Computer Software and Media Application (2023) Volume 6 Issue 1
doi: 10.24294/csma.v6i1.3062

ORIGINAL RESEARCH ARTICLE

PDHSMICK: A partial key-based secure EHRs with distributed cloud for healthcare systems applying MI-CRYSTALS-Kyber

Prathima Subramanian*, R. Durga

School of Computing Sciences, Department of Information Technology, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai 600117, India

*** Corresponding author:** Prathima Subramanian, prathismanian@gmail.com

ABSTRACT

Healthcare, which raises several challenges regarding security and privacy, is the largest sector of society dealing with patients' health data. Numerous approaches have been developed regarding healthcare; however, these approaches lead to high response time and leakage of transaction privacy. Thus, this work proposes PDHSMICK, a partial key-based secure EHRs with distributed cloud for Healthcare Systems applying Mimetic Interpolation-based CRYSTALS-Kyber (MI-CRYSTAL-Kyber). Primarily, the patient and doctor register with the system. After that, the patient logs in and books an appointment with the doctor. Then, consultation is done, and their details are sent to the Third-Party Auditor (TPA), where they store the information in the log file. Afterwards, the patient verifies the log file and then uploads the information to the concerned Cloud Server (CS). During uploading, the files are split, and the attributes from the files and Distributed Cloud Server (DCS) are extracted. Then, by utilizing Interpolation-based Pearson Correlation Coefficient (I-PCC), a particular DCS is selected. After that, for authorization purposes, a hash code and digital signature are generated by utilizing Exponential-based Extensible Output Functions (EX-EOF) and Double Mod-Digital Signature Algorithm (DM-DSA). Concurrently, the files are encrypted by employing MI-CRYSTAL-Kyber techniques. Lastly, the doctor logs on to the system, and hash code and digital signature verification are performed. The system allows the doctor to download the data if the verification matches. Also, the experimental assessment was validated, which shows the proposed technique's efficacy.

Keywords: Mimetic Interpolation-based CRYSTALS—Kyber (MI-CRYSTAL-Kyber); Third Party Auditor (TPA); Distributed Cloud Server (DCS); Interpolation-based Pearson Correlation Coefficient (I-PCC); Exponential-based Extensible Output Functions (EX-EOF); Double Mod-Digital Signature Algorithm (DM-DSA); healthcare data

ARTICLE INFO

Received: 20 October 2023
Accepted: 29 October 2023
Available online: 15 December 2023

1. Introduction

In the current world, there is an active rise in elderly patients as well as chronically ill patients who require a remote-control system for health monitoring. Owing to various challenges like travel time and queuing, in addition to the risk of contracting several viruses by these patients while journeying through the contaminated surroundings, it is extremely difficult to transfer patients from their house to healthcare centres for periodic check-ups in the past few decades[1]. Globally, hospitalization, as well as patient care, rises with the increase in the number of patients. The reason behind this is the inaccurate dosage, incorrect medications, plus delayed treatment. Hence, Healthcare Systems (HCS) models were developed that help in cases like these and reduce the load on staff and other healthcare professionals[2]. These systems aid in enhancing patient quality of life, improving patient outcomes, increasing the overall efficiency of e-healthcare services, managing health information and

patient data, increasing collaboration, and decreasing costs[3]. In the HCS, patient data is stored in the CS. In the healthcare business's digital transformation, the next turning point is the cloud. This technology's complete advantage is taken by hospitals and other non-Information Technology (IT)-specific enterprises[4]. The data that is saved in the CS is subjected to privacy-leakage and gets attacked by unapproved users and assaulters. Many security attacks are vulnerable to the HCS. Thus, the major challenges in this network are security and privacy threats[5]. In any system, the first level of effective security is encompassed by both authentication and authorization mechanisms[6]. Trusted cryptography like a public key or else symmetric key cryptography and Elliptical Curve Cryptography (ECC)-based cryptography are common encryption techniques that are wielded to encrypt health data for security purposes. Further, data confidentiality is ensured by encryption; it could also be utilized for addressing concerns about data privacy, thus avoiding attacks from malicious users as well as CS. But these models resulted in higher computational costs, longer implementation times, and limited processing resources[7,8].

Emerging technologies like artificial intelligence (AI) and a cloud-based secure approach are wielding a secure HCS, while technology has enhanced significantly[9]. In analyzing the attacker's malicious behavior, a vital role is played by the AI, thus providing an endogenous security solution. By employing sophisticated prediction analytics on those participating in clinical trials, a broader range of data is evaluated by medical professionals, thereby reducing the expenses as well as the time required for medical tests[10]. However, there are a few limitations that are not sufficient in a secure healthcare model, like high response time and computation cost. Hence, the current work proposes PDHSMICK, a partial key-based secure distributed healthcare systems utilizing Mimetic Interpolation-based CRYSTALS-Kyber and I-PCC.

1.1. Problem definition

Prevailing research methodologies proposed for the security and preservation of electronic health records have some limitations. The limitations existing are:

- In the existing systems, the main vulnerability is the leakage of transaction privacy that occurs in the context of different data, tasks, and model architectures.
- Contemporary cryptographic algorithms like Rivest Shamir Adleman (RSA) and ECC can collapse owing to the development of quantum computing.
- The prevailing system has a high response time, and the whole data was able to be hacked by the attacker.

1.2. Objectives

Thus, to overcome these problems, the work has proposed PDHSMICK: a partial key-based secure EHR with distributed cloud for healthcare systems applying MI-CRYSTALS-Kyber. The main contributions made by the proposed research work are:

- To perform efficient user authorization by third-party auditors utilizing the UTF-32 technique.
- To enhance data security and perform efficient DCS selection, the work utilized the MI-CRYSTAL-Kyber algorithm and I-PCC techniques.
- To create complex hash codes and DSC for authorization, the work utilized the EX-EOF and DM-DSA techniques.

The paper's structure is systematized as follows: Section 2 elucidates the related works; Section 3 deals with the proposed technique; Section 4 describes the outcomes and evaluation; and lastly, Section 5 winds up the paper.

2. Literature survey

Zhang et al.[11] established homomorphic encryption-centric privacy-preserving federated learning in IoT-enabled HCSs. For replacing the conventional weight calculation technique grounded on the amount of data, a weighted average approach centred on data quality was wielded. The outcomes exhibited the model's superior performance. But, owing to the high-dimension parameter, the homomorphic encryption system did not encrypt every model.

