International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-25
https://doi.org/10.70917/ijcisim-2026-2034
Article

An Efficient Blockchain-Enabled MAES-RAC Framework for Secure Storage and Controlled Access of Healthcare Data

Sanketi Raut[1], Rahul Thour[2]

[1] Department of Computer Science & Engineering, Desh Bhagat University, Punjab, India. sanketiraut28@gmail.com
ORCID: 0009-0007-1511-8019
[2] Department of Computer Science & Engineering, Desh Bhagat University, Punjab, India. t.rahul@deshbhagatuniversity.in
ORCID: 0009-0008-7688-1998

Abstract: In healthcare, secure transmission and access control of medical data are critical. While various systems have been introduced for secure medical data storage and access, existing healthcare systems often rely on public cloud storage for patient data collection and storage. Although convenient, these approaches introduce significant challenges, including reduced security, higher computational load, low scalability, and increased encryption and decryption times. Therefore, this research proposes a Modified Advanced Encryption Standard and Rule-based Access Control (MAES-RAC) framework to address these drawbacks and provide secure medical data storage and access. The MAES-RAC framework enhances security through the incorporation of a Modified Advanced Encryption Standard (MAES) algorithm, which accelerates the encryption process with lower computations and offers improved security. Moreover, the Modified Rule-based Access Control (MRAC) mechanism ensures security by integrating the Bell-LaPadula (BLP) model, preventing unauthorized data access. Additionally, employing blockchain technology to store medical data enhances the scalability and interoperability of the MAES-RAC framework. Experimental results demonstrate that the MAES-RAC framework achieves an encryption time of 1.44ms, a decryption time of 1.23ms, memory usage of 394.6KB, a genuine user rate of 0.931, and a responsiveness of 4.255ms when using a healthcare management system dataset. .

Keywords: Medical data, cloud computing, Blockchain, Data security, Access Control mechanism.

1. Introduction

Healthcare and medical data play a significant role in patients' lives, enabling the identification of patterns, a better understanding of medical issues, and more effective disease diagnosis and treatment [1]. This leads to improved patient satisfaction through personalized care and efficient management of chronic conditions. The rapid advancement of technology and the increasing digitalization of the medical field have led to a progressive rise in both the volume and variety of medical data [4]. When the need to manage patient medical data arises, only authorized individuals should be able to access it within the network [2, 5]. However, conventionally stored medical data is often subject to modification and theft by unauthorized persons, making data security a key challenge for healthcare organizations [3, 7]. Consequently, there is a need for automated medical data storage to eliminate complications in data storage and exchange between healthcare providers [4]. Electronic Health Records (EHRs) are among the most widely used resources in the healthcare sector, providing a comprehensive interpretation of a patient's medical status and encompassing all medical-related data and medical history [5]. EHRs are generally created and shared among doctors and nurses via cloud computing, offering a more convenient method for managing this type of medical data [6]. However, storing patients' medical data in the cloud has introduced new challenges related to authentication, data access control, authorization, and compliance [7]. This medical data encompasses a broad range of attributes, including genomic data, real-time patient monitoring data, imaging data, and medical histories. It is crucial to protect this data from cyber-attacks, unauthorized access, and breaches [8], as a lack of data

security in healthcare organizations can result in significant losses [9]. Today, various types of hackers are emerging who can alter patient data and introduce dangerous malware into hospital computer systems, rendering the data inaccessible [10]. Therefore, there is an urgent need to develop a system to secure the entire healthcare organization. Conventional healthcare systems rely on deep learning and machine learning algorithms, which often face challenges in secure cloud storage, continuous data sharing among healthcare providers, and effective data transmission. The effectiveness of traditional ML approaches is limited by the diversity of medical data and the inherent complexities they encounter [11,13]. Similarly, while DL approaches, such as recurrent neural networks and convolutional neural networks, excel at identifying patterns in medical images and medical sensor data, they also exhibit computational issues, particularly in the area of secure data storage [12].

The MAES-RAC (Modified Advanced Encryption Standard and Rule-based Access Control) framework is proposed to securely store and access patients' healthcare data or medical files, overcoming the notable drawbacks of existing methods. The medical files are securely stored using the MAES algorithm, with the key generated from a key generation center. A Break Glass Key Access (BGKA) mechanism generates a key break auxiliary message, used to extract the break glass key (BGK) to decrypt the medical files. The Emergency Access Control (EAC) mechanism authenticates the Emergency Contact Person (ECP). The other main contribution of this research is described as follows below.

