International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

A CRYPTOGRAPHIC AUTHORIZATION MODEL FOR SECURE EHRS IN HEALTHCARE CLOUD INTEGRATING MAC VERIFICATION WITH HASHING METHODS

S. Prathima[1] Dr. R. Durga[2]

[1]Research Scholar, Department of Computer Science, School of Computing Sciences

[2]Associate Professor, Department of Computer Science, School of Computing Sciences

Vels Institute of Technology and Advanced Studies (VISTAS), Chennai, India

Email: prathismanian@gmail.com[1], drrdurgaresearch@gmail.com[2]

ABSTRACT:

In the current era of rapid technological advancement, electronic/digital health records are increasingly used by medical institutions. These electronic health records require a safer cloud to store and process data. Nonetheless, establishing a cloud-based medical data centre comes with hefty construction expenses and specialized technological assistance. Hence, we suggest a cryptographic authorization model using MAC verification and hashing (CAEHRMH). At first, patient registration is done and then the nodes are initialized. After that, the test packet is transmitted to create the authenticated path by calculating the cipher text using CRROT13. Then, the doctor's appointment is booked and the consultation is done. After the successful consultation, parameters are extracted and converted into hash code using DF-ASH512 while the secret key is generated using LDH. With this key, the MAC address is generated using the LDH-TDES-MAC algorithm. Next, the data is converted into encrypted formats at a double time using CREDCC. Then, the doctor will log in with the system and the parameters are extracted and converted into the hash code for generating the MAC address. The MAC address is matched with the patient's MAC address. Upon matching, the doctor downloads the encrypted data and decrypts it. The experimental findings demonstrated that the suggested methodology is significantly more secure than the current methods.

KEYWORDS: *Electronic Health Records (EHRs), Digit Folding-based Algorithm for Secure Hashing (DFASH512), Log Diffie Hellman (LDH), Log Diffie Hellman (LDH) based Triple Data Encryption Standard MAC (LDH-TDES-MAC), Cube Root Edwards Curve Cryptographic (CREDCC).*

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

I. INTRODUCTION

In the last ten years, Internet of Things (IoT) has become well known and accepted because of it's low power and lossy networks (Saba et al., 2020). Since information technology advances more quickly, increasing number of medical facilities use digital information systems, that generate enormous amounts of health-care data daily. The Internet of Medical Things (IoMT) emerged as a result of integration of medical equipment into the IoT (Yaacoub et al., 2020). Improved patient care, faster emergency response times and more precise disease control and management can all result from the sharing of health-care information among various providers (Abdellatif et al., 2020). Nevertheless, providing services of high quality are challenged by several additional challenges. The rise in e-fraud cases globally during the recent years has been mostly attributed to medical identity theft. Additionally, patient names, diagnostic data and provider-ids are broadcast as plain text transmission on a limited basis, resulting in the exposure of diagnostic as well as patient's personal data (Wang & Cai et al., 2020). Luckily, a favorable solution is accomplished by cloud storage technologies. At present, the healthcare systems greatly benefit from cloud storage technologies. Some of its advantages are fast delivery, good sharing, more storage space, lower cost, easy-access, along with dynamic association (Misra et al.., 2020). Healthcare's rapid digitalization has led to the creation of enormous amounts of patient's electronic health records. This progress in healthcare opens the door to demands for data security that have never existed before (Chehri et al., 2021). A new generation of technology called Electronic Health Records (EHRs) allows for the upkeep of medical records in the cloud medium at reasonable cost and with high levels of interoperability and excellent quality of these EHRs (Uddin et al., 2021). Regardless of the geographical location, EHRs play an important role in enabling physicians and healthcare workers to access all the patient records at any time and from anywhere. For such real-time access to patient info cloud-based EHR implementation across all levels of the healthcare system is always recommended (Yang et al., 2021). However, when users store EHR data on the cloud server, the data will suffer a variety of security threats, involving privacy of the data and authentication of the data. Various security and privacy concerns include data confidentiality, access control, integrity, cyber security issues, interoperability, and accountability (Ambarkar & Shekokar et al., 2020). A growing number of medical-service centres are now using blockchain technology to share EHRs across various platforms. In spite of this, because of the size of the blockchains and their price it's difficult to store all EHR data (Kim et al., 2020). In order to mitigate such risks, this paper has proposed a cryptographic authorization model for secure EHRs in healthcare cloud integrating MAC verification with hashing (CAEHRMH).

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

1.1. PROBLEM DEFINITION

Many approaches have been developed for securing and preserving EHR data. But the Existing research methodologies have some drawbacks, which are described as follows

- In the existing systems, the authorised user could not perform the authorization process as the access policy gets lost.
- The main vulnerability is that the transparency of transactions is compromised because every user with access to the network will see the balances and details of the public keys.
- In the existing Hash function-based Authentication system, the shorter length of input to hashing algorithm led to an attack, which reduces the authentication of the entire system.

1.2. OBJECTIVES

Hence, the primary objective of this work is intended to design a cryptographic authorization model for secure EHRs in healthcare cloud integrating MAC verification with hashing (CAEHRMH). The research work's aims to accomplish the following contributions:

- To introduce a LDH technique that creates an efficient secret key.
- To perform the node authorization process, the CRROT-13 ciphering technique is used.
- To improve data security, the CREDCC algorithm is introduced.
- To perform efficient user authorization, LDH-TDES-MAC Algorithm is proposed.
- To create complex hash code for authorization, DF-ASH512 hashing technique is proposed.

The remaining sections of this paper is divided into Sections 2 and 3 that survey the existing literature pertaining to the proposed work. Section 4 that describes the results and discussions resulting from the proposal according to performance metrics, and section 5 that discusses the implications of the proposal.

II. LITERATURE SURVEY

Sharmila *et al.* [15] identified the E-MHMS, a healthcare monitoring system based on MAC technology for Delay-Aware data transmissions. For Encryption and key distribution, E-MHMS employed the time-based elliptic curve method. A better performance in terms of crucial network parameters was shown by the E-MHMS methodized in the simulation environment. But the scheme took more time to complete the entire process.

Sowjanya *et al.* [16] Recommended a key management technique for Internet of Things-based healthcare systems. Complex decryption processes and the key-escrow issues were the two main problems with the usability of CP- ABE for a constrained context like Internet of Things. For that, a lightweight key was developed by the Ciphertext Policy-Attribute Based Encryption (CP-ABE) approach inferred from Elliptic Curve Cryptography (ECC). The proposed CP-ABE approach was

found to be key-escrow free as well as it minimized the data receiver's decryption overhead. Unfortunately, this approach was vulnerable to collusion attacks.

Preveze *et al.* [13] devised a Hospital Healthcare Monitoring System with Adaptive Switching (AS) technique. As Internet of Health Things (IoHT) paradigm was proposed and combined with a software-defined network (SDN). The AS technique was proposed as an adaptive access strategy built on attentively hoping between Go-Back-N and Selective Repeat approaches that are already in use. The simulation results proved the effectiveness of the developed model. However, secure data transmission has not been dealt with.

Limbasiya *et al.* [9] proposed a Privacy-Preserved Mutual Authentication and Key Agreement Scheme for multi- server Healthcare Systems. For preserving privacy, this method made use of: concatenation operations, bit-wise XOR, as well as SHA-256. Furthermore, this scheme prevented the use of a stolen smart card and user impersonation while also achieving several privacy and security features for the medical users. Although, it was unable to maintain forward secrecy and it remained vulnerable to insider attacks.

Ahmad *et al.* [2] presented a local coordination X- MAC (LCX-MAC) for an improved energy efficient and

end-to-end secure protocol for the IoT Healthcare applications. To determine the impact of the local coordination on X-MAC, a mathematical model of LCX-MAC was developed that included the local coordination. Analysis revealed that in terms of latency, energy and throughput LCX-MAC performed well than X-MAC. But, while the network was over overcrowded, it lacked a method to prevent collisions.

Denis *et al.* [7] created a new structure for cloud-basedhealthcare systems to communicate securely with medical data. The uniqueness was the application of the Adaptive Genetic Algorithm for Optimal Pixel Adjustment Process (AGA- OPAP), that improved its ability to conceal data and enhance its imperceptibility qualities. Attacks like steganalysis attacks were thwarted by the created model. But the scheme did not enable faster access to health records.

Monowar *et al.* [11] developed the 'ThMAC' Thermal aware duty cycle MAC protocol for the Internet of Things healthcare. ThMAC also introduced a super frame structure. Additionally, ThMAC used an emergency data management system to guarantee the timely and accurate distribution of intermittent generated data. The results of the evaluation of ThMAC's performance revealed exceptional performance. On the other hand, the technique caused hiccups, which resulted in brief productivity losses.

Chinnasamy et al. [6] suggested a framework for secured Electronic Healthcare Data (EHD) retrieval in the health cloud environment. A hybrid cryptography algorithm was implemented that used Improved Key Generation Schemeof the RSA (IKGSR) algorithm for healthcare data encryptions. This

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

framework was compared with existing methods and found that the suggested work was able to provide enhanced security as well as retrieved the data effectively. Text and multimedia files cannot be used with it, though.

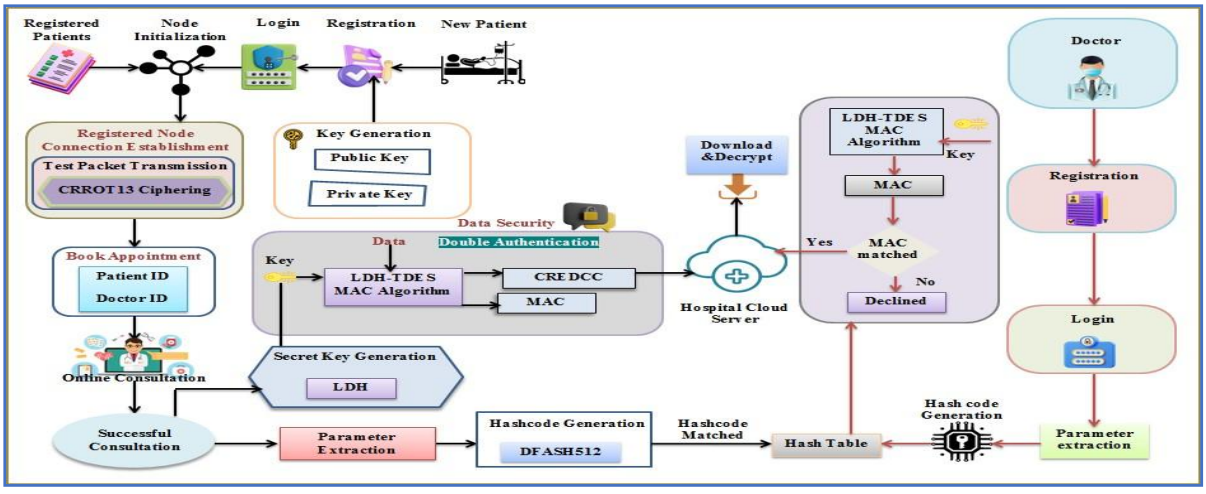
Anbarasan and Natarajan et a1. [4] introduced the B-DEAH (blockchain assisted delay and energy-aware healthcare monitoring system). A spotted hyena optimizer was used to implement cluster formation and cluster head selection. After that, cluster–based routing was created utilizing the MOORA method for Multi Objective Optimization based on Ratio Analysis. The simulation was conducted, in which the better performance of B-DEAH was proved. But the process led to high energy consumption.

Parah et al. [12] proposed a high payload together with reversible EHR embedding mechanism for the EHR to securely safeguard patient data and verify the received contents. The strategy was established on checksum calculations, Rivest Cipher 4 (RC4) encryption, Pixel Repetition Method (PRM) and Left Data Mapping (LDM). The strategy performed better than many cutting-edge methods, according to experimental findings. But, RC4 cannot be used in smaller streams of data.

III. THE PROPOSED MODEL

In this paper, a cryptographic authorization model for secure EHRs in healthcare cloud integrating MAC verification with hashing (CAEHRMH) is proposed. In this system, at first, patient registration is done and then the nodes are initialized. After that, the test packet is transmitted to create the authenticated path by calculating the cipher text using CRROT13. Then, the doctor's appointment is booked and the consultation is done. After the successful consultation, parameters are extracted and converted into hash code using DF-ASH512 while the secret key is generated using LDH. With this key,the MAC address is generated using the LDH-TDES-MAC algorithm.

Next, the data is converted into encrypted formats at a double time using CREDCC.



International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

FIGURE 1- PROPOSED MODEL

The MAC address is matched with the patient's MAC address. Upon matching, the doctor downloads the encrypted data and decrypts it. The proposed model's block diagram is displayed in FIGURE 1.

A. NODE INITIALIZATION

The proposed system starts with node initialization of the registered nodes. The registered patients are considered as nodes. If new patients come, the registration process must be done first. During registration, the patient enters their personal information, such as name, mail ID, age, etc. At the same time, the key generation centre generates the public (K_{pub}) and private (K_{priv}) keys. Then, the new patients log in by entering the username, password, and the generated key. After successful login, the new patients are considered as nodes. Now, the initialized nodes are expressed as,

$$P = \{ p_1, p_2, ..., p_n \} \quad (1)$$

Where, P denotes the total number of nodes and p_n is the n^{th} number of the node.

B. TEST PACKET TRANSMISSION

This phase explains the transmission of the test pack from one node to another node for making energy efficient network. In this operation, sender and receiver node id, hospital id, and public keys are used and combined to create cipher text. Then this cipher text is compared between the sender and receiver nodes. If the created cipher text of the sender node is the same as the receiver node, then both nodes are recognized as the registered node with the same hospital. Otherwise, the node is considered as a different network. Here, the cipher text by sender and receiver nodes is created using Caesar Reverse ROT-13 (CRROT 13) ciphering technique. Rotate by 13 places(ROT13) is not broken through frequency analysis or the exploitation of pattern words. The current ROT13 masks the text by substituting every single letter with the thirteenth letter of the alphabet. But, in the frequency analysis of cipher text for the corresponding plain text, the existing ROT-13 algorithm is easily broken. To overcome this shortcoming, the paper uses Caesar and the Reverse technique.

After arranging the alphabets with the corresponding cipher alphabets as per the ROT 13 model, depending on the value of the public key, the associated cipher alphabets are shifted. Then the original text □ is encrypted in the Caesar cipher technique by utilizing the below expressions:

$$\varepsilon(\psi) = (\psi + S) \bmod 26 \quad (2)$$

Here, $\varepsilon(\psi)$ represents the generated cipher text, and S denotes the shift value.

C. BOOK APPOINTMENT AND CONSULTATION

After the installation of the energy-efficient network, the patients can log in and book appointments with doctors and select the doctors with their appropriate specialty using the Patient ID

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

and Doctor ID. Following that, the online consultation will take place at the scheduled time. After completing the successful consultation, the parameters extraction, hash code generation, and secret key generation are carried out and they are explained in the following phases.

D. PARAMETER EXTRACTION

After the successful consultation, some important parameters are extracted for creating a network with high

security. Primary features such as patient id, doctor id, hospital id, consultation time, patient public key, doctor public key, consultation status, etc., are the ones extracted. The extracted features are expressed as,

$$E_m = \{E_1, E_2, E_3, E_M\} \quad (3)$$

Here, $E_m \in (P, \delta)$ denotes the number of extracted features, and δ denotes the doctor.

E. HASH CODE GENERATION

By utilizing the extracted parameters E_m, hash code is generated using DFASH512. The algorithm for secure hashing 512 (ASH512) algorithms is highly portable and produces identical hashes across all platforms. In the current hashing technique, hash values are generated based on the remainder of a division. This procedure results in a collision with hashing values. The Digit Folding technique is introduced in the ASH512 hashing algorithm to solve this problem. Here, the extracted parameters are taken as the input message and are converted to a message digest of length 512-bit to get the output. Initially, there are 1024-bit blocks, the input message is divided and the intermediate hash code is used to process individual blocks. The following are the steps involved in the process:

(1) APPEND THE PADDING BITS

The message applies padding, so that it equals 896 modulo 1024 in length. As a result, the total.no of the padding bits are in the order of 1 to1024.

(2) APPEND THE LENGTH

Following the padding process, a 128-bit block containing the message length (before padding), is added to the padded message.

(3) INITIALIZE MD BUFFER

The hash function's results are stored in a 512-bit buffer. Eight 64-bit registers are used to represent the buffer. These registers (R) are initialized as,

$$R = \{R0, R1, R2, R3, R4, R5, R6, R7\} \quad (4)$$

(4) PROCESSING MESSAGES IN 1024-BIT BLOCKS

For every round the algorithm receives two inputs: (i). 512-bit carry vector and (ii). 1024-bit message block. Through the conclusion of the l th stage, a message digest of 512-bit is generated by performing

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

out the ensuing steps:

STEP 1: MODIFICATION FUNCTION-1:

Every1024-bit length input block is subdivided into 128 sub-blocks, each of which contains 8 bits. The modification function is divided into two sub-functions.

a) **SUB-FUNCTION-1***:*

The Temp8[] undergoes a XOR operation with the input message's first eight-bit sub block and the obtained value is copied back into the Temp8 []. The above operation is repeated till the 1024-bit block message is exhausted.

b) **SUBFUNCTION-2:**

In order to generate a modified 1024-bit block, an XOR operation is performed with (i). every 8-bit unit of the 1024-bitblock and (ii)Temp8 [], where the obtained result is incremented by 1.

STEP 2: EXPANSION OPERATION-1:

By copying the contents of R0 to R7 into an array, 1024 bits are obtained. Then, Concatenation is expressed as follows:

$$R0\|R1\|R2\|R3\|R4\|R5\|R6\|R7\|R0\|R1\|R2\|R3\|R4\|R5\|R6\|R7\|$$
$$R0\|R1\|R2\|R3\|R4\|R5\|R6\|R7\|R0\|R1\|R2\|R3\|R4\|R5\|R6\|R7\| \quad (5)$$

The 2048 bit is converted into 1024 bit by using the digit folding technique is,

$$(R0,R0)+(R1,R1)+(R2,R2)+(R3,R3)+(R4,R4)+(R5,R5)+(R6,R6)+(R7,R7)+(R0,R0)+(R1,R1)+(R2,R2)+(R3,R3)+(R4,R4)+(R5,R5)+(R6,R6)+(R7,R7) \quad (6)$$

Here, || represents the concatenation operator.

STEP 3: 16-BIT XORED OPERATION:

The extended vector is XORed with the converted1024-bit block

STEP 4: EXPANSION OPEARTION-2:

The outcome of the previous step is split into 16 identical sub-blocks, each of which has 64 bits. Each sub block is subjected to two applications of the DES expansion table, resulting in 80 and 96 bits in the first and the second rounds. The other sub-blocks go through the same procedure.

STEP 5: SPLITTING PROCESS:

From the outcome of step 4, every 96-bit block derivedis further split into two equal 48-bit blocks

STEP 6: AREA CALCULATION PROCESS:

Each successive 48-bit block resulting from the previous step is split into three triangular points.

STEP 7: MODIFICATION FUNCTION-2

The area of the triangle is converted to binary numbers. The resulting bits are analysed and the number of bitsif below 16, then zeroes are appended to the left side of the resultant bits. The new 16-

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

bit block is sub-divided into 4 sub- blocks, each 4-bits in length. After this sub division, every 4-bit sub-block is XORed with Temp4 []. This process is repeated for the rest of the sub-blocks. Finally, we get hexadecimal numbers.

F. OUTPUT

If the message is a single 1024-bit block, then the message digest is created by copying in order each hexadecimal digit into the carry vector. If not, previous carry vector is added to modulo 512, and the result is used as input to the hash function. From this process, the obtained hash code is denoted as $(H_{E_m})^{pat}$ and it is stored in the hash table. The pseudo code for the proposed-work **DFASH512**:

Input: Extracted parameters E_m,
Output: Hash Code $(H_{E_m})^{pat}$

Begin
Initialize MD buffer R
Append padding bits to 1024
Append length
Process messages in 1024-bit blocks
For $1 to l^{th}$ *stage*
Divide into 128 sub blocks
Apply Sub function-1
For all the sub blocks, **do**
8 bit sub block $\oplus$ Temp8 []
End for
Apply Sub function-2
For each sub block, **do**
Increment by 1
End for
Apply expansion function-1
Concatenate the registers to 2048 bits
For every 16 bits **{**
Perform XOR
}
Divide as 16 sub blocks
For 1 to 16 sub blocks
While $z = 1st\ subblock$
Apply DES for twice
At first {
Produce 80 bits
}
At second {
Produce 96 bits
}
End while
End for
Divide as two 48 bit blocks
For 1st 48 bit block

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

Divide triangle point

End for

Convert triangle into binary numbers b

If $(b<0)$ **{**

Add 0 to the left side

} else {

No need to add 0

}

Perform XOR operation

Obtain 16 bits

Return $\left(H_{E_m}\right)^{pat}$

End begin

G. SECRET KEY GENERATION

This phase explains the generation of the secret key by using the technique of LDH. At the time of exchanging data over a network, a shared secret key is used for secret communication. Diffie Hellman algorithm is most used to add perfect forward secrecy to secure communication. The major issue in Diffie Hellman is that both the sender and the receiver can exchange the secret user key among them. As a result, during the user key exchange the attackers may intrude and find out the secret key. Hence the log-based public key updation is performed instead of exchanging the user key. Here, the input is considered as the doctor's public key $\left(\kappa_{pub}{}^{*}\right)$. Here, an elliptic curve is used for generating points and getting a secret key using the parameters. The elliptic curve is defined as:

$$x^3 = u^3 + uy + w \tag{7}$$

Where, u, w are integers, x, y denotes the parameters that define the function. The receiver's private key will be used to decrypt the data after it has been encrypted by the authorised user using the receiver's public key. To encrypt and decrypt, the private and public keys are generated. Initially, the private keys for both patient and doctor are chosen randomly over the curve. By considering the private keys, public keys for both of them are calculated.

The equation used to generate the public key is:

$$\Gamma = \eta * R \tag{8}$$

Where, Γ denotes the public key, R denotes the private key that was generated randomly, and η is the curve's pivotal point. Based on selected parameters, the secret keys (μ) are generated as:

$$\mu = log\left(\kappa_{pub}{}^{*}\right)^{R} mod(\Gamma) \tag{9}$$

$$\mu = log\left(\kappa_{pub}\right)^{R} mod(\Gamma) \tag{10}$$

The secret keys of the patient and doctor are further used for the MAC address generation and it is explained in the below steps.

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

H. DOUBLE ENCRYPTION

(1) MAC ADDRESS GENERATION WITH TDES-BASED ENCRYPTION

Continuing the secret key generation, the MAC address is generated using the LDH-TDES-MAC algorithm. A MAC algorithm provides data-origin authentication and data integrity protection. There is no proof that the message is actually sent by the sender. So, in order to solve the problem and improve data security Log Diffie Hellman TDES algorithm is used. Generally, the MAC is generated using the secret key and input message. Therefore, the generated secret key (μ) and the consultation data (D) are taken as input.

For the purpose of generating a MAC address, the input data is encrypted using Triple Data Encryption Standard(TDES) algorithm. TDES prevents a meet-in-the-middle attack. Existing TDES cipher experiences a significant instability due to its short (64-bit) block size, which is the encryption size for plaintext. In order to solve this problem, Log Diffie Hellman-based secret key is used.

TDES was created by implementing the DES algorithm three times throughout the encryption process. TDES contains **3** 168-bit keys (**3** 56-bit keys that is generated 168-bit secret key). The TDES algorithm is split into three steps, and in every step the DES algorithm is implemented. DES has 16 rounds of blocks. Here, the input is 64-bit size plain text and the output is a cipher text of 64-bits size. A 56-bit external-key (μ_{56}) is used in all 16 rounds. The incoming block of plaintext first passes through an initial permutation (IP) and then it is divided into two 32-bit halve, 32 right bits (D_{right}), and 32 left bits (D_{left}). To expand the right bits to 48 from 32, duplication is performed using the expansion permutation. Then, it is combined with the external key (μ) using XOR operation and is expressed as:

$$C = (\mu_{56}) \oplus (D_{right}) \quad (11)$$

Here, C represents the combined bits. Then, the output is further split into eight 6-bits and then supplied into the eight substitution boxes. According to a non-linear transformation, the 6 input bits are replaced with the 4 output bits. The outputs are passed through a straight permutation. The result is combined with the (D_{left}). This is the complete process of one round. In this, the right bits are taken as the left bits and for the next round. After the 16th iteration, the right and left bits are concatenated and then passed through a final permutation. The output is the encrypted data (D_{encry}), which is obtained from the first step of the TDES algorithm. The second step involves utilizing a second external key to operate on the first pre-ciphertext and encrypt it with DES algorithm to produce the second pre-cipher text. The third step involves utilizing a third external key to operate on the second pre-cipher text and encrypt it with DES algorithm, resulting in cipher text (D_{encry}). Following

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

the encryption, the MAC address generation process is carried out. MAC is a security code that is appended to the message sent by the sender to the receiver for providing message authentication and integrity The MAC of the patient side $m_{patient}$ is expressed as:

$$m_{patient} = F(\mu, D_{encry}) \quad (12)$$

Here F, represents the MAC function. The pseudo-code of the proposed LDH-TDES-MAC is,

Input: Doctor's public key κ_{pub}^{*}

Output: MAC address $m_{patient}$

Begin
Initialize u, w, x, y
Define Elliptic curve
Choose random point R
If (R =finite field) **{**
Consider as private key
} Else {
Choose random point
}
Compute public key Γ
Evaluate secret key (μ)
Generate MAC address
For 1st DES stage
For $(i = 1 to 16\ rounds)$
While $i = 1$
Pass the input through IP
Divide 64 bits into two 32 bit
Expand right side bit to 48
Apply secret key (μ)
Perform XOR operation
$C = (\mu_{56}) \oplus (D_{right})$
Divide the output into eight 6-bit
For $j = 1\ to eight\ 6\ bit$
While $j = 1st\ 6\ bit$
Replace with the 4 output bits
End while
End for
Pass the output through SP
Combine them with (D_{left})
End while
End for
End for
Generate MAC address
Return $m_{patient}$
End begin

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

(2) CREDCC-BASED ENCRYPTION

In this phase, the data is again encrypted (D_{encry}) using CREDCC. Elliptic Curve Cryptographic (ECC) algorithm operates on a small set of points on an elliptic curve to provide digital signatures that are highly efficient and secure. The point selection in the traditional elliptic curve is from one line. Hence, the attacker may choose that point randomly. So, this research methodology introduces the Edward curve instead of the elliptic curve process. The base point is selected by the closed formation of the curve with addition and doubling law. Also, to reduce the complexity problem for the big numbers, this research methodology uses the cube root calculation, which also improves the security level. The CR-EDCC is used as the cryptographic key algorithm, where the Edward curve with the cube root calculation is defined as:

$$q^3 + u^3 = 1 + \sqrt[3]{dq^2u^2} \quad (13)$$

Where, $q, u \in D_{encry}$ denotes the scalar parameters. The user and the receiver can encrypt and decrypt the data, with the user using receiver's public key and the receiver using its own private key. Encryption and Decryption is done by the generated private and public keys. The equation used to generate the public key is:

$$\kappa_{pub} = \Phi * \kappa_{pri} \quad (14)$$

Where, (κ_{pub}) denotes the public key, κ_{pri} denotes the private key that is generated randomly, and Φ is the curve's pivot point. Now, the input data is encrypted. During the encryption process, the secret key is added by the encryption formula. The encrypted data consists of two cipher texts, which can be defined as:

$$\beta_1 = (\Phi * \Re) \quad (15)$$

$$\beta_2 = (\Gamma_{\chi o \mu} + \vartheta * \Re) \quad (16)$$

Where, $\Re$ is a random number of the range (1, n-1), β_1 and β_2 are the two ciphertexts. Thus, by double encryption, the data can be preserved from attacks. Then finally the encrypted data is stored in the cloud server for further use.

I. DOCTOR REGISTRATION, PARAMETER EXTRACTION AND HASHCODE GENERATION

After completing the process of storing the data about the patient securely, the doctor logs in to the system using the generated private and the public key. After succeeding in the registration process, the necessary parameters, such as patient id, doctor id, hospital id, consultation time, patient public key, doctor public key, consultation status, etc. are extracted. The extracted parameters are further converted into the hash code using the DFASH512 algorithm, which is mathematically

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

elaborated in section 3.5. The generated codes are stored in the hash table. If the hash of the doctor $\left(H_{E_m}\right)^{doc}$ is matched with the patient's hash code, MAC verification is performed.

J. MAC VERIFICATION

In this phase, the generated MAC address at both sides of the patient and doctor are compared. If they match data is downloaded from the cloud. For this purpose, the MAC address for the doctor is created using the LDH-TDES-MAC algorithm. Then, the MAC address is compared with the patient's MAC address m_{doctor} for authentication. When the MAC address is also matched, the system allows the user to decrypt and download data from the hospital's private cloud data center. The data is then evaluated.

$$\wp = \begin{cases} authenticated\ data, if\ m_{patient} = m_{doctor} \\ unauthenticated\ user, if\ m_{patient} \neq m_{doctor} \end{cases} \quad (17)$$

Here, $\wp$ denotes the MAC comparison for downloading data. The data is decrypted double time using CREDCC

and TDES is,

$$D_{encry} = \beta_2 - \left(\kappa_{pub} * \beta_1\right) \quad (18)$$

$$D = (D_{encry}, \mu) \quad (19)$$

Here, d represents the decryption function. After obtaining the desired data the performance of every novelty has been evaluated with some quality metrics in the following results section.

IV. RESULTS AND ANALYSIS

In this section, the execution of the suggested CAEHRMH system's performance is experimentally analysed and compared to cutting-edge methodologies while operating on the platform of PYTHON with SQLite.

A. DATABASE DESCRIPTION

The proposed system works using five different disease datasets from https://www.kaggle.com/datasets, namely Breast Cancer Wisconsin (Diagnostic) Data Set (DS-I), Diabetes Dataset (DS-II), Heart Disease Dataset (DS-III), Obesity Dataset (DS-IV), and Polycystic Ovary Syndrome (PCOS) (DS-V).

B. PERFORMANCE ANALYSIS OF HASH CODE GENERATION

Based on hash code generation time, the proposed DFASH512's is compared to that of the existing ASH512, Secure Hash Algorithm 256 (SHA 256), Message- Digest 5(MD5), and SHA 1 to demonstrate the superiority of the proposed model.

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

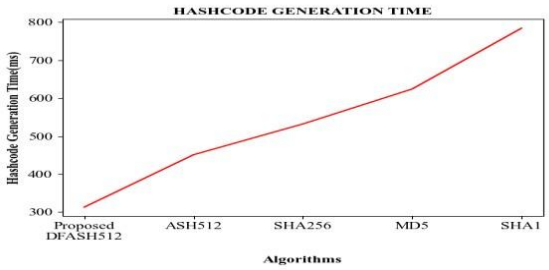


FIGURE 2: GRAPHICAL REPRESENTATION OF THE HASH CODE GENERATION TIME

FIGURE 2 unveils the performance assessment of the proposed and the existing methods. The proposed work takes less time than the compared models which indicates higher performance. According to FIGURE 2, the time taken for hash code generation is 313 ms, which is too lower than the existing ASH512 (412ms), SHA256 (532ms), and MD5 (624ms). From this analysis, it can be concluded that the modification in the existing model works out very well.

C. PERFORMANCE ANALYSIS OF CREDCC

The proposed CREDCC's performance is analysed andcompared with existing techniques like ECC, RSA, ElGamal and DES.

TABLE 1: (a) ENCRYPTION TIME ANALYSIS

Techniques	DS-I	DS-II	DS-III	DS-IV	DS-V
Proposed CREDCC	1147	903	705	2032	1137
ECC	2014	1041	854	3015	1954
RSA	3014	1145	926	4215	3065
DES	4015	1254	1098	5184	4047
ElGamal	5201	1302	1175	6074	5247

TABLE 1: (b) DECRYPTION TIME ANALYSIS

Techniques	DS-I	DS-II	DS-III	DS-IV	DS-V
Proposed CREDCC	1254	934	814	2145	1198
ECC	2478	1051	952	2451	2354
RSA	3125	1161	1062	3145	3002
DES	4154	1291	1195	4214	4114
ElGamal	5468	1380	1315	5614	5514

The results of the evaluation of the proposed work and the existing models are represented in TABLE 1. The proposed work takes less time than the compared models which indicate the better performance of the proposed work. As per that, the encryption time of the proposed model when the system process with the DS-I is 1147ms, which is 867ms more than ECC, 1867ms improved than RSA, and 2868ms improved than DES. Likewise, the proposed model's decryption time is 1254 for DS-I. This gives 1224ms faster than existing ECC and 1871ms faster than RSA. Likewise, the encryption time and decryption time for the remaining dataset is analyzed. From this analysis, it can be proved

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

that the proposed model provides faster and more efficient performance.

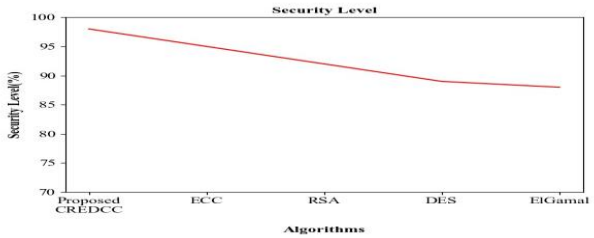


FIGURE 3: SECURITY LEVEL ANALYSIS

FIGURE 3 gives analysis of security level while encryptingthe data using proposed as well as existing model. The higher value of proposed model shows higher security. For instance, security level obtained from the proposed model is 98%, which is greater than the already existing models, like: ECC (95%), RSA (92%), DES (89%), and ElGammal (88%). Hence, the proposed CREDCC system transfers data more securely when compared to the existing techniques.

TABLE 2: (a). MEMORY USAGE ON ENCRYPTION

Techniques	DS-I	DS-II	DS-III	DS-IV	DS-V
Proposed CREDCC	149874545	121474512	136454544	152154545	148755778
ECC	155454645	131455455	142144455	162454545	155471255
RSA	165784541	142145454	152457845	173453453	163124874
DES	175784646	155454554	163155444	189874562	176478542
ElGamal	199547845	173145445	185478544	208579658	196974454

TABLE 2: (b). MEMORY USAGE ON DECRYPTION

Techniques	DS-I	DS-II	DS-III	DS-IV	DS-V
Proposed CREDCC	148755778	129874521	138755778	156547812	148755778
ECC	154878789	137824564	144878789	166478455	155471255
RSA	165847752	146598745	155847752	175745545	163124874
DES	176547812	159658745	165454845	189874562	176478542
ElGamal	195784412	176547844	184574455	208579658	196974454

TABLE 2: (c). MEMORY USAGE ON ENCRYPTION OVERHEAD IN BITS

Techniques	DS-I	DS-II	DS-III	DS-IV	DS-V
Proposed CREDCC	120.35475	21.654	35.454	254.35478	120.3547
ECC	121.64577	22.2454	36.14542	256.147545	121.65474
RSA	122.565454	22.8975	37.4887	256.98745	122.2451
DES	122.65474	23.7548	38.8547	257.65474	122.65445
ElGamal	122.7845451	23.9874	38.98445	257.91424	123.2544

Table 2 gives the performance of the currently available methods vs the proposed model by analysing the memory usage of encryption and decryption. Lower usage shows the efficient performance of the

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

model. When considering the Diabetes Dataset encryption, the proposed model uses 121474512kb memory spaces. It takes 99,80,943kb lower memory than Existing ECC. When considering the Diabetes Dataset encryption, the proposed model uses 129874521kb memory spaces. It takes 79,50,043kb lower memory than existing ECC. Likewise, memory usage for encryption and decryption is analyzed and compared with the existing model. The encryption overhead in bits of the proposed model is 120.35475kb. So, from these results, it can be confirmed that the proposed model occupies only lower memory than the conventional models.

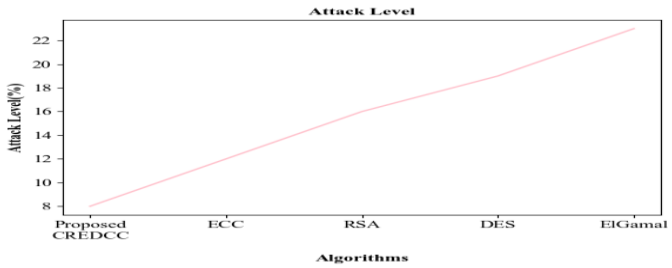


FIGURE 4: PERFRORMANCE ANALYSIS BASED ON ATTACK LEVEL

The levels of attack for the proposed and current models is represented in FIGURE 4. The attack level of the proposed model is 8%. But, the existing ECC attains 12%, RSA attains 16%, and DES attains 19%. Therefore, the proposed model has lower data attacks when compared to the existing models.

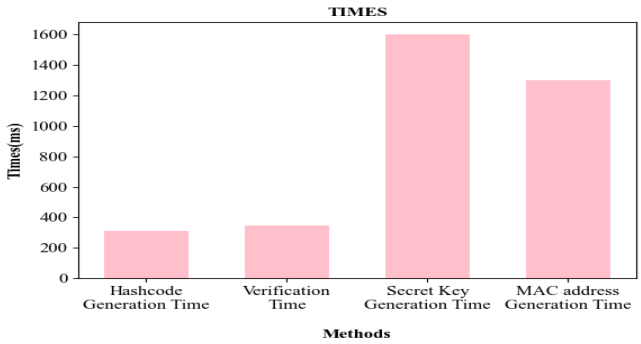


FIGURE 5: PROPOSED MODE'S TIME ANALYSIS

In FIGURE 5, the time analysis for each step of the proposedsystem is shown. The time taken for hash code generation is 300ms, for verification is 354ms, for secret key generation is 1600ms, and for MAC address generation it is 1300ms. These are highly improved than the existing model by modified steps of the proposed model.

D. COMPARATIVE ANALYSIS BASED ON OBJECTIVES

International Journal of Multidisciplinary Research in Arts, Science & Commerce (IJMRASC) ISSN Online: 2583-018X

Vol. 4(1), March 2024, pp. 48-67

This section performs the comparison between our proposed LDH-TDES-MAC, Thermal-aware duty cycle MAC protocol (ThMAC) proposed by Monowar et,al [11]), local coordination X-MAC (LCX-MAC) proposed by (Ahmad et al.,[2]), and adaptive switching (AS) proposed by (Preveze et al., [13]).

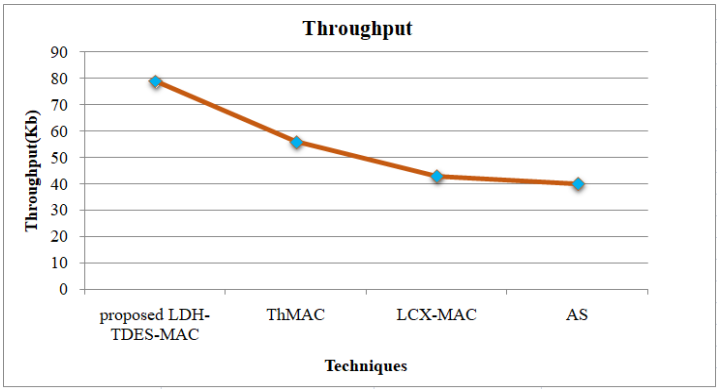


FIGURE 6: ANALYSIS OF PERFORMANCE BASED ON THROUGHPUT

In FIGURE 6, the throughput performance of proposed and the existing models are analyzed. Throughput can be described as the bulk of data successfully transmitted to the end node in a given time. The higher the throughput, higher the performance. In this proposed methodology, the data is encrypted twice and securely stored in the cloud. Also, the MAC address verification is considered by comparing the patient and doctor sides. Hence, decrypting the data by the intermediate is difficult. The throughput of the model is increased to a greater extent by MAC verification, which improves the throughput (79kb) of the proposed work. From this, we can infer that the proposed work performs well than the prevailing methods.

V. CONCLUSION

This work has proposed a cryptographic authorization model for secure EHRs in healthcare cloud integrating MAC verification with hashing (CAEHRMH). The proposed method undergoes the following processes: registration, node initialization, test pack transmission, book appointment, consultation, parameter extraction, hash code generation, secret key generation, MAC address generation, and verification. After executing all the steps, the performance analysis of the CAEHRMH model (that combines the proposed LDH-TDES- MAC, CREDCC and DF-ASH512) was performed and compared with the available techniques in terms of various metrics. The final outcomes reveal that the proposed model achieves a security level of 98%. Similarly, the proposed model achieves the best result for all other metrics, such as encryption time, decryption time, attack level, etc. Hence, from the results of various metrics, the proposed CAEHRMH is found to be more efficient for EHRs security. This work has less concentration on data integrity. Since the Data Owner (DO) is also not fully

trustworthy, DO may maliciously forge the result of data integrity. So, in order to solve this problem, in the future, Log file-based data integrity verification and partial private key-based data security will be introduced.

REFERENCES

1. Abdellatif, A. A., Al-Marridi, A. Z., Mohamed, A., Erbad, A., Chiasserini, C. F., & Refaey, A. (2020). SsHealth: Toward secure, blockchain-enabled healthcare systems. IEEE Network, 34(4), 312–319. https://doi.org/10.1109/MNET.011.1900553.
2. Ahmad, A., Ullah, A., Feng, C., Khan, M., Ashraf, S., Adnan, M., Nazir, S., & Khan, H. U. (2020). Towards an improved energy efficient and end- to-end secure protocol for iot healthcare applications. Security and Communication Networks, 1-10. https://doi.org/10.1155/2020/8867792.
3. Ambarkar, S. S., & Shekokar, N. (2020). Toward smart and secure IoT based healthcare system. In Studies in Systems, Decision and Control, 266. 283-303.https://doi.org/10.1007/978-3-030-39047-1_13.
4. Anbarasan, H. S., & Natarajan, J. (2022). Blockchain based delay and energy harvest aware healthcare monitoring system in WBAN environment. Sensors, 22(15), 1–29. https://doi.org/10.3390/s22155763.
5. Chehri, A. (2021). Energy-efficient modified DCC-MAC protocol for IoT in e-health applications. Internet of Things, 14, 1-22. https://doi.org/10.1016/j.iot.2019.100119.
6. Chinnasamy, P., & Deepalakshmi, P. (2022). HCAC-EHR: Hybrid cryptographic access control for secure EHR retrieval in healthcare cloud. Journal of Ambient Intelligence and Humanized Computing, 13(2), 10011019.https://doi.org/10.1007/s12652-021-02942-2.
7. Denis, R., & Madhubala, P. (2021). Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems. In Multimedia Tools and Applications, 80(14),21165-21202. https://doi.org/10.1007/s11042-021-10723-4.
8. Kim, M., Yu, S., Lee, J., Park, Y., & Park, Y. (2020). Design of secure protocol for cloud-assisted electronic health record system using blockchain. Sensors, 20(10), 1–21. https://doi.org/10.3390/s20102913.
9. Limbasiya, T., Sahay, S. K., & Sridharan, B. (2021). Privacy-preserving mutual authentication and key agreement scheme for multi-server healthcare system. Information Systems Frontiers, 23(4), 835–848. https://doi.org/10.1007/s10796-021- 10115-x.
10. Misra, S., Bishoyi, P. K., & Sarkar, S. (2020). i-MAC: In-Body Sensor MAC in wireless body area networks for healthcare IoT. IEEE Systems Journal, 15(3), 4413–4420. https://doi.org/10.1109/jsyst.2020.3020306.
11. Monowar, M. M., & Alassafi, M. O. (2020). On the design of thermal- aware duty-cycle mac protocol for iot healthcare. Sensors, 20(5), 1-20. https://doi.org/10.3390/s20051243.
12. Parah, S. A., Kaw, J. A., Bellavista, P., Loan, N. A., Bhat, G. M., Muhammad, K., & De Albuquerque, V. H. C. (2021). Efficient security and authentication for edge-based internet of medical things. IEEE Internet of Things Journal, 8(21), 15652–15662. https://doi.org/10.1109/JIOT.2020.3038009.

13. Preveze, B., Alkhayyat, A., Abedi, F., Jawad, A. M., & Abosinnee, A.S. (2022). SDN-Driven internet of health things: A novel adaptive switching technique for hospital healthcare monitoring system. Wireless Communications and Mobile Computing, 1-11.https://doi.org/10.1155/2022/3150756.
14. Saba, T., Haseeb, K., Ahmed, I., & Rehman, A. (2020). Secure and energy-efficient framework using internet of medical things for e- healthcare. Journal of Infection and Public Health, 13(10), 1567–1575. https://doi.org/10.1016/j.jiph.2020.06.027.
15. Sharmila, A. H., & Jaisankar, N. (2020). E-MHMS: enhanced MAC- based secure delay-aware healthcare monitoring system in WBAN. Cluster Computing, 23(3), 1725–1740. https://doi.org/10.1007/s10586-020-03121-2.
16. Sowjanya, K., Dasgupta, M., & Ray, S. (2021). A lightweight key management scheme for key-escrow-free ECC-based CP-ABE for IoT healthcare systems. Journal of Systems Architecture, 117, 1-10.https://doi.org/10.1016/j.sysarc.2021.102108.
17. Uddin, M., Memon, M. S., Memon, I., Ali, I., Memon, J., Abdelhaq, M., & Alsaqour, R. (2021). Hyperledger fabric blockchain: Secure and efficient solution for electronic health records. Computers, Materials and Continua, 68(2), 2377–2397.https://doi.org/10.32604/cmc.2021.015354/.
18. Wang, X., & Cai, S. (2020). Secure healthcare monitoring framework integrating NDN-based IoT with edge cloud. Future Generation Computer Systems,112, 320–329. https://doi.org/10.1016/j.future.2020.05.042.
19. Yaacoub, J. P. A., Noura, M., Noura, H. N., Salman, O., Yaacoub, E., Couturier, R., & Chehab, A. (2020). Securing internet of medical things systems: Limitations, issues and recommendations. Future Generation ComputerSystems,105,581-606.https://doi.org/10.1016/j.future.2019.12.028.
20. Yang, G., Li, C., & Marstein, K. E. (2021). A blockchain-based architecture for securing electronic health record systems. Concurrencyand Computation: Practice and Experience, 33(14), 1-10.