Kumar et al.[12] recommended a secure together with effective cloud-centered Internet of-medical-things-enabled smart HCS with public verifiability. The system fetched the medical data as of multiple sensors implanted on the patient's body, signcrypted, and aggregated them under the Escrow-Free Identity-Based Aggregate Signcryption (EF-IDASC) system. The outcomes displayed the system's superior performance in the HCS. But, owing to the fluctuation in EF-IDASC, the model further needed enhancement in performance as well as efficacy.

Benil and Jasper[13] propounded cloud-centric security on outsourcing utilizing blockchain in e-health Systems. The digital signature to share and store data in the concerned cloud storage was generated by the ECC encrypted medical data and Certificateless Aggregate Signatures (CAS) scheme. As per the outcomes, the framework was more efficient in the health system. However, verification overhead ensured the model's security weakness.

Qiu et al.[14] introduced secure health data sharing for medical cyber-physical systems in healthcare. The model, which protected the data on a trusted device like the end-user smartphone, was grounded in a user-centric design; also, the end-user was allowed to control the access for data sharing. In the HCS, superior performance was shown by the model, as per the outcomes. For protecting the data with higher-level protection, data integrity, together with efficacy on smartphone platforms, the model was wielded. Nevertheless, the model was not suitable for protecting a series of numerous files.

Mubarakali et al.[15] explored a Protected and Powerful Healthcare based Blockchain (SRHB) technique with attribute-centric encryption for transmitting healthcare details. By utilizing wearable devices in a centralized HCS, the system gathered data from the patient. It observed the patient's health condition by measuring heartbeat while sleeping and walking over a distance. The data obtained from the patient was uploaded and saved on a cloud repository server. The outcomes showed that the model was more efficient in the HCS. But the model took more time for health record sharing.

Parah et al.[16] established a security framework utilizing Left Data Mapping (LDM) and Pixel Repetition Method (PRM). Here, the input was initially upscaled utilizing PRM; also, the binary secret data was encrypted utilizing the Rivest Cipher 4 (RC4) encryption approach. The obtained decimal equivalents were then encoded utilizing LDM. The outcomes proved the model's superiority by employing the Peak Signal to Noise Ratio (PSNR). However, authentication was not possible with RC4 techniques.

Mubarakali et al.[17] proffered Attribute-centric Health Records Protection (AHRP) approach for providing information access control, confidentiality, secrecy, plus the credibility. The presented approach involved the arrangement of access control for encrypting the information and also a privileged mode to authenticate a message devoid of uncovering the patient's personal information. The outcomes exhibited the model's superior performance with reduced encryption time. However, the model achieved a higher encryption time.

Guo et al.[18] employed Cipher text-Policy Attribute-Based Encryption (CP-ABE) system in cloud for healthcare data preservation. In the process of encrypting a Personal Health Record (PHR), the patient is able to divide it into various files analogus to importance. The hierarchical files were encrypted utilizing an

Integrated Access Tree Structure (IATS) related to the cipher text. Therefore, confidentiality was preserved with access control flexibility and fine gradient property. But the model led to a complex structure and was difficult to implement correctly.

Liang et al.[19] recommended In Distinguishability under Multi-Source Ordered Chosen Plaintext Attack (IND-MSOCPA)-based privacy-preserving technique for multiple users. For performing a range query over encrypted multi-source, the range query approach, which was a variant of the encryption approach, was wielded. The outcomes displayed that the model was efficient in privacy preservation. Nevertheless, the model required much time for privacy preservation.

Masud et al.[20] propounded a robust and lightweight secure access strategy for cloud-centric healthcare services. Here, a secure interface of access was utilized to permit only legitimate entities for storing and accessing the patient's information. As per the investigation, the developed model was lightweight; it also exhibited security properties, namely integrity, freshness, confidentiality, and authentication. Yet, the method was not cost-effective.

3. The proposed model

In this paper, a partial key-based secure EHR with distributed cloud for healthcare systems applying MI-CRYSTALS-Kyber (PDHSMICK) is proposed. The proposed system undergoes the following processes, namely registration, login, attribute extraction, selection of distributed cloud server (DCS), hash code generation, digital signature creation (DSC), and encryption. The proposed model's block diagram is displayed in **Figure 1**.

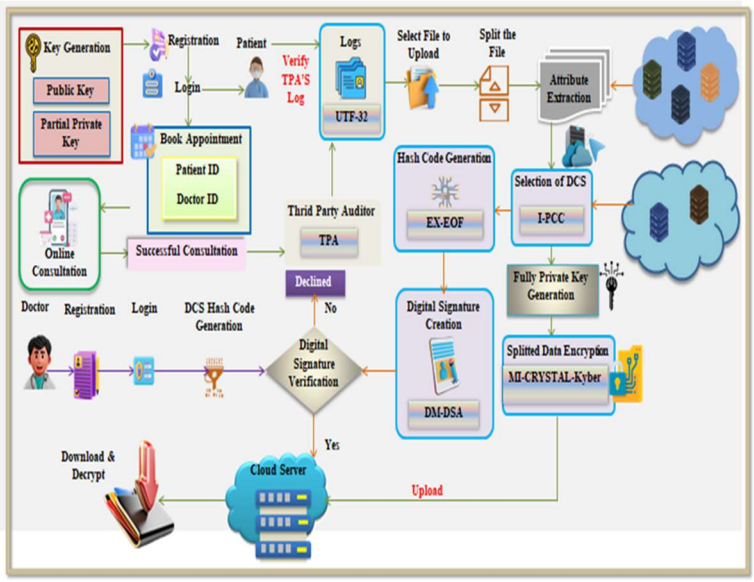

Figure 1. Block diagram of the proposed PDHSMICK.

3.1. Registration

Primarily, the registration phase was carried out in which the patient details (P_D) like name, address, mobile number, sex, age, username, and password, and the doctor details (Dtr_D), namely doctor name, age, sex, address, mobile number, medical department, branch, specialization, username, and password were entered and registered in the system, which is modeled as,

$$R_D = [P_D, Dtr_D] \tag{1}$$

where R_D signifies the registered details. At the same time, public and partial private keys are generated from the key generation center. During registration, the patient *ID* (P_{ID}) and doctor *ID* (d_{ID}) was generated, which are represented as:

$$ID = [P_{ID}, d_{ID}] \tag{2}$$

ID indicates the *ID* of the patient and doctor.

3.2. Login

After successful registration, the patients log in to the system with a username and password. If the password and username match with the registered patient details, then the system allows the patient to use the service.

$$L_{ogin} \xrightarrow{u_{pass}, pass(valid)} book_{app} \tag{3}$$

$$L_{ogin} \xrightarrow{u_{pass}, pass(invalid)} reject \tag{4}$$

where L_{ogin} symbolizes the login, $book_{app}$ exemplifies the book appointment, u_{pass} signifies the username, and $pass$ denotes the patient's password. After login, an appointment is booked by the patient with the doctor utilizing the doctor *ID* and patient *ID*. Next, consultation was done at the appointment schedule. After successful consultation, the consultation information was forwarded to the TPA, which verifies the integrity of information on demand of the patients. Then, TPA stores the consultation information in the log file in the form of an encoded format utilizing the UTF-32 algorithm. UTF-32 is a fixed length of encoding wielded to encode Unicode points in which each character is composed of 4 bytes. Thus, the encoded format of information $(E_{inf}())$ that is stored in log files is represented as,

$$E[E_1, E_2, E_3, .., E_a]_{inf} \tag{5}$$

where E_a denotes the a^{th} number of the encoded format of information. After that, to upload the EHRs, the patient logs in to the system. At that time, the consultation details are verified with the log file. On successful verification, only the patient can be able to upload the EHR. After successful verification, the patient selects the files (F) to upload into the CS. Next, the selected files are splitted into several parts grounded on file size, which is denoted as,

$$F = \{F_1, F_2, F_3, .., F_b\} \tag{6}$$

where F_b epitomizes the b^{th} number of splitted files.

3.3. Attributes extraction

After splitting the files (F), the attributes from the splitted files, such as file types, splitted file size, state, compressed, et cetera, and the attributes from the DCS, such as Random Access Memory (RAM) Size, Million Instructions Per Seconds (MIPS), bandwidth, OS, memory, speed, et cetera are extracted to improve the performance of the model, which is given by,

$$A_{ext} = [A_F, A_{DCS}] \tag{7}$$

where A_{ext} symbolizes the extracted attributes, A_F signifies the attributes from the splitted files, and A_{DCS} denotes the attributes from the *DCS*.

3.4. Selection of the distributed cloud server

After the extraction of attributes, the extracted attributes (A_{ext}) are fed into the selection phase for selecting the particular DCS. Here, the selection of DCS was done by utilizing the I-PCC technique, which has

the advantage of measuring the strength and direction of the relationship between two extracted attributes. However, the conventional PCC could not determine the non-linear relationship between extracted attributes, which results in improper selection of the distributed cloud server (DCS). Hence, to overcome this problem, the work proposed an interpolation-based PCC technique, which is wielded for determining the non-linear relationship between extracted attributes. Hence, the proposed framework is named as I-PCC algorithm.

Initially, (A_{ext}) is fed into the DCS selection phase. After that, the relationship between extracted attributes (α) determined by I-PCC is modelled as,

$$\alpha = \overline{g} + \left(h - \overline{h}\right) * \frac{\sum\left(h - \overline{h}\right)(g - \overline{g})}{\sqrt{\sum\left(h - \overline{h}\right)^2 (g - \overline{g})^2}} \tag{8}$$

where $\overline{g}, \overline{h}$ signifies the values of the function at one point (A_{ext}) and g, h elucidates the value of the function at another point (A_{ext}). Afterward, grounded on this relationship, the DCS is selected, which is represented as,

$$S_{DCS} = [S_1, S_2, S_3, \ldots S_c] \tag{9}$$

where S_{DCS} denotes the selected *DCS*, S_c symbolizes the c^{th} number of the selected *DCS*. After that, hash code generation and DSC (digital signature creation) were carried out for authorization purposes.

3.5. Hash code generation

After the selection of DCS, the hash code generation was done by utilizing split files name $F(N)$ selected DCS name S_{DCS} (N), patient ID (P_{ID}), and doctor ID (d_{ID}). For hash code generation, the patient ID and doctor ID generated during the registration phase were utilized. In this work, the hash code was generated by utilizing the EX-EOF algorithm. The standardization of SHA-3 included the specification of two functions, namely SHAKE 128 and SHAKE 256, where both shakes are referred to as EOF. Here, the work concentrated on SHAKE 256, which is more resistant to attack and provides better security. However, in conventional EOF, the maximum distance between two occurrences of different data has the same element in an array, which results in poor hashing. Therefore, exponential-based EOF was proposed to overcome these issues. Hence, the proposed method is named EX-EOF.

Initially, the input data (x) is given by,

$$x = [F(N), S_{DCS}(N), P_{ID}, d_{ID}] \tag{10}$$

Then, the input data is given to the exponential function for enhancing the model's performance, which is given by,

$$y = exp(x) \tag{11}$$

where y indicates the exponential of input data. After that, y is fed into the SHAKE-256, which is for protecting data and a unique 576-bit signature ($2 \times 256 = 512$) for specific text. It utilizes sponge construction, where the information is consumed by the sponge and the result is squeezed out. Primarily, the exponential data was encoded, which is represented as,

$$\Re = \nabla(y) \tag{12}$$

where $\Re$ signifies the encoded data and $\nabla(y)$exemplifies the encoding of y. After encoding, the hash code was generated. Lastly, the generated hash is given by,

$$H_{code} = \lambda\left(S_{hke}(\Re)\right) \tag{13}$$

where H_{code} signifies the generated hash code, $\lambda(\bullet)$ epitomizes the hash of encoded data, and $S_{hke}(\Re)$ exemplifies the SHAKE of encoded data.

3.6. Digital signature creation

After hash code creation (H_{code}), the signature was created by utilizing the DM-DSA technique for validating the authenticity and data integrity. DSA works grounded on algebraic properties of discrete logarithmic functions and modular exponentiations for creating a digital signature that comprises two 160-bit numbers derived as of the message digests as well as private keys. However, in the prevailing DSA, the algorithm is grounded on the Diffie-Hellman Key Exchange technique. So, the attacker can verify the signature utilizing a public parameter. Therefore, the work proposed the Double Mod technique, which overcomes these problems. Hence, the proposed system is named DM-DSA. **Figure 2** depicts the architecture of the DM-DSA.

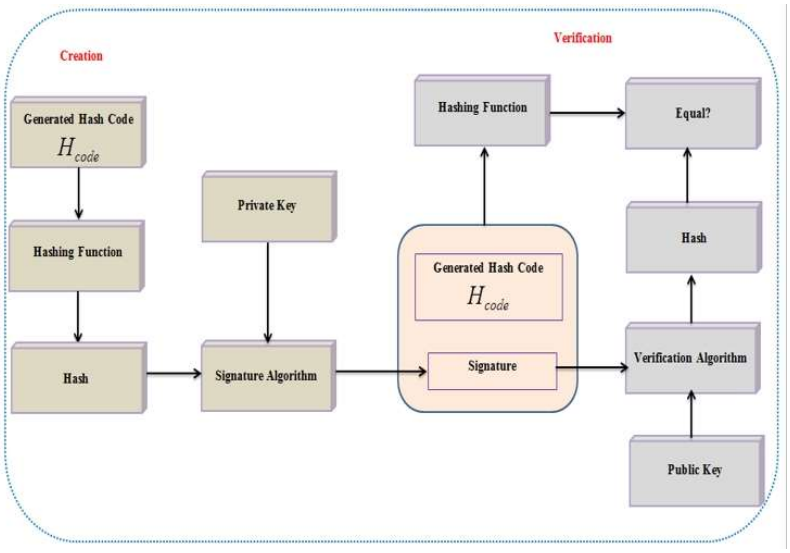

Figure 2. Architecture of DM-DSA.

The DM-DSA algorithm involves two phases, namely signature creation and signature verification, which are explained further:

3.6.1. Signature creation

For creating the message signature, initially, compute two integer values $rands$. The value of (r) is computed utilizing the random number and the base point, which is given by,

$$r = (\beta * bp \, mod \, \chi) \, mod \, \gamma \tag{14}$$

where β exemplifies the random number, bp denotes the base point, χ,γ indicates the prime number. If the value of (r) is zero, a new random number is chosen and the value (r) is computed again. Next, the value of (s) is computed by taking H_{code}, β, and the secret key (φ). Thus, the value of (s) is represented as,

$$s = (\beta^{-1} H_{code} + \varphi + \|r\|) \, mod \, \gamma \tag{15}$$

where $\|r\|$ specifies the double mod of (r). If the value of (s) is zero, a new random number is chosen and the value (rs) is computed again. After signature creation, it started to send the request as well as the signature to the receiver side.

3.6.2. Signature verification

After receiving the request, verification was done to validate the signature. To verify the signature, at first, compute (m) the modular multiplicative inverse of $s \, mod \, \gamma$, which is denoted as

$$m = s^{-1} \, mod \, \gamma \tag{16}$$

After that, the mathematical formula for digital signature verification (D_{verf}) is represented as,

$$D_{verf} = \left((k_1 * bp + k_2 - \varphi)\right) mod \, \gamma \tag{17}$$

where k_1 and k_2 are the variables that are specified as,

$$k_1 = H_{code} * m \, mod \, \gamma \tag{18}$$

$$k_2 = \|r\| * m \, mod \, \gamma \tag{19}$$

The validity of the digital signature was checked by utilizing the condition,

$$\begin{cases} D_{verf} = \|r\|, matched \\ D_{verf} \neq \|r\|, notmatched \end{cases} \tag{20}$$

The pseudocode of the proposed **DM-DSA**

Input: hash code H_{code}
Output: Verification of digital signature
Begin
Initialize hash code H_{code} and secret key (φ)
For all training steps **do**
\\ Signature creation
Compute $r$and $s$using (11) and (12)
Obtain digital signature $\{r, s\}$
\\ Signature verification
Compute m
$m = s^{-1} \, mod \, \gamma$
Verify signature
$D\_verf = ((k\_1 * bp + k\_2 - \varphi)) \; mod \; \gamma$
If $\left(D_{verf} = r\right)$
Signature matched
Else
Signature not matched
End If
End For
End

After the digital signature was created, concurrently, the split files are encrypted for secure uploading of the files.

3.7. Encryption

Here, a fully private key was generated, which is wielded for encrypting the split files. The encryption was done by utilizing the MI-CRYSTAL-Kyber technique, which efficiently encrypts the files for uploading and downloading the files from the CS. CRYSTAL-Kyber is a Key Encapsulation Method (KEM) designed to be resistant to malicious attacks. The conventional CRYSTAL-Kyber generates unstable parameters, which degrade the data security. Hence, the work proposed Mimetic Interpolation (MI) technique, which efficiently secures the data from the attack. Hence, the proposed approach is named MI-CRYSTAL-Kyber.

Initially, the created digital signature ($\|r\|$) was fed into the MI-CRYSTAL-Kyber algorithm, which included 3 steps, namely key generation, encryption, and decryption. First, the parameters are set to 256 to encapsulate keys with 256 bits of entropy. Let W indicates the ring, which is computed by utilizing the MI technique. MI is a technique for interpolating different forms, which is wielded to efficiently secure the data from attack. Thus, the ring is modelled as,

$$W = \frac{\aleph[B]}{(1 - B^i) + B^i} \tag{21}$$

where $\aleph[B]$ denotes the integer polynomial and $\left(1 - B^i\right)$ symbolizes the $i^{th} i^{th}$ cyclotomic polynomial.

3.7.1. Key generation

To generate the key, at first, compute τ and υ, which represents the seed by utilizing EOF. EOF of τ is represented as,

$$E_{of}(\tau) = M \sim W^{l \times l} \tag{22}$$

where $E_{of}(\tau)$ specifies the EOF of τ, M indicates the matrix, and l denotes the positive integer. Then, EOF of υ is denoted as,

$$E_{of}(\upsilon) = (o, p) \sim bi^l \times bi^l \tag{23}$$

where $E_{of}(\upsilon)$ signifies the EOF of υ, (o, p) exemplifies the noise term, and bi symbolizes the binomial distribution. Next, the compressed value is given by,

$$\phi = compress(Mo + p, \vartheta) \tag{24}$$

where ϑ epitomizes the positive integer. Lastly, the fully private key was generated as p_{key}.

3.7.2. Encryption

In encryption, at first, decompression of ϕ is done, which is modelled as,

$$\phi = decompress(\phi, \vartheta) \tag{25}$$

After that, the cipher text is created, which is represented as,

$$c_1 = compress(M^T \varpi + n_1, \zeta) \tag{26}$$

$$c_2 = compress\left(\phi^T \varpi + n_2 + \left[\frac{f}{2}\right] \bullet r, q\right) \tag{27}$$

where c_1 and c_2 are the cipher text, M^T signifies the transpose of a matrix, ϖepitomizes the random number, n_1 and n_2 symbolizes the noise in encryption function, ζ,q exemplifies the positive integer values, ϕ^T denotes the transpose of ϕ, and f represents the polynomial modulo. Finally, the data are encrypted and uploaded to the CS.

3.7.3. Decryption

Here, the decompression of cipher text is done to decrypt the data. The mathematical expression for the decryption phase is represented as,

$$c_1 = decompress(c_1, \zeta) \tag{28}$$

$$c_2 = decompress(c_2, q) \tag{29}$$

After successful uploading, the registered doctor login to the system, and further verification processes, namely hash code verification and digital signature verification were carried out, which are explicated in sections 3.5 and 3.6, correspondingly.

$$\begin{cases} H_{code}, D_{verf}(matched), Downloaddata \\ H_{code}, D_{verf}(unmatched), Declined \end{cases} \quad (30)$$

If the verification is matched, the system allows the doctor to download and decrypt the data; else, the request is declined.

Pseudocode of the proposed **MI-CRYSTALS-Kyber**

Input: Patient and Doctor
Output: Data download and Decrypt
Begin
Initialize patient, doctor
Register details of the patient and the doctor
For all training **do**
Generate public and partial private key
\\ Patient Side
Login patients
If (valid)
{
Allow patient to use services
Else
Not allowed
}
End IF
Book appointment
Compute consultation
Store in log files by TPA
Verify log files
Select files to upload
Split files
$F = \{F_1, F_2, F_3, .., F_b\}$
Extract attributes
$A_{ext} = [A_F, A_{DCS}]$
Select DCS
$S_{DCS} = [S_1, S_2, S_3, \ldots S_c]$
Generate hash code using EX-EOF
Create digital signature using DM-DSA
Encrypt splitted files
\\ Doctor Side
Login system
Verify hash code and digital signature
If $(D_{verf} = r)$
{
Download and decrypt data
Else
Declined
}
End If
End For
End

4. Evaluation

In this section, the performance of PDHSMICK, the proposed system, is analyzed by analogizing its outcomes with those of other conventional methods.

In the working platform of PYTHON, the proposed PDHSMICK is implemented with a system configuration of Windows 10 with a 1.75 GHz processor speed, random access memory of 4 GB, and SQLite for data storage and retrieval. The proposed work's performance evaluation is detailed further.

4.1. Specification of the database

The proposed system uses 5 types of disease-related datasets from the website https://www.kaggle.com/datasets, namely Breast Cancer Wisconsin (Diagnostic) Data Set (DST-I), Diabetes Dataset (DST-II), Heart Disease Dataset (DST-III), Obesity Dataset (DST-IV), and Polycystic Ovary Syndrome (PCOS) (DST-V).

4.2. Performance analysis

The proposed MI-CRYSTAL-Kyber's performance analysis is validated regarding (i) attack levels, (ii) key generation time, (iii) encryption time, (iv) decryption time, (v) encryption overhead, (vi) memory usage on encryption, (vii) Memory usage on decryption, and (viii) security levels. After that, the obtained results are compared with other existing models like CRYSTAL-Kyber, ECC, RSA, and ELGamal.

The attack levels of the proposed and prevailing techniques are elucidated in **Figure 3**. The attack levels are the number of bits prevented from attack that the model achieves. The lower attack levels lead to better performance of the system. The attack level achieved by the proposed one is 1.229%, which is low when contrasted with the prevailing approaches, whereas the attack levels obtained by the existing models are 4.615% for CRYSTAL-Kyber, 6.368% for ECC, 10.679% for RSA, and 14.599% for ELGamal. Hence, it is proved that the proposed mechanism performed better than the existing techniques.

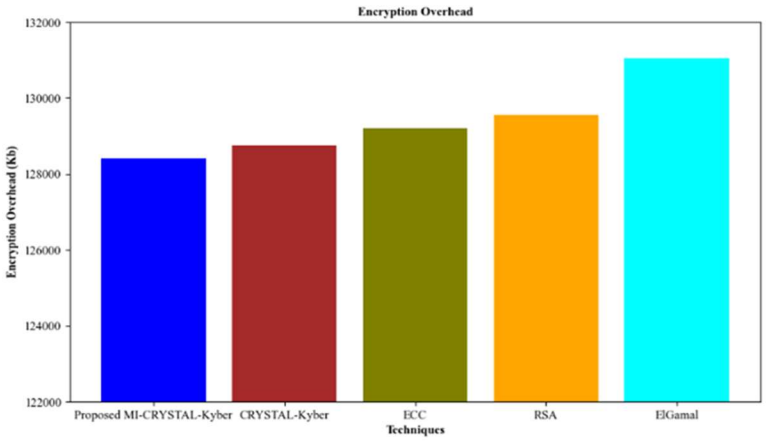

Figure 3. Attack levels of the proposed and existing models.

Regarding key generation time, decryption time, and encryption time, the comparative analysis of the proposed and prevailing techniques is demonstrated in **Table 1**. The proposed approach attains an encryption time of 20,254 ms, which is low when contrasted with the prevailing approaches, whereas the encryption times achieved by the existing models like CRYSTAL-Kyber, ECC, RSA, and ELGamal are 2215 ms, 3714 ms, 4412 ms, and 4828 ms, respectively. Likewise, the proposed framework's decryption time and key generation time are 2627 ms and 4361 ms, respectively. Hence, it is clear that the proposed technique outperforms the prevailing systems.

Table 1. Comparative analysis of the proposed and existing models.

Methods	Encryption time (ms)	Decryption time (ms)	Key generation time (ms)
Proposed MI-CRYSTAL-Kyber	20,254	26,277	4361
CRYSTAL-Kyber	22,151	25,135	4981
ECC	37,146	38,113	5576
RSA	44,125	45,131	6142
ElGamal	48,289	49,113	6894

The encryption overhead value of the proposed and prevailing models is elucidated in **Figure 4**. The lower encryption overhead value results in the model's superior performance. The proposed technique attains an encryption overhead of 1284 Kb, whereas the existing models like CRYSTAL-Kyber, ECC, RSA, and ELGamal attain an encryption overhead of 287 Kb, 1292 Kb, 1295 Kb, and 1310 Kb, respectively. Hence, it is observed that the technique shows superior performance to conventional approaches.

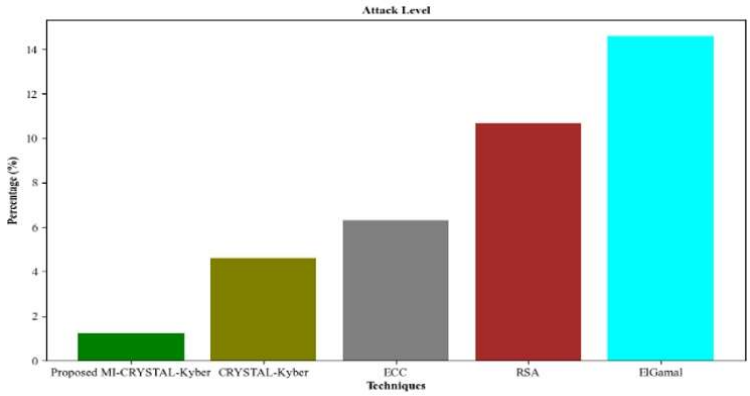

Figure 4. Encryption overhead of the proposed and existing models.

Table 2 displays the comparative assessment of the proposed and prevailing mechanisms regarding encryption on memory usage and decryption on memory usage. During encryption, the proposed approach attains a memory usage of 13,358 Kb, which is low when analogized with the prevailing approaches like CRYSTAL-Kyber, ECC, RSA, and ELGamal. Likewise, during decryption, the proposed mechanism attains a memory usage of 1157 Kb, which is also low when analogized with the prevailing approaches. Therefore, it is proved that the proposed system achieved better performance in memory usage during encryption and decryption.

Table 2. Memory usage on encryption and decryption.

Method	Memory usage on encryption (Kb)	Memory usage on decryption (Kb)	Method	Memory usage on encryption (Kb)	Memory usage on decryption (Kb)
Proposed MI-CRYSTAL-Kyber	133,581	115,761	Proposed MI-CRYSTAL-Kyber	133,581	115,761
CRYSTAL-Kyber	136,599	118,356	CRYSTAL-Kyber	136,599	118,356
ECC	139,365	120,317	ECC	139,365	120,317
RSA	141,967	122,232	RSA	141,967	122,232
ElGamal	1,005,043	833,941	ElGamal	1,005,043	833,941

Figure 5 depicts the security levels of the proposed and conventional mechanisms. The security levels are the measure of strength that the model achieves. It is computed based on its key strength, i.e., for the number of bits in its key, the attacker has to perform 2n attempts to crack it. Here, the proposed one attains a

security level of 98.77% by using SHAKE 256, which contains 576 bits, which is due to the fact that the MI techniques used in the proposed technique efficiently encrypt the data when compared with other existing models. Hence, it is shown that the proposed technique is more efficient in encryption.

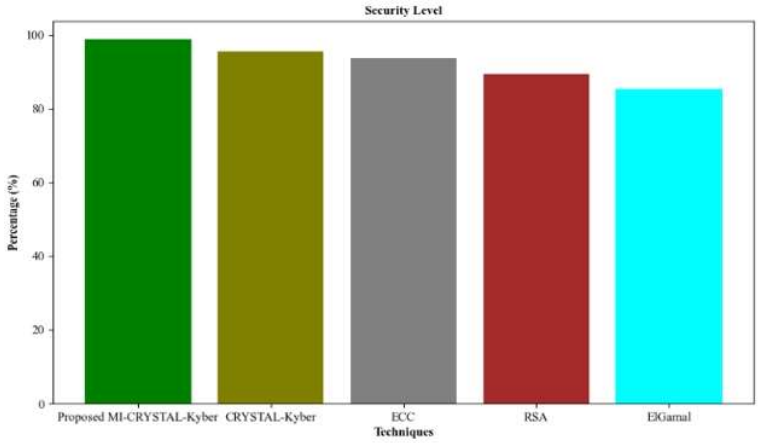

Figure 5. Security levels of the proposed and existing models.

4.3. Performance analysis based on hash code generation

The proposed EX-EOF's performance evaluation is carried out regarding hash code generation time; then, the obtained result is compared with the existing models like EOF, SHA-512, Message Digest 5 (MD5), and BLAKE.

The hash code generation time of the proposed and prevailing approaches is displayed in **Figure 6**. The time taken by the proposed mechanism for generating the hash code is named hash code generation time. The hash code generation time taken by the proposed framework is 813 ms, which is low when analogized with the prevailing approaches, whereas the hash code generation time obtained by the prevailing models is 950 ms for EOF, 1008 ms for SHA-512, 1206 ms for MD5, and 1404 ms for BLAKE. Therefore, it is observed that the proposed technique is more efficient in the generation of hash codes.

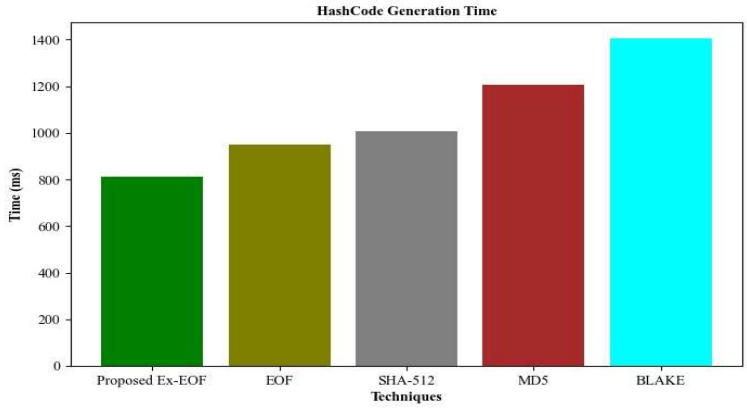

Figure 6. Hash code generation time of proposed and existing models.

4.4. Performance analysis of selection of the distributed cloud server

The proposed I-PCC's performance is validated in terms of response time. Afterward, the obtained results are analogized with the existing models, namely PCC, Chi-Square, *T*-test, and IG.

Figure 7 depicts the response time of the proposed and conventional frameworks. The time taken by the proposed system for selecting the DCS is named response time. The approach attains a lower response time of 820 ms, whereas the response time taken by the existing models like PCC, Chi-Square, *T*-test, and IG are 1210

ms, 1503 ms, 1802 ms, and 1908 ms, respectively. Therefore, the proposed technique is more efficient in the selection of the DCS.

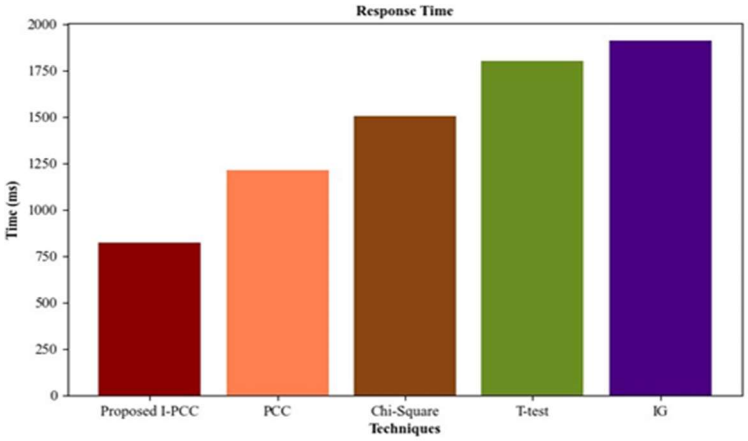

Figure 7. Response time of the proposed and existing models.

4.5. Performance analysis w.r.to file size, upload time

The proposed mechanism's performance assessment is validated concerning upload time, memory usage, verification delay, computation delay, and communication overhead.

Grounded on upload time and memory usage concerning file size, the proposed technique's performance analysis is exhibited in **Table 3**. For 20 Mb of file size, the time taken by the proposed one to upload the file is 12,973 ms. Likewise, the time taken by the proposed technique increases for a varying number of file sizes, which displays the model's better performance. Similarly, for 25 Mb of file size, the memory utilized by the proposed system is 1538 Kb. Likewise, for a varying number of file sizes, the memory used by the proposed technique should be increased. Therefore, it is concluded that the proposed mechanism shows better performance in a secure HCS.

Table 3. Performance analysis of the proposed model.

File size (Mb)	Upload time (ms)	Memory usage (Kb)
20	12,973	153,831
25	17,682	196,593
30	21,983	259,653
35	23,586	299,872
40	24,729	348,712
45	26,215	380,377

The proposed approach's performance evaluation concerning (a) verification delay, (b) computation delay, and (c) communication overhead is depicted in **Figure 8**. From **Figure 8(a)**, it is observed that for 10 of signatures, the verification delay obtained by the proposed framework is 102 ms. Likewise, for a varying number of signatures, the proposed algorithm attains a lower verification delay. Similarly, the computation delay of the doctor, DCS, and hospital is shown in **Figure 8(b)**. It is observed that with the increase in the number of patients, the computation delay of the doctor, DCS, and hospital also increased, which displays the proposed one's superior performance.

Likewise, from **Figure 8(c)**, the communication overhead between the hospital and patient has been analyzed. Here, the proposed mechanism is built on DCS, which maintains secure transactions. Therefore, the

proposed mechanism is efficient in communication overhead. Thus, it is clear that for secure healthcare, the proposed system is more efficient.

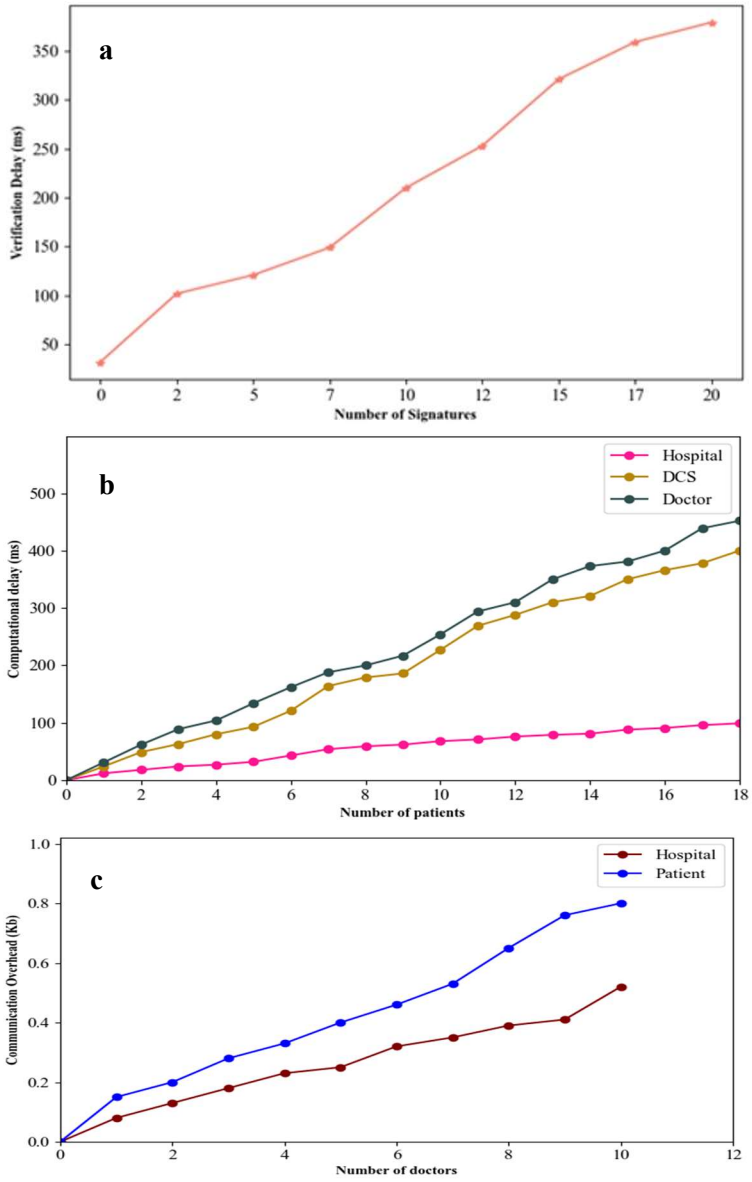

Figure 8. Performance analysis of the proposed model.

Concerning encryption time, the comparison of the proposed and the literature survey models is demonstrated in **Table 4**. The literature survey model attains a higher encryption time, such as 23,456 ms for Parah et al.[16], 264,35 ms for Mubarakali et al.[15], and 30,123 ms for Guo et al.[18], which is owing to the leakage of information. Hence, the MI technique wielded in the proposed technique is more efficient in securing the information in the DCS, which leads to a lower encryption time of 20,254 ms. Hence, it is proved that the proposed technique is more efficient in encryption.

Table 4. Comparison of proposed and literature survey models.

Techniques	Encryption time (ms)
Proposed model	20,254
Parah et al., 2021	23,456
Mubarakali et al., 2020	26,435
Guo et al., 2019	30,123

5. Conclusion

This work proposes a partial key-based secure EHR with distributed cloud for healthcare systems applying MI-CRYSTAL-Kyber (PDHSMICK). Here, the encryption was done by utilizing MI-CRYSTAL-Kyber techniques; also, the DSC was created by utilizing DM-DSA. Further, by utilizing the EX-EOF technique, the hash code was generated. Afterward, the experimental assessment was performed, where the proposed technique's performance was assessed concerning various metrics. The outcomes revealed that a security level of 98.77% was achieved by the proposed one. The encryption time, decryption time, hash code generation time, and attack levels achieved by the proposed framework are 20,254 ms, 26,277 ms, 813 ms, and 1.229%, respectively. Thus, the system had high performance and achieved a low response time. Therefore, the proposed technique is more efficient in securing HCS (healthcare systems).

Here, the work does not concentrate on the scalability and interoperability of distributed systems. To solve this issue, the work may use advanced techniques to enhance the significance of DCSs in the near future.

Author contributions

Conceptualization, PS and RD; methodology, PS; software, PS; validation, RD, PS; formal analysis, PS; investigation, PS; resources, PS; data curation, PS; writing—original draft preparation, PS; writing—review and editing, PS; visualization, PS; supervision, RD; project administration, RD. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Rehman A, Abbas S, Khan MA, et al. A secure healthcare 5.0 system based on blockchain technology entangled with federated learning technique. *Computers in Biology and Medicine* 2022; 150: 106019. doi: 10.1016/j.compbiomed.2022.106019
2. Sujith A, Sajja GS, Mahalakshmi V, et al. Systematic review of smart health monitoring using deep learning and Artificial intelligence. *Neuroscience Informatics* 2022; 2(3): 100028. doi: 10.1016/j.neuri.2021.100028
3. Butpheng C, Yeh KH, Xiong H. Security and privacy in IoT-cloud-based e-health systems—A comprehensive review. *Symmetry* 2020; 12(7): 1191. doi: 10.3390/sym12071191
4. Haleem A, Javaid M, Singh RP, et al. Blockchain technology applications in healthcare: An overview. *International Journal of Intelligent Networks* 2021; 2: 130–139. doi: 10.1016/j.ijin.2021.09.005
5. Thilagam K, Beno A, Lakshmi MV, et al. Secure IoT healthcare architecture with deep learning-based access control system. *Journal of Nanomaterials* 2022; 2022: 2638613. doi: 10.1155/2022/2638613
6. Pelekoudas-Oikonomou F, Zachos G, Papaioannou M, et al. Blockchain-based security mechanisms for IoMT edge networks in IoMT-based healthcare monitoring systems. *Sensors* 2022; 22(7): 2449. doi: 10.3390/s22072449
7. Chaudhary RRK, Chatterjee K. A lightweight security framework for electronic healthcare system. *International Journal of Information Technology* 2022; 14: 3109–3121. doi: 10.1007/s41870-022-01034-4
8. Xue L. DSAS: A secure data sharing and authorized searchable framework for e-healthcare system. *IEEE Access* 2022; 10: 30779–30791. doi: 10.1109/ACCESS.2022.3153120
9. Alabdulatif A, Khalil I, Saidur Rahman M. Security of blockchain and AI-empowered smart healthcare: Application-based analysis. *Applied Sciences* 2022; 12(21): 11039. doi: 10.3390/app122111039
10. Javaid M, Haleem A, Singh RP, et al. Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks* 2022; 3: 58–73. doi: 10.1016/j.ijin.2022.05.002
11. Zhang L, Xu J, Vijayakumar P, et al. Homomorphic encryption-based privacy-preserving federated learning in IoT-enabled healthcare system. *IEEE Transactions on Network Science and Engineering* 2022; 10(5): 2864–2880. doi: 10.1109/TNSE.2022.3185327
12. Kumar M, Chand S. A secure and efficient cloud-centric internet-of-medical-things-enabled smart healthcare system with public verifiability. *IEEE Internet of Things Journal* 2020; 7(10): 10650–10659. doi: 10.1109/JIOT.2020.3006523
13. Benil T, Jasper J. Cloud based security on outsourcing using blockchain in e-health systems. *Computer Networks* 2020; 178: 107344. doi: 10.1016/j.comnet.2020.107344

14. Qiu H, Qiu M, Liu M, Memmi G. Secure health data sharing for medical cyber-physical systems for the healthcare 4.0. *IEEE Journal of Biomedical and Health Informatics* 2020; 24(9): 2499–2505. doi: 10.1109/JBHI.2020.2973467
15. Mubarakali A. Healthcare services monitoring in cloud using secure and robust healthcare-based BLOCKCHAIN (SRHB) approach. *Mobile Networks and Applications* 2020; 25: 1330–1337. doi: 10.1007/s11036-020-01551-1
16. Parah SA, Kaw JA, Bellavista P, et al. Efficient security and authentication for edge-based internet of medical things. *IEEE Internet of Things Journal* 2021; 8(21): 15652–15662. doi: 10.1109/JIOT.2020.3038009
17. Mubarakali A, Ashwin M, Mavaluru D, et al. Design an attribute based health record protection algorithm for healthcare services in cloud environment. *Multimedia Tools and Applications* 2020 79: 3943–3956. doi: 10.1007/s11042-019-7494-7
18. Guo R, Li X, Zheng D, Zhang Y. An attribute-based encryption scheme with multiple authorities on hierarchical personal health record in cloud. *The Journal of Supercomputing* 2018; 76: 4884–4903. doi: 10.1007/s11227-018-2644-7
19. Liang J, Qin Z, Xiao S, et al. Privacy-preserving range query over multi-source electronic health records in public clouds. *Journal of Parallel and Distributed Computing* 2019; 135: 127–139. doi: 10.1016/j.jpdc.2019.08.011
20. Masud M, Gaba GS, Choudhary K, et al. A robust and lightweight secure access scheme for cloud based e-healthcare services. *Peer-to-peer Networking and Applications* 2021; 14: 3043–3057. doi: 10.1007/s12083-021-01162-x