The MAES algorithm introduces a modification in the mix columns step, executed by reordering the fixed matrix's columns based on the rank value. This modification reduces computation time and speeds up the encryption process. Moreover, the MAES algorithm overcomes issues of computational overhead and offers improved security. MRAC, an access control formed through the combination of role and attribute-based access control, checks the privacy policy. The BLP model is integrated with the MRAC mechanism to validate the privacy policy based on BLP properties and grant access to the data requester. This MRAC mechanism reduces the risk of intentionally or accidentally revealing healthcare data. Additionally, the incorporation of blockchain in the MAES-RAC framework boosts the overall security level in storing medical files. An in-depth security review of the revised algorithm of AES is performed to show diffusion strength, resistance to widespread cryptographic attack and retention of AES security level.

This research article is organized as follows: Section 2 covers the literature review and challenges, Section 3 presents the system model, Section 4 details the overall methodology of the MAES-RAC framework, and Section 5 describes the results and discussion. The research concludes with a conclusion and future work in Section 6.

2. Literature Survey

Ali, A. et al. [1] proposed a permission-based blockchain approach to enhance healthcare system security. This approach improved both security and data privacy through the use of standard encryption and biometric authentication methods. Furthermore, it addressed the scalability limitations of conventional blockchain technology. However, it did not effectively manage large volumes of healthcare data. Sangeetha, S.B. et al. [3] introduced a robust framework called the Secure Healthcare Access Control System (SHACS) to improve healthcare effectiveness and security. Integrating SHACS with various healthcare administrations could increase user satisfaction and reduce reputational and legal risks, but it required significant computational resources.

Oh, J. et al. [4] presented a secure medical data-sharing approach aimed at balancing privacy and data confidentiality. By using Key Aggregate Encryption (KAE), the approach enhanced data security by minimizing the risk of unauthorized disclosures and data breaches. However, the implementation of a homomorphic encryption algorithm resulted in a high computational load on each unit. Kala, M.K. and Priya, M. [5] developed a Blockchain-based Decentralized Safe Sharing (BDSS) system using the Shuffled Random Starvation Link Encryption (SRSLE) method to secure Personal Health Records (PHR). Complex PHR cases were organized and analyzed using a Quasi-sensitive attribute Identification approach. While the proposed BDSS approach could be efficiently implemented in a real-world medical environment, it suffered from limitations such as scalability issues, high execution costs, and interoperability challenges.

Ryu, J. and Kim, T. [6] developed an authentication protocol to secure medical data within healthcare systems. Their approach encrypted medical data and records on the blockchain, allowing for secure exchange between hospitals. The scheme utilized an elliptic curve cryptography (ECC)-based private-public key system, simplifying the encryption process for data transmission. However, it did not store the encrypted medical data on the blockchain itself. Almalawi, A. et al. [7] introduced a Lionized remora optimization-based serpent (LRO-S) encryption approach to protect healthcare data from hackers and unauthorized access. Sensitive medical data was

encrypted using LRO-S before being stored in the cloud, and an asymmetric hash signature approach was used to enhance security. While the LRO-S approach exhibited lower encryption and decryption times and improved confidentiality, it incurred higher cost consumption.

Reegu, F.A.et al.[8] proposed an interoperable blockchain-based EHR (BCIF-EHR) to secure health data. The BCIF-EHR framework used wallets and cloud agents to store EHR data, while blockchain technology was used to store automatically verifiable data and maintain public key signatures. However, the BCIF-EHR framework has limitations: it cannot be applied to all groups contributing to the EHR scheme and is difficult to implement effectively. Da Costa, L.et al.[9] proposed Sec-Health, a blockchain-based procedure designed to secure patients' medical records. Decentralized networks and attribute-based cryptography were used to secure storage and ensure access control, integrity, and confidentiality. However, the Sec-Health approach suffers from high memory overhead and longer execution times.

2.1 Challenges

- While the approach suggested in [1,14] eliminates traditional scalability issues and enhances transaction performance, it fails to address the computational challenges posed by the large volume of healthcare data.
- Although the SHACS approach suggested in [3,15] strengthens the security of healthcare systems, the increased number of users and growing volume of medical data lead to latency and computational overhead issues.
- The LRO-S approach suggested in [7,17] minimizes encryption and decryption time and improves the confidentiality rate. Nevertheless, it increases cost consumption and execution time.
- While the Sec-Health approach suggested in [9,18] highly secures data storage and access, it exhibits memory overhead issues and increases overall execution time.

2.2 Research Gaps:

Despite significant advancements in secure and scalable healthcare systems, current methods still face critical limitations in real-world, data-intensive environments. Prior research reveals unresolved issues concerning computational efficiency, scalability, and resource utilization, leading to the following research gaps:

- The method in [1,14], while improving scalability and transaction speed, struggles with the heavy computational load of large and continuously growing healthcare datasets.
- The SHACS approach in [3,15], although enhancing system security, suffers performance degradation with an increasing number of users and medical records, resulting in noticeable delays and higher processing overhead.
- The LRO-S technique in [7,17], while strengthening data confidentiality and reducing cryptographic time, incurs higher costs and longer overall execution times.
- The Sec-Health model in [9,18], despite ensuring strong data protection, exhibits increased memory and processing time requirements that limit its efficiency in large-scale healthcare systems.

3. System Model

Figure 1 illustrates the system model for the MAES-RAC framework. In an emergency, a patient's medical data should be accessed only by authorized personnel over the network, ensuring the data's privacy and security. However, data security remains a key challenge for medical institutions. To address this, patients first encrypt their medical data using a secure cryptographic algorithm. Simultaneously, auxiliary messages are generated using the BGKA mechanism. Both the encrypted file and the auxiliary messages are then securely stored. In the event of an emergency, an emergency contact person (ECP) can request access to the data. After undergoing various authentication checks, the encrypted medical file is decrypted and provided to the ECP. This process mitigates security challenges related to accessing and storing medical data.

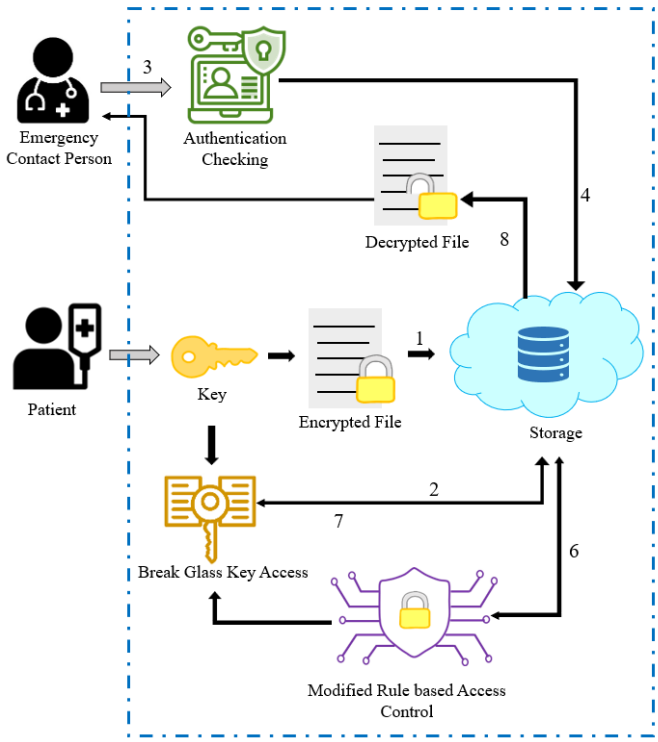


Figure 1. System Model

4. Modified Advanced Encryption Standard and Rule-based Access Control framework for secure storing and accessing of medical data

This research aims to securely store and access medical data while maintaining access control. To achieve this, we propose a MAES-RAC framework for secure storage and access. The MAES-RAC framework includes several phases: initialization, registration, login and ECP assignment, data storage, and data access. During the registration phase, both patients and ECPs register their details (e.g., identity, passwords, and biometrics) in the medical institute database. After successful registration, patients log in using their registered credentials and assign ECPs to access their medical files in case of an emergency. The system automatically rejects login attempts made with invalid credentials. After ECPs are assigned, keys are generated by the key generation center during the data storage phase. Simultaneously, the BGKA framework generates the break-glass key auxiliary message. The patient's medical file is then encrypted using the MAES algorithm, and both the encrypted file and the break-glass key auxiliary message are securely stored on the blockchain. If an emergency arises after successful data storage, the ECPs request access to the patient's medical file through the Emergency Access Control (ECA) mechanism. If access is granted, a request is sent to the blockchain. The blockchain then requests the MRAC mechanism to verify the privacy policy. If the policy is valid, the MRAC mechanism grants permission to the blockchain. Subsequently, the blockchain requests the BGKA framework to extract the break-glass key using the generated auxiliary message. The BGKA framework verifies the auxiliary message and provides a BGK to the blockchain. Upon receiving the key, the same MAES algorithm is used to decrypt the patient's medical file, which is then provided to the ECPs for review. Figure 2 illustrates the block diagram of the MAES-RAC approach.

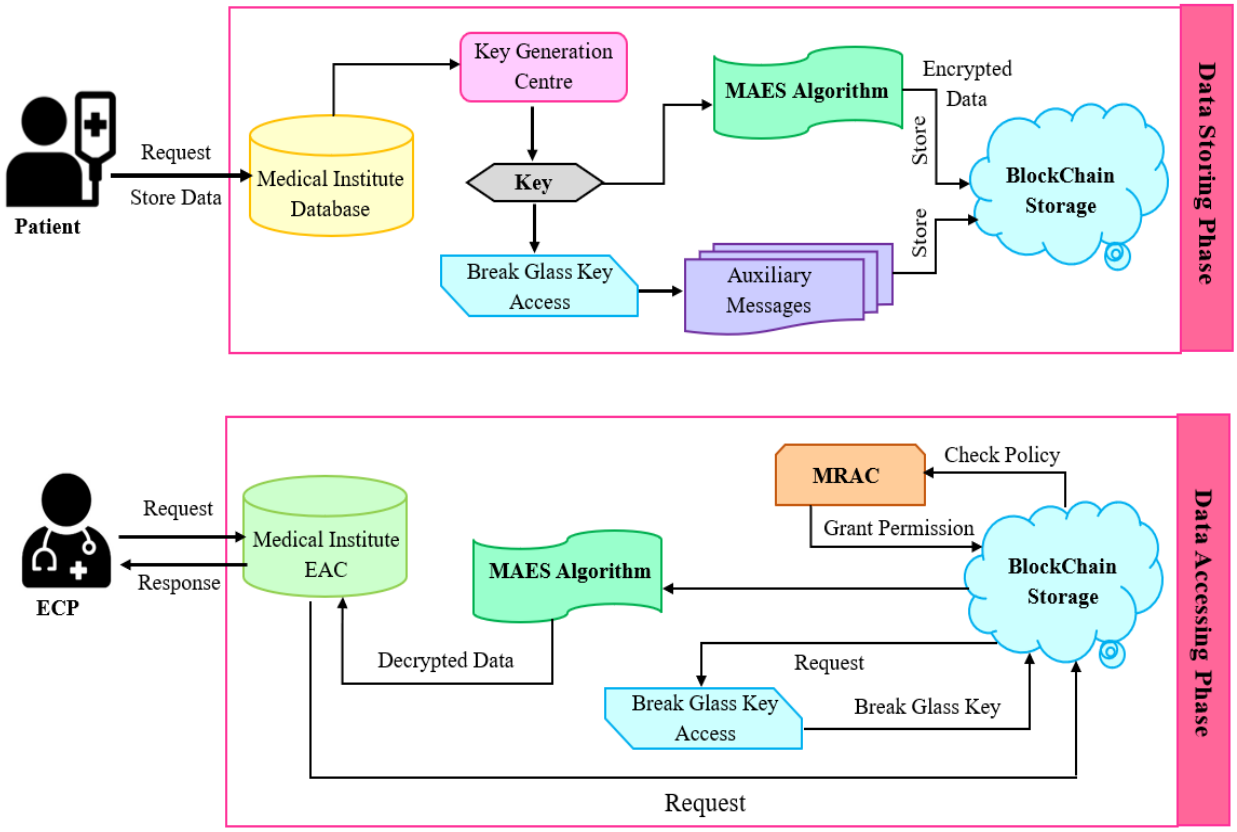


Figure 2. Overall Block Diagram for MAES-RAC Framework

The proposed MAES-RAC framework is based on the coordinated changes in the encryption mechanism as well as the access control model so that healthcare data can be safely handled. The traditional Advanced Encryption Standard (AES) is streamlined to give the changed AES (MAES) in the encryption layer. As shown in the data storing step, the specific key generation centre creates session-based keys, where the encryption rounds are simplified, vis-a-vis basic simplification of the key scheduling procedure and elimination of redundant transformations. The changes reduce the calculation and memory required and lower the processing time of AES. The encrypted medical records are then formatted directly in blockchain storage which enhances further the scalability, integrity and interoperability.

The Modified Rule-based Access Control (MRAC) mechanism is a logical expansion of the traditional Bell-LaPadula (BLP) confidentiality model implemented in the access control layer to ensure the implementation of more dynamic healthcare conditions. MRAC does policy checking on the real time and the stage of data accessing before allowing MAES to decrypt data. The read-up and write-down classical BLP properties are strictly observed, and there are more intelligence-based rules which allow authorization that depends on the situation. In addition, it has a controlled Break-Glass Key Access feature which is intended to allow emergency contact person (ECP) access under dire circumstances, but with all activities safely registered on blockchain making them auditable. These additions help the suggested framework to have enhanced confidentiality, speed cryptographic operations, and effective and realistic access controls that are policy-driven and focus on the current healthcare contexts.

4.1 Initialization phase

In this system, the patient plays the role of data owner. The patient set V is mathematically denoted as,

$$V = \{v_1, v_2, \ldots v_j, \ldots v_m\} \tag{1}$$

$$v_j = \{v\_id_j, v\_pw_j, v\_bio_j\} \quad (2)$$

where, v_m indicates the total number of patients, v_j represents the j^{th} patient in the set of patients V. The v_j consists of the patient's identity, password, and biometric and they are denoted as v_id_j, v_pw_j, and v_bio_j respectively. The ECP is the doctor or the nurse, who can access the medical file of the patients when the patients are in an emergency medical situation. The ECP set F is mathematically denoted as,

$$F = \{f_1, f_2, ...f_j, ...f_n\} \quad (3)$$

$$f_j = \{f\_id_j, f\_pw_j, f\_bio_j\} \quad (4)$$

where, f_n indicates the total number of ECPs, f_j represents the j^{th} ECP in the set of ECPs F. The f_j comprises of ECPs identity, password, and biometric and they are represented as f_id_j, f_pw_j, and f_bio_j respectively. The medical data of the j^{th} patient, which needs to be stored on the cloud server is mathematically represented as,

$$D_j = \{d_1^j, ...d_N^j\} \quad (5)$$

where D_j contains a different number of medical files of the j^{th} patient, N denotes the total number of medical files.

4.2 Registration Phase

The Registration Phase (RP) is used to register all the details of both patients and the ECPs in the medical institute database. The RP is responsible for creating the attributes, that aid for the right authorization. Initially, the patients register their details into the medical institute database by giving requests to the database with their credentials, such as identity, password, and biometrics. After the successful registration, the database sent valid acknowledgment to the patients. Likewise, the ECPs send requests to the medical institute database with their credentials, such as identity, password, and biometrics to make the registration. After registering the details, the database sent a valid acknowledgment to the ECPs as a reply. Figure 3 depicts the system of the registration phase.

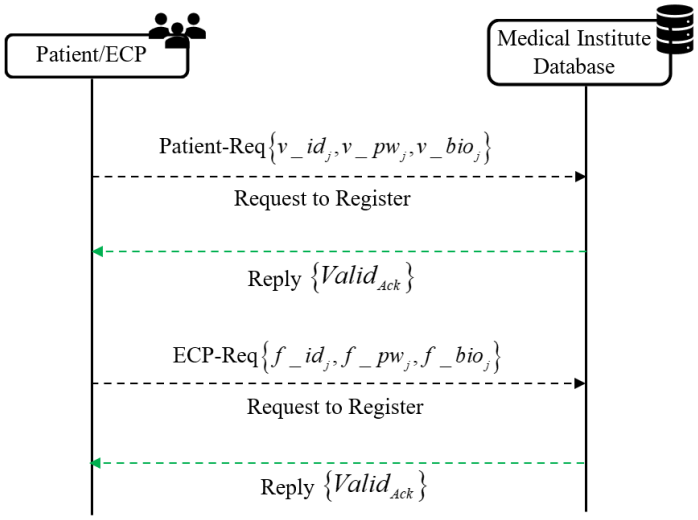


Figure 3. Process of Registration Phase

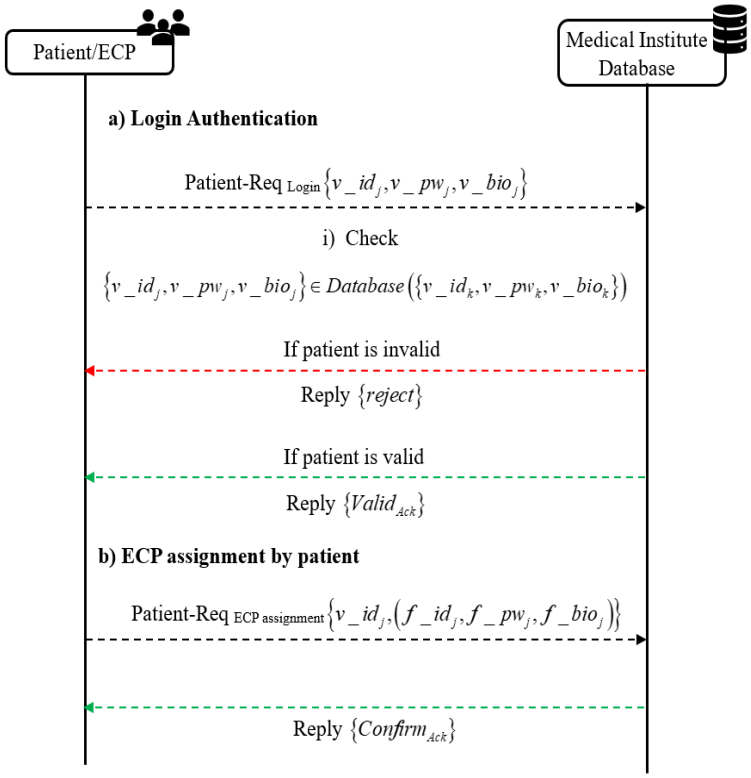


Figure 4. Login and Assignment of ECP Phase

4.3 Login and Assignment of ECP Phase

In this phase, the patients give login requests to the database with their credentials, such as passwords, identity, and biometrics. After receiving the login request, the database checks, that the credentials are already stored in the database. If it is already stored in the database, it grants access to the further process by sending a valid acknowledgment as a reply to the patient or otherwise rejecting the login request. After the successful login, the patient assigns their ECPs by sending a request with their identity and the ECP's identity, passwords, and biometrics to the database. Then the database stores the ECP details for that particular patient and sends a confirmed acknowledgement as the reply to the patient. In case of any emergencies, the assigned ECP can only access the patient's medical data. Figure 4 depicts the workflow of Login and Assignment of ECP Phase.

4.4 Data Storing Phase

After login and assignment of ECPs, the user gives a request to the database to store all the details. For secure storing, initially, the keys are produced from the key generation center. The patients give requests for generating to the key generation center with their identity and password. Then, the key generation center provides the key as a reply to the patient. Simultaneously, the BGKA framework [23] generates the break-glass key auxiliary message through the attribute-based break glass key generation mechanism, which is clearly explained in the following section

4.4.1 Attribute-based break glass key generation mechanism

Let G be the prime order's bilinear group and the generator of G is represented as q. The patient's identity v_id_j, and password v_pw_j are taken as input to generate break-glass key auxiliary messages. The patients randomly choose the variables ($\eta_1, \eta_2, \mu_1, \mu_2$), which belong to the whole number set Z_o^* and λ, λ_1 belong to the generator set G. Also, the break glass key bgk_j is the same as the key K, which is generated from the key generation center. Then, $K = bgk_i = \mu$ and additionally compute,

$$\lambda_2 = K\left(q^{\mu_1+\mu_2}\right)^{\eta_1} \cdot \left(\lambda_1\right)^{-1} \tag{6}$$

$$C_{v\_id_j} = \left(q^{\mu_1+\mu_2}\right)^{\eta_2} \cdot q_1^{H\left(v\_id_j\right)} \tag{7}$$

where $C_{v\_id_j}$ indicates the patient's cipher identity, $H\left(v\_id_j\right)$ represents the hash of the patient's identity. The break-glass key auxiliary messages are utilize to aid the ECPs retrieve the break-glass key to decrypt the patient's encrypted medical files and they are computed as,

$$bgk_{j,1} = \left(\mu_1, \lambda_1, q^{\eta_1}, q^{\eta_2}, C_{v\_id_j}\right) \tag{8}$$

$$bgk_{j,2} = \left(\mu_z, \lambda_2, q^{\eta_1}, q^{\eta_2}, C_{v\_id_j}\right) \tag{9}$$

Finally, the generated auxiliary messages $bgk_{j,1}$ $bgk_{j,2}$, patient's medical data, and privacy policy P are encrypted using the Modified Advanced Encryption Standard (MAES) algorithm. P is a policy describing who can access d_1^j and the privacy policy contains an access privacy level, that will allow or deny the ECPs to access the medical data d_1^j . The process of encryption through MAES is clearly explained in the following section

4.4.2 Modified Advanced Encryption Standard Algorithm for Encryption

The MAES algorithm is a symmetric-key block cipher designed to overcome computational overhead and improve throughput performance. In the traditional AES algorithm [22] [24], the mixed column step is the most computationally intensive operation, slowing down the overall encryption process. To address this, the MAES algorithm introduces a modified mix columns step to accelerate encryption and decryption with fewer computations and enhanced security. The number of encryption and decryption rounds is determined by the key size; a larger key size increases the algorithm's security level. One round of the MAES algorithm comprises four steps, and a total of 10 rounds are executed for complete encryption. These four steps are substitution bytes, shift rows, modified mix columns, and round keys. The architecture of the MAES algorithm is depicted in Figure 5.

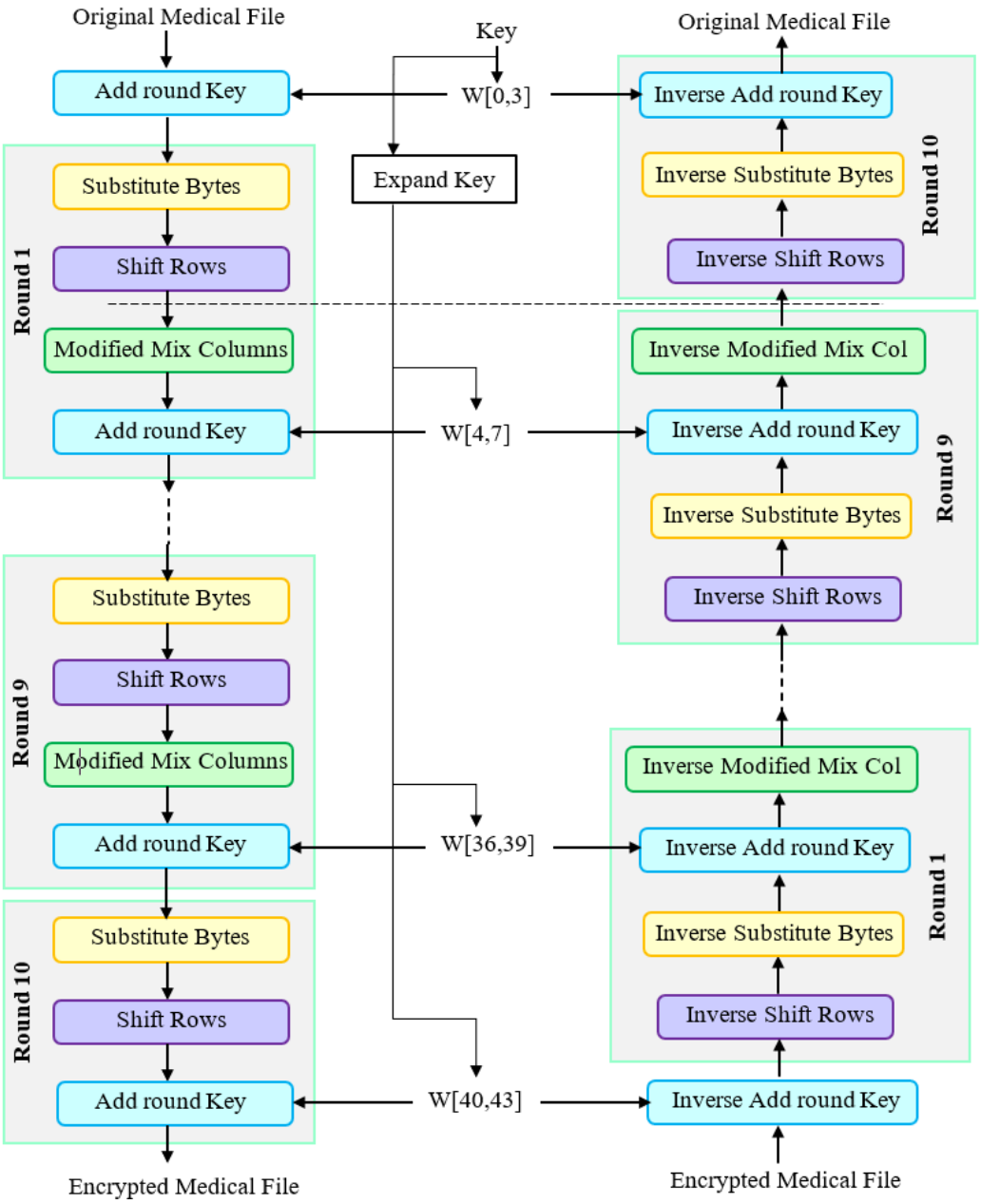


Figure 5. Structure of Modified Advanced Encryption Algorithm

a) Substitute Bytes

The substitute bytes are the first step of every round in the MAES algorithm and this step relies on the non-linear Substitution box (S-box), which has a unique mathematical property to change a byte into another byte. The security level of the MAES algorithm is improved through the step of byte substitution because it approves the need for nonlinearity. This step substitutes the input block's every byte with the columns and rows of the S-box. During the process of decryption, the S-box is utilized to reverse the result of the substitute bytes step. The working of the substitute byte step is shown in Figure 6.

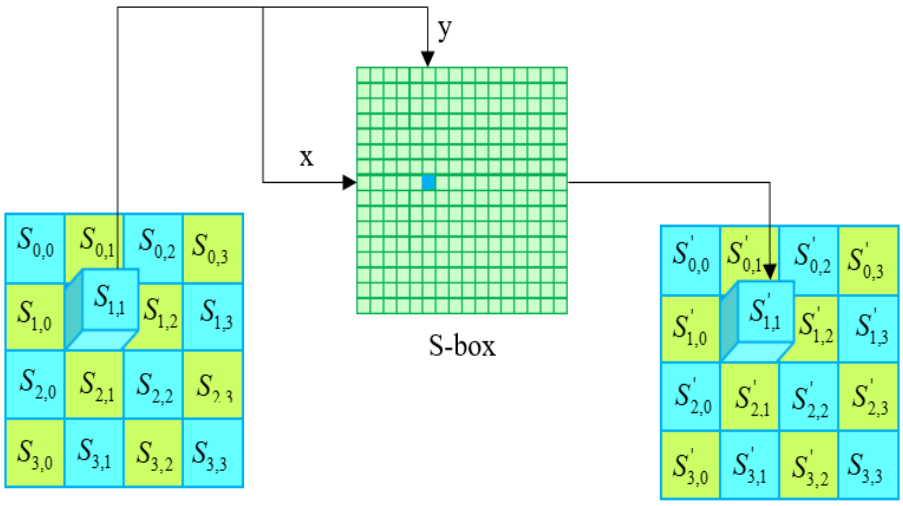


Figure 6. Working of Substitute Bytes Process

b) Shift Rows

After the substitute byte step, the next step executed in the round is shift rows. The core idea over this shift row step is to move the bytes circularly to the left of every row. The first row remains unaltered throughout this shift rows operation but the operation of circular shift is executed on the fourth, third, and the second rows respectively. Similarly, in the process of decryption, the first row remains unaltered, but the remaining rows are shifted to the right side as same as they shifted to the left side during the process of encryption. The working of the substitute byte step is shown in Figure 7.

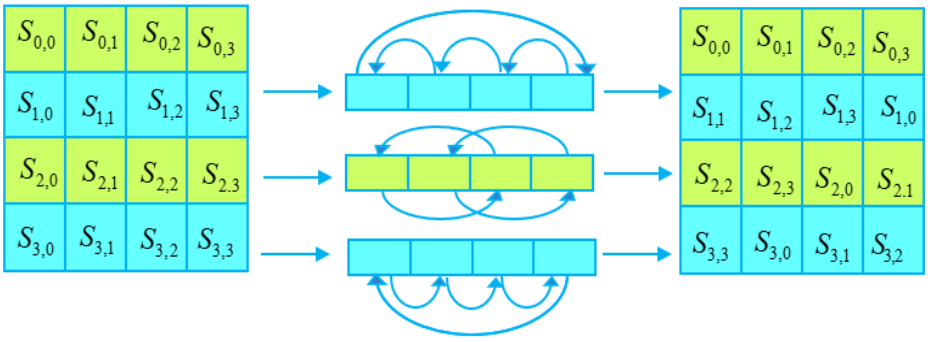


Figure 7. Working of Shift Rows Process

c) Modified Mixed Column

The modified mixed column operation is applied after the execution of the shift row operation. The modified mix column step is performed by rearranging the column of the fixed matrix using the value of rank. The single fixed matrix is represented below

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix}$$

The Rank value is obtained by performing the XOR operation on every row of the round key matrix and it is mathematically denoted as,

$$h = \begin{pmatrix} h_{0,0} & h_{0,1} & h_{0,2} & h_{0,3} \\ h_{1,0} & h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,0} & h_{2,1} & h_{2,2} & h_{2,3} \\ h_{3,0} & h_{3,1} & h_{3,2} & h_{3,3} \end{pmatrix} \tag{10}$$

The rank values to mix the columns are generated based on the following equation

$$\begin{cases} R_{V0} = h_{0,0} \oplus h_{0,1} \oplus h_{0,2} \oplus h_{0,3} \\ R_{V1} = h_{1,0} \oplus h_{1,1} \oplus h_{1,2} \oplus h_{1,3} \\ R_{V2} = h_{2,0} \oplus h_{2,1} \oplus h_{2,2} \oplus h_{2,3} \\ R_{V3} = h_{3,0} \oplus h_{3,1} \oplus h_{3,2} \oplus h_{3,3} \end{cases} \tag{11}$$

After computing all rank values, the rank value with the highest number is considered as the rank ‘1’ and the rank value with the lowest number is considered as the rank ‘4’. Then finally, based on the rank value, the single fixed matrix is rearranged as follows,

$$rearranged\ matrix = \begin{pmatrix} 03 & 01 & 01 & 02 \\ 01 & 02 & 03 & 01 \\ 02 & 03 & 01 & 01 \\ 01 & 01 & 02 & 03 \end{pmatrix} \tag{12}$$

Likewise, in the process of decryption, