryptographic hash function, such as SHA-256.

   (b) Generates a random number r, where $r \in$ curve's order(n).

   (c) R=$r \cdot G$.

   (d) $x_R = R_x \mod p$; where p is a large prime number in the curve.

   (e) $S = \left(r^{-1} * \left(H_{CSR\_D} + x_R * HHS_{i_{SK}}\right)\right) \mod n$, where r^{-1} is the modular multiplicative inverse of r modulo n.

   (f) $Sig_{HHS_i} = (x_R, S)$.

3. Submitting the CSR_{HHS_i}, including $HHS_{i_{PK}}$, Sig_{HHS_i}, and $HHS_{i_{CSR\_D}}$ to $TTPH$ once ready, through a secure channel such as secure email.

$TTPH_i$ **Side**

The $TTPH$ receives the CSR_{HHS_i}, which includes (Sig_{HHS_i}, $HHS_{i_{CSR\_D}}$, and $HHS_{i_{PK}}$). The $TTPH$ creates a digital certificate and a digital signature using its private key. To manage this phase, the $TTPH$ performs the following steps:

1. Using the CSR_{HHS_i} parameters (Sig_{HHS_i}, $HHS_{i_{CSR\_D}}$, and $HHS_{i_{PK}}$).
2. CSR_{HHS_i} Verification: The following are the verification steps:
   (a) Compute $Q = Sig^{-1}_{HHS_i}\ (modn)$.
   (b) Compute $C1 = (hash(HHS_{i_{CSR\_D}}) * Q)\ modn$.
   (c) Compute $C2 = (x_R * Q)\ (modn)$.
   (d) Compute the point $x'_R = (C1 * G) + (C2 * HHS_{i_{PK}})$.
   (e) If x'_R matches x_R, the digital signature is considered valid. Otherwise, the digital signature is invalid.
3. Issuing a Digital Certificate for HHS_i: After successful verification of CSR_{HHS_i}, a certificate is issued. The certificate template includes $HHS_{i_{CSR\_D}}$, $HHS_{i_{PK}}$, $Cert_{issuedate}$, $Cert_{expirationdate}$, $TTPH_{CA}$, and $TTPH_{PK_i}$.
   (a) $Cert_{HHS_i} = HHS_{i_{CSR\_D}} || HHS_{i_PK} || Cert_{\text{Start date}} || Cert_{Expirationdate} || TTPH_{CA}, TTPH_{PK_i}$.
   (b) Calculate $HCert_{HHS_i} = hash(Cert_{HHS_i})$.
   (c) Generate a random number $r', r' \in$ curve's order (n) .
   (d) Calculate $R' = r' * G$.
   (e) Calculate $nx_R = R'_x\ mod p$.
   (f) $S' = (r'^{-1} * (HCert_{HHS_i} + nx_R * TTPH_{SK_i}))modn$, where r'^{-1} is the modular multiplicative inverse of r' modulo n.
   (g) $Sig_{TTPH} = (nx_R, S')$.
   (h) $TTPH$ sends $Cert_{HHS_i}, Sig_{TTPH}, TTPH_{PK_i}$ to HHS_i using a secured channel.

2) Registration of the EHR into their servers

Difference entities of EHR should register in the exact healthcare institution, these entities could be patients or users. In context, the user might be (employees, doctors, or administrator). We proposed to portion this phase into two parts: patient registration and user registration.

1. **Patient Registration:**
   Sensitive and crucial information about the patient should be sent to its healthcare establishment server (HHS_i) such as the patient's name (ID_p), patient password (PW_p). Additionally, necessary information like patient address (Add_p), patient email address ($Mail_p$), etc. The patient sends the tuple (ID_p, PW_p, Add_p, $Mail_p$) to HHS_i. The HHS_i performs the anonymous steps to generate patient ID and password:
   - $ID'_{P_i} = hash(P_i)$.
   - $PW'_{P_i} = hash(PW_{P_i} || ID_{P_i})$.
   - Generate a random number as a shared key (SHK_{P_i}).
   - Update the EHR information of the patient P_i.
   - Send SHK_{P_i} to P_i to use in the next phases.
2. **User Registration:**
   Doctors, employees, and administrators are the entities of the exact healthcare establishment, all these entities must register with such establishment to be able to manage the EHR according to their privilege could be specified by their job. For this reason, the user ((U_i)) sends his request to the server (HHS_i) which includes ((ID_{U_i}), (PW_{U_i}), ($Mail_{U_i}$),... etc.). Upon receiving the user request, (HHS_i) performs the following:
   - ($ID'_{U_i} = hash(ID_{U_i})$).
   - ($PW'_{U_i} = hash(PW_{U_i} || ID_{U_i})$).
   - Generate a random number as a shared key SHK_{U_i}.
   - Store the updated information for (U_i).
   - Distribute the privilege according to the job of U_i.
   - Send (SHK_{U_i}) to (U_i) to use during local data exchange.

C. Login Authentication Phase

In order to provide safe access in a healthcare institution, there must be secured access to crucial data and information. However, the authentication process guaranteed a secure login. Moreover, it prevents the access of unauthorized persons, protects the user privacy-preserving, and establishes accountability. The different entities of the system must follow the login steps authentically. . Our proposed scheme includes login and authentication for the patient (P_i) and user (U_i) individually.

1. **Patient Login and Authentication:**
   The patient P_i initiates the login process by providing their ID (ID_{P_i}) and password (PW_{P_i}).
   $P_i \rightarrow HHS_i$

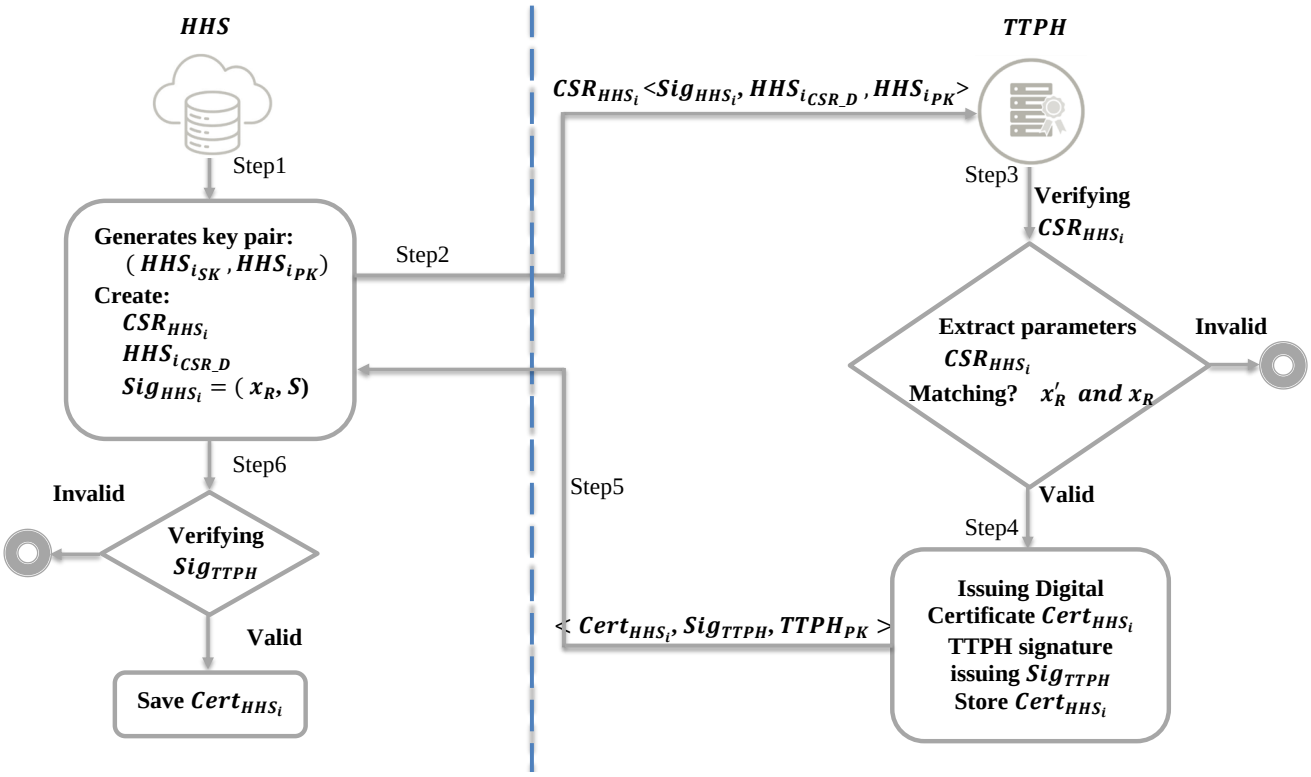


Fig. 1. Registration of HHS_i and issuing $Cert_{HHS_i}$ in TTPH

- Generate an integer random number $r_i \in \mathbb{Z}_n^*$.
- Calculate $ID''_{P_i} = hash(ID''_{P_i})$ and $PW''_{P_i} = hash(hash(PW_{P_i} || ID_{P_i}) \oplus r_i)$. Note the significance of this step in generating a password for each patient login request.
- Calculate $r'_i = r_i \oplus SHK_{P_i}$.
- Send the query $\{R_{P_i} = \langle ID''_{P_i}, PW''_{P_i}, r'_i \rangle\}$ to HHS_i.

$HHS_i \rightarrow P_i$

- Upon receipt of the patient's query (R_{P_i}), HHS_i calculates $r_i'' = r_i' \oplus SHK_{p_i}$.
- Restores PW'_{P_i}, then calculates $PW'''_{P_i} = hash(PW'_{P_i} \oplus r_i'')$.
- PW''_{P_i} is compared with PW'''_{P_i}; if the result matches, the patient is granted the privilege to log in to the system and update their EHR_i. Otherwise, access is denied.
- Upon successful patient identity verification and login, the session key is maintained as $(r_i \oplus SHK_{p_i})$.

2. **User login and Authentication**
The user U_i initiates the login process by providing their ID (ID_{U_i}) and password (PW_{U_i}).
$U_i \rightarrow HHS_i$

- Generate an integer random number $r_i \in \mathbb{Z}_n^*$.
- Calculate $ID''_{U_i} = hash(ID''_{U_i})$ and $PW''_{u_i} = hash(hash(PW_{U_i} || ID_{U_i}) \oplus r_i)$. Note the significance of this step in generating a password for each user login request.
- Calculate $r'_i = r_i \oplus SHK_{U_i}$.
- Send the query $\{R_{U_i} = \langle ID''_{U_i}, PW''_{U_i}, r'_i \rangle\}$ to HHS_i.

$HHS_i \rightarrow U_i$

- Upon receipt of the user's query (R_{P_i}), HHS_i calculates $r''_i = r'_i \oplus SHK_{U_i}$.
- Restores PW'_{U_i}, then calculates $PW'''_{U_i} = hash(PW'_{U_i} \oplus r''_i)$.
- PW''_{U_i} is compared with PW'''_{U_i}; if the result matches, the user is granted the privilege to log in to the system and update their EHR_i. Otherwise, access is denied.
- Upon successful user identity verification and login, the session key is maintained as $r_i \oplus SHK_{U_i}$.

D. Migration and Verification Phase

Data exchange and data migration are both important processes for sharing data securely in managing the EHR. Data exchange plays a vital role in securely sharing data among EHR entities, ensuring the privacy and preservation of the entities. On the other hand, data migration involves sharing crucial EHR data among different servers ($HHS_1, HHS_2, \ldots, HHS_n$). Sometimes, a patient needs to undergo certain medical tests outside their healthcare institution, such as (HHS_k). In such cases, the external healthcare institution requests verification of the patient's identity and retrieval of their EHR. This entails the migration of the patient's EHR from their healthcare institution (HHS_i) to HHS_k. The previous scenario has been addressed by our proposed scheme as follows:

$HHS_k \rightarrow HHS_i$**:**

- Restores $Sig_{TTPH}(Cert_{HHS_k})$, the certificate of HHS_k which has been signed by $TTPH$.
- Obtains the patient ID (ID_{P_i}) of patient P_i.
- Sends a request to HHS_i including $(Sig_{TTPH}(Cert_{HHS_k}), ID_{P_i})$.

$HHS_i \rightarrow TTPH$**:**

Upon receiving the request $(Sig_{TTPH}(Cert_{HHS_k}), ID_{P_i}), HHS_i$ must verify HHS_k through the $Cert_{HHS_k}$ as well as perform patient identity verification.

1. **Verification of HHS_k:**
   HHS_i sends the request $Sig_{TTPH}\left(Cert_{HHS_k}\right)$ to $TTPH$.

   $TTPH \rightarrow HHS_i$**:**

   - Receives Sig_{HHS_k} from HHS_i, which includes $Cert_{HHS_k}$ and the public key HHS_{k_PK}.
   - Extracts parameters of Sig_{HHS_k}, (x_{R_k}, S_k).
   - Extracts $Cert_{HHS_k}$ and $HHS_{k_{PK}}$.
   - Restores $TTPH_{PK_K}$, the public key of $TTPH$, which signed $Cert_{HHS_k}$ during the registration phase. This is utilized to verify the certificate.
   - Verifies $Cert_{HHS_k}$:
     - Computes: $H_k = hash\left(Cert_{HHS_k}\right)$
     - Computes $Q = \left(Sig_{HHS_k}\right)^{-1} \ (mod\ n)$
     - Computes $C1 = (H_k * Q)\ mod\ n$
     - Computes $C2 = (x_{R_k} * Q)\ mod\ n$
     - Computes the point $x'_R = (C1 * G) + (C2 * HHS_{k_{PK}})$
     - If x'_R matches x_R, the digital signature is valid. Otherwise, it is invalid.
       Note: x_R is the x-coordinate calculated during the registration phase and certificate issuance.
   - After successful verification, $TTPH$ replies to HHS_i, verifying the authenticity and integrity of HHS_k's digital certificate and establishing trust for secure communication and EHR_i migration.

2. **Verification of $\mathbf{P}_i$:**
   At this level, HHS_i checks the patient ID ID_{p_i} sent by HHS_k to verify the identity and authenticity of the patient P_i.

   - Computes $ID''_{P_i} = hash\,(ID_{p_i})$
   - Compares ID''_{P_i} with ID'_{P_i}, which was issued during the registration phase. If they match:
     - Sends a confirmation link to (P_i) using the patient's registered email.
     - the (P_i) must confirm the link within the time limit set by HHS_i. If there is no response, the connection is canceled.
     - HHS_i receives the confirmation from P_i, ensuring the patient's identity is verified.
   - After successfully verification of $HHS_k and P_i, HHS_i$ migrate the EHR_i of the patient to start recording a report of a new treatment served by an outer healthcare institution (HHS_k).

Fig. 2 displays the migration phase with two main parts of verification (communicated healthcare servers and patients that request a service from outside healthcare institutions).

IV. Security Analysis

From a security perspective, we conducted a two-part security analysis of the proposed protocol:

A. Formal Security Analysis

We employed the Scyther tool, a well-established tool for automated analysis and verifying the security of cryptographic protocols against well-known attacks. This analysis assessed the protocol's robustness against potential vulnerabilities during data communication. The tool facilitates the automated, unbounded, and symbolic analysis of security protocols. It offers expressive languages to precisely define protocols and adversary models, enabling comprehensive security evaluations [23, 24]. In the following, we showed the result of the migration phase using Syther tool. Fig. 3 and Fig. 4 respectively display the result of the proposed migration phase. Fig. 3 we implemented our protocol with some weaknesses to see the result of attacks. As you can see there were some attacks when the server HHS_k requests EHR to be migrated by the server HHS_i. In the following, Fig. 4 shows the power of the proposed scheme during the migration phase. The result shows the verification of our proposed scheme. While

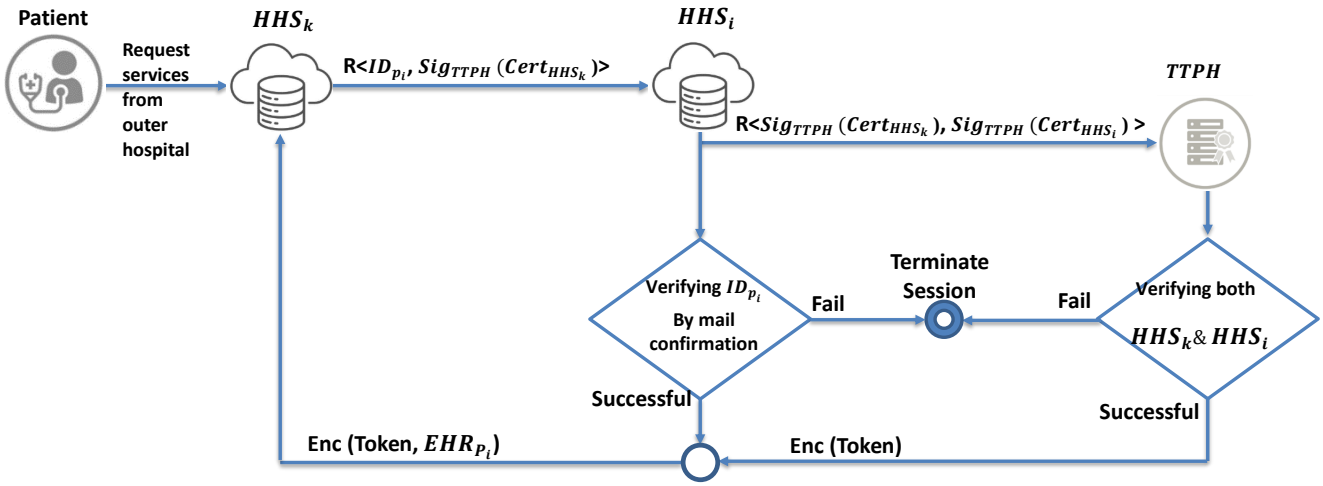


Fig. 2. Migration Phase through Cloud Servers

Fig. 5 below shows the verification of digital certificates by $TTPH$ among healthcare servers during communications and migrates the EHR among servers. Where the result shows the verification and power of the scheme (no attacked, reachable failed, and verification).

B. Informal Security Analysis

We will complement the formal analysis with an informal security analysis detailed in this paper. This section employs informal security analysis to demonstrate the resilience of the proposed scheme against common attacks, including Man-in-the-Middle (MITM) attacks, DoS attacks, replay attacks, and insider attacks. Furthermore, the analysis highlights the scheme's security strengths, such as user anonymity and privacy, mutual authentication, and session key agreement. **Key management session and user anonymity:** to safeguard patient and user privacy, we achieved anonymity. The servers of healthcare institutions are responsible for generating and distributing symmetric encryption keys to patients and authorized users, including doctors. Additionally, these servers anonymously assign unique identifiers and passwords to system entities, further enhancing security by reducing the likelihood of compromising user credentials. For example, our proposed scheme explains how the ID and Password stored anonymity to the HHS.

$$ID''_{P_i} = hash(ID'''_{P_i})$$

$$PW''_{P_i} = hash\,(hash(PW_{P_i} \parallel ID_{P_i}) \oplus r_i)$$

$$r'_i = r_i \oplus SHKp_i$$

In addition to this point, the proposed scheme achieves a high level of anonymity, effectively preventing linkability and insider attacks as discussed earlier, as well as the EHR shred and maintaining securely by encrypting using ECC. Mutual Authentication: our proposed scheme achieved mutual authentication to protect the system entities' identifications from spoofing attacks by intruders and attackers who try to spoof the identification to gain unauthorized access. Moreover, it prevents impersonation and man-in-the-middle (MiTM) attacks, when someone steals your login details and tries to impersonate any system's entity. Below is mutual authentication proof between $TTPH$ and HHS_i certificates and the other side patient P_i and HHS_i:

- HHS_i **to** $TTPH \leftrightarrow TTPH$ **to** HHS_i**:**
  - HHS_i**:** sends request $R_i < Sig_{TTPH}(Cert_{HHS_i}) >$, the request is supported by the digital certificate of HHS_i with the signature of $TTPH$. The $TTPH$ verifies the identity using a digital certificate, in this case, HHS_i certificate.
  - $TTPH$**:** calculates the point $x'_R = (C1 * G) + (C2 * HHS_{k_{pk}})$. If x'_R matches x_R, the digital signature is considered valid and $HHSi$ is recognized as the authorized server.
  - HHS_i**:** receives the reply from $TTPH$ and verifies TTPH's certificate.
- HHS_i to $P_i \leftrightarrow P_i$ to HHS_i

  HHS_i checks (ID_{P_i}), which is received by P_i request to verify the identity and authenticity of the patient P_i.

  a. Computes $ID''_{P_i} = hash(ID_{P_i})$
  b. Compares ID''_{P_i} with ID'_{P_i}
  c. Sends $Enc(token)$ to P_i by confirmation mail

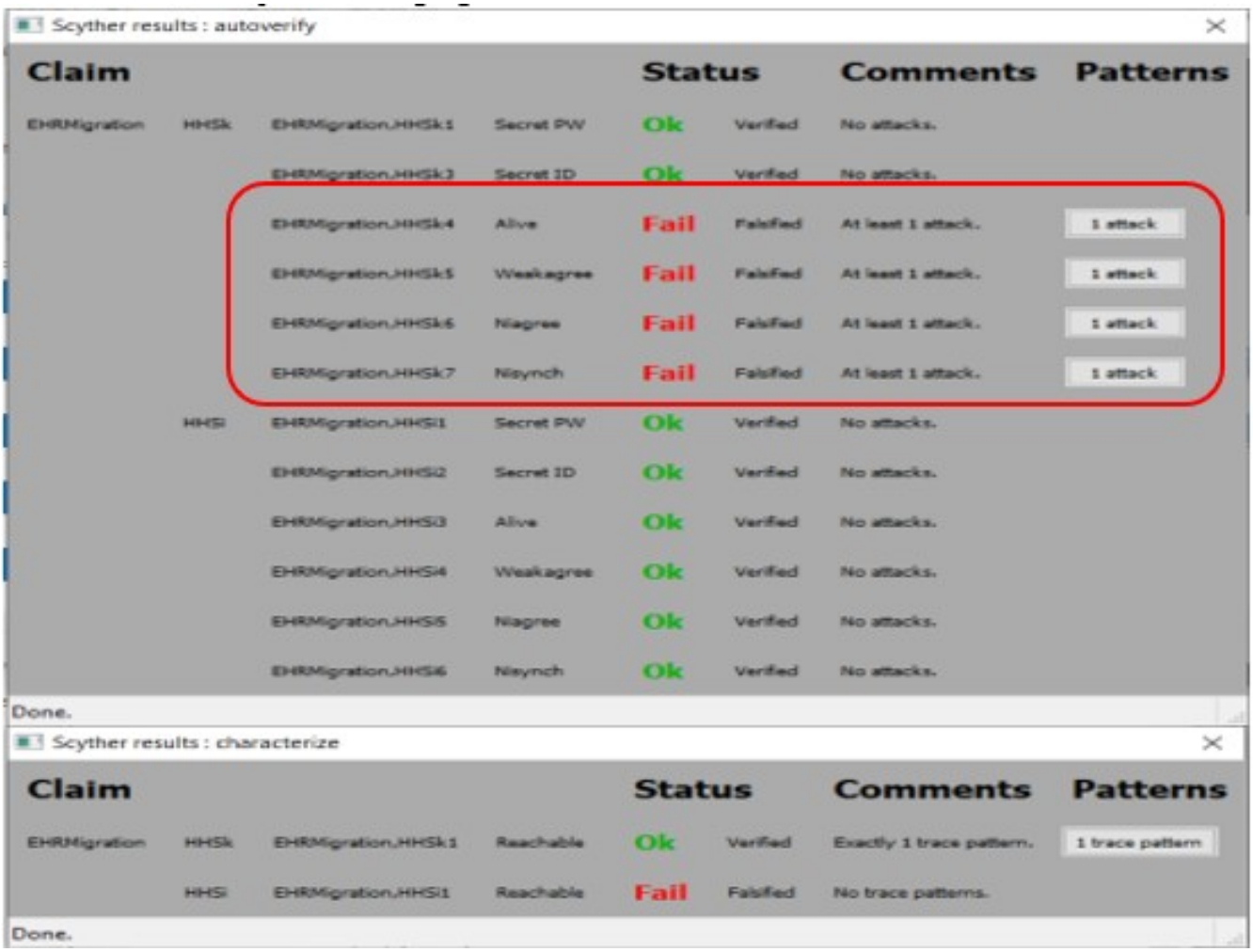


Fig. 3. Weakness of traditional migration system in communication between server HHS_k and HHS_i

- P_i: receives Enc(token), then decrypts the token and verifies the ID of TTPH.

DoS attacks: our proposed scheme enjoyed a high level of scalability and this prevents denial-of-service (DoS) attacks to cripple a system by overwhelming it with traffic. Verifying the identifications of the sender, receiver, and other users and preventing unauthorized persons from flooding the system with any fake overwhelms.

V. Comparing The Relevant Approaches and our Contribution

To evaluate the efficiency of the proposed scheme, we analyze its temporal complexity, which reflects the computational cost associated with its execution. Table II describes the scales considered to evaluate the computational efficiency of our technique relative to the most relevant and significant similar schemes through a cost comparison. However, the two operations $T_{\oplus}$ and $T_{\|}$ not assessed according to their extremely short time [25, 26].

Table III shows the processing time per millisecond of computational cost. Building upon existing work, our proposed scheme offers a balancing approach that achieves the security protocol with less processing time compared with the relevant approach

$$6T_h + 2T_{S_G} + 6T_{\oplus} + 4T_{\|} \approx 7.7092\,\text{ms}$$

. Linking with Table I our proposed scheme is interested in high security as well.

To increase the robustness of our proposed scheme, especially in the migration phase, another comparison was made according to the communication cost. Which is considered a basic measure for security analysis. However, the message

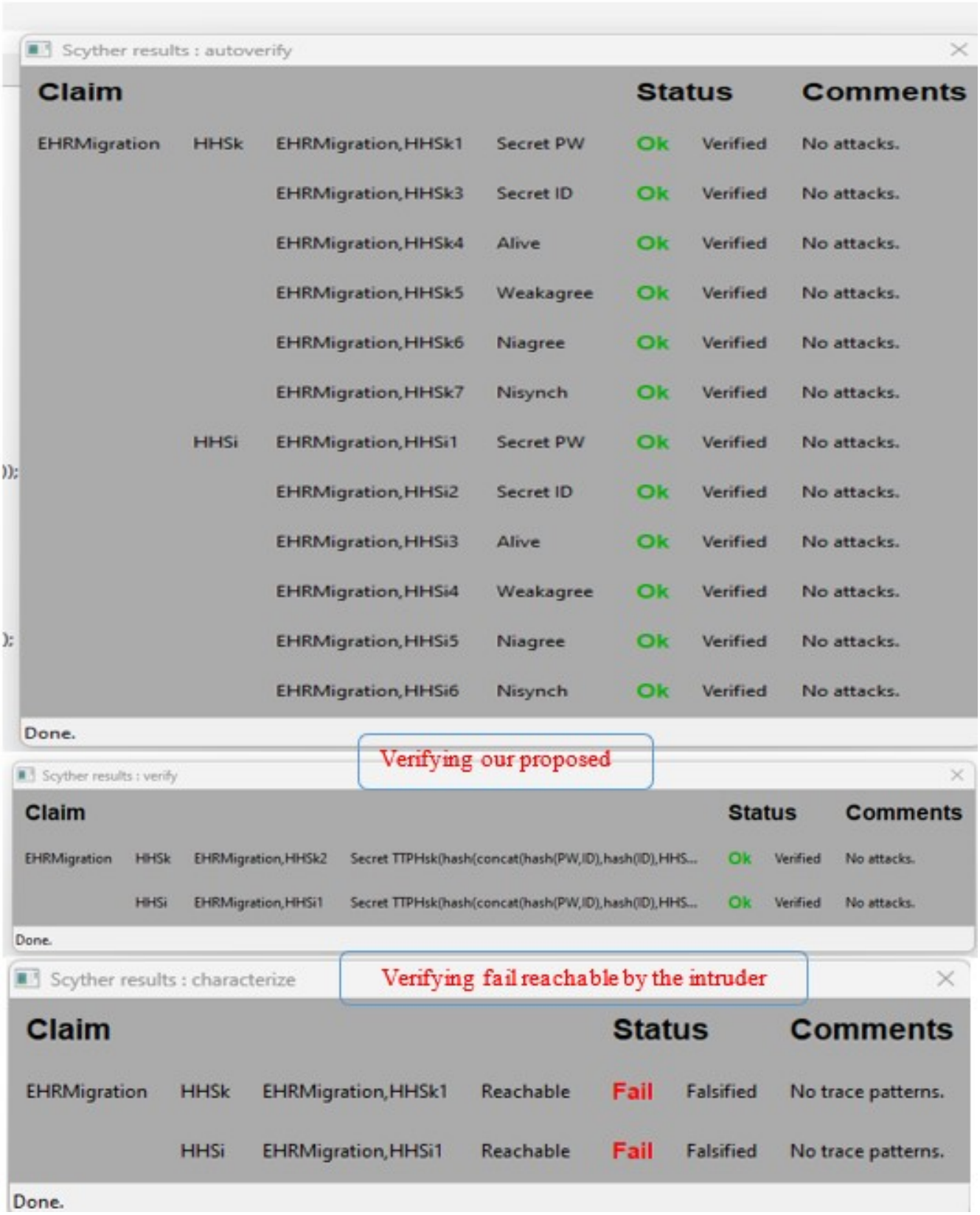


Fig. 4. shows the power of the proposed Migration system verification

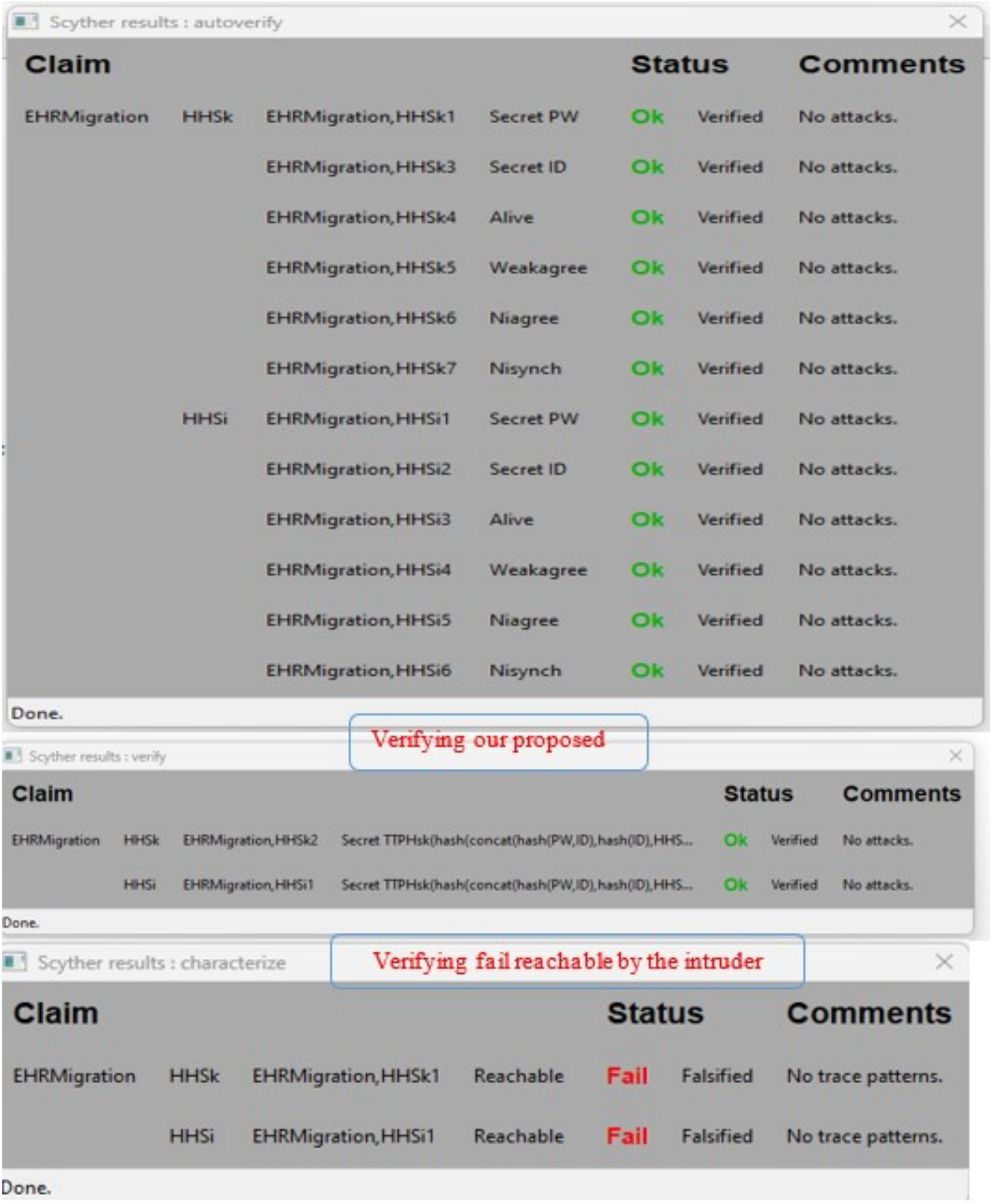


Fig. 5. *TTPH* Verified the digital certificates of servers

TABLE II.
DISPLAYS THE NOTATION FOR FUNCTIONS AND ESTIMATED EXECUTION TIME.

Sample	Description	Estimated Time (ms)
T_h	The processing time of the crypto hash function	0.0023
T_E	The processing time for a symmetric encryption function.	0.0046
TS_G	Signature processing time	3.85
$T_{\oplus}$	XOR operation processing time	NA
$T_{\|\|}$	Processing time of Concatenation operation	NA

passing through EHR sharing and login authentication consists of some different variables. We supposed the identity size is 32-bit, the hash value's size is 160 bits, the size of the RSA digital signature is 512 bits, the El Gamal digital signature is 512 bits, the Chameleon digital signature is 512 bits, and the size of the ECDSA signature is 112 bits [27, 28]. The length of digital signatures taken according to the moderate security, where the 512-bit key length in (RSA, Chameleon, and El Gamal) corresponds to the 112-bit key length of ECDSA. Table IV shows the communication cost comparison with the relevant.

Fig. 6a shows the computational and Fig. 6b shows the communication costs compared to the existing related schemes. As can be seen, our proposed method achieves noticeably higher computational and communication efficiency. This is reflected in the figures, which show the lowest costs and shorter message length among the comparable systems.

TABLE III.
COMPUTATION COSTS OF DIFFERENT SCHEMES

Schemes	Computation Cost	Total Cost (ms)
N. M. Hamed and A. A. Yassin [16]	$8T_H + 4T_E + 4T_D + 6T_{\oplus} + 2T_{S_G}$	≈ 7.7552
Manohara Pai et al. [18]	$16T_H + 4T_{S_G} + 10T_{\oplus}$	≈ 15.4368
Shukla and S. Patel [19]	$10T_H + 4T_{S_G} + 8T_{\oplus}$	≈ 15.423
F. Lalem and et al. proposed [22]	$8T_H + 8T_{S_G} + 8T_{\oplus} + 2T_{\|\|}$	≈ 30.8184
Our Proposed Scheme	$6T_h + 2T_{S_G} + 6T_{\oplus} + 4T_{\|\|}$	≈ 7.7092

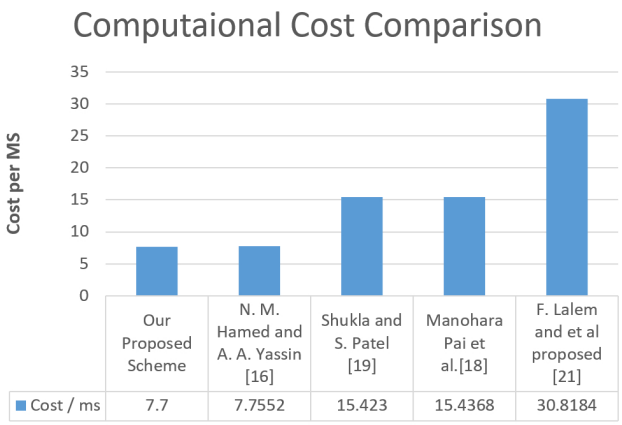
TABLE IV.
COMMUNICATION COST COMPARISON: OURS VS. RELATED WORKS

Schemes	Message Length (bits)	Total Message
N. M. Hamed and A. A. Yassin [16]	2272	3
Manohara Pai et al. [18]	4800	5
Shukla and S. Patel [19]	1664	4
F. Lalem and et al. proposed [22]	4416	3
Our Proposed Scheme	1248	3

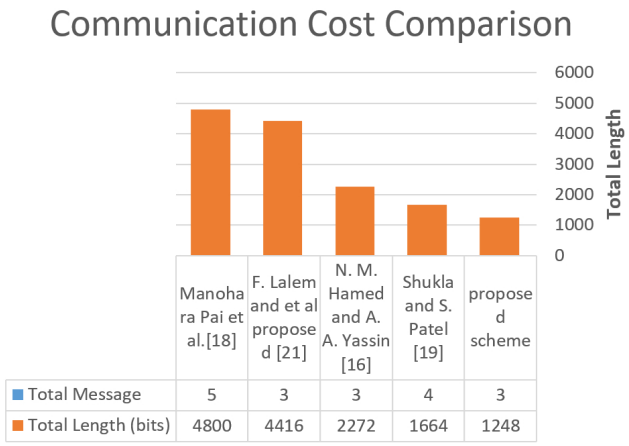
Based on the previous discussion, our proposed scheme exhibits a wonderful balance between the computational complexity of the encryption performance and the speed of communication leading to a secure and efficient solution for data communication.

VI. CONCLUSION

In conclusion, this research addresses the challenge of balancing privacy and security with efficiency in EHRs represented to manage the EHR migration among different hospital cloud servers. We propose a login authentication approach that utilizes two steps, the first step is patient login authentication, and the second is user login authentication such as doctor and admin, and leverages Scyther, a formal security tool, to validate its security against various attacks. Our approach demonstrably achieves efficient EHR migration with verification of the certificates of the hospital's healthcare servers (HHS) utilizing ECDSA and signature of the certificates. Whereas, this obtains a communication speed and high-security level as displayed in the security analyses phase. Consistently, the informal security analysis ensures well-known attack resistance such as man-in-the-middle (MITM) attacks, DoS attacks, replay attacks, and insider attacks due to key management sessions, anonymity, and mutual authentication. This proposed system offers a well-balanced solution for secure and efficient EHR migration in healthcare. However, to increase the security level and enhance threat detection, some techniques could be taken into consideration such as a machine learning machine learning model to identify and respond to new security threats dynamically. The integration of blockchain and fog computing will be an excellent technique as well.



(a)



(b)

Fig. 6. Computational cost.

REFERENCES

[1] A. S. Radwan, A. A. Abdel-Hamid, and Y. Hanafy, "Cloud-based service for secure electronic medical record exchange," in *2012 22nd International Conference on Computer Theory and Applications (ICCTA)*, pp. 94–103, IEEE, 2012.

[2] M. Yıldırım and I. Mackie, "Encouraging users to improve password security and memorability," *International Journal of Information Security*, vol. 18, pp. 741–759, 2019.

[3] A. Ometov, S. Bezzateev, N. Mäkitalo, S. Andreev, T. Mikkonen, and Y. Koucheryavy, "Multi-factor authentication: A survey," *Cryptography*, vol. 2, no. 1, p. 1, 2018.

[4] J. C. Mandel, D. A. Kreda, K. D. Mandl, I. S. Kohane, and R. B. Ramoni, "Smart on fhir: a standards-based, interoperable apps platform for electronic health records," *Journal of the American Medical Informatics Association*, vol. 23, no. 5, pp. 899–908, 2016.

[5] L. Zhang, Y. Zhang, S. Tang, and H. Luo, "Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2795–2805, 2017.

[6] R. Ssembatya and A. V. Kayem, "Secure and efficient mobile personal health data sharing in resource constrained environments," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pp. 411–416, IEEE, 2015.

[7] T. Tsegaye and S. Flowerday, "A clark-wilson and ansi role-based access control model," *Information & Computer Security*, vol. 28, no. 3, pp. 373–395, 2020.

[8] O. Ajayi, M. Abouali, and T. Saadawi, "Secure architecture for inter-healthcare electronic health records exchange," in *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–6, IEEE, 2020.

[9] M. Rajakumar and S. Thavamani, "Migration of ehealth cloud to sehealth cloud," in *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, pp. 1–4, IEEE, 2021.

[10] S. Saleh, N. El Arnaout, L. Abdouni, Z. Jammoul, N. Hachach, and A. Dasgupta, "Sijilli: a scalable model of cloud-based electronic health records for migrating populations in low-resource settings," *Journal of Medical Internet Research*, vol. 22, no. 8, p. e18183, 2020.

[11] M. De Vincenzi, G. Costantino, I. Matteucci, F. Fenzl, C. Plappert, R. Rieke, and D. Zelle, "A systematic review on security attacks and countermeasures in automotive ethernet," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–38, 2024.

[12] R. Sivan and Z. A. Zukarnain, "Security and privacy in cloud-based e-health system," *Symmetry*, vol. 13, no. 5, p. 742, 2021.

[13] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE access*, vol. 7, pp. 41678–41689, 2019.

[14] H. Shahriar, H. M. Haddad, and M. Farhadi, *Assessing HIPAA compliance of open source electronic health record applications*, pp. 995–1011. IGI Global, 2022.

[15] M. Jayabalan and T. O'Daniel, "A study on authentication factors in electronic health records," *Journal of Applied Technology and Innovation*, vol. 3, no. 1, 2019.

[16] N. M. Hamed and A. A. Yassin, "A secure and authentication scheme to preserve the privacy of electronic health records in the healthcare system," in *2022 Iraqi International Conference on Communication and Information Technologies (IICCIT)*, pp. 32–37, IEEE, 2022.

[17] T. Sudhakar, V. Natarajan, M. Gopinath, and J. Saranyadevi, "An enhanced authentication protocol for multi-server environment using password and smart card," *Wireless Personal Communications*, vol. 115, pp. 2779–2803, 2020.

[18] M. M. Pai, R. Ganiga, R. M. Pai, and R. K. Sinha, "Standard electronic health record (ehr) framework for indian healthcare system," *Health Services and Outcomes Research Methodology*, vol. 21, no. 3, pp. 339–362, 2021.

[19] S. Shukla and S. J. Patel, "A novel ecc-based provably secure and privacy-preserving multi-factor authentication protocol for cloud computing," *Computing*, vol. 104, no. 5, pp. 1173–1202, 2022.

[20] Y. Chen and J. Chen, "A secure three-factor-based authentication with key agreement protocol for e-health clouds," *The Journal of Supercomputing*, vol. 77, pp. 3359–3380, 2021.

[21] I. A. Obiri, Q. Xia, H. Xia, E. Affum, S. Abla, and J. Gao, "Personal health records sharing scheme based on attribute based signcryption with data integrity verifiable," *Journal of Computer Security*, vol. 30, no. 2, pp. 291–324, 2022.

[22] F. Lalem, A. Laouid, M. Kara, M. Al-Khalidi, and A. Eleyan, "A novel digital signature scheme for advanced asymmetric encryption techniques," *Applied Sciences*, vol. 13, no. 8, p. 5172, 2023.

[23] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols: Tool paper," in *International conference on computer aided verification*, pp. 414–418, Springer, 2008.

[24] S. Meier, B. Schmidt, C. Cremers, and D. Basin, "The tamarin prover for the symbolic analysis of security protocols," in *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*, pp. 696–701, Springer, 2013.

[25] Y. Salami, V. Khajehvand, and E. Zeinali, "E3c: a tool for evaluating communication and computation costs in authentication and key exchange protocol," *arXiv preprint arXiv:2212.03308*, 2022.

[26] 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.

[27] Z. Sann, T. Soe, and K. Nwe, "Comparison of public key cryptography in different security level," *International Journal of Recent Development in Engineering and Technology*, vol. 8, no. 12, 2019.

[28] A. A. Imem, "Comparison and evaluation of digital signature schemes employed in ndn network," *arXiv preprint arXiv:1508.00184*, 2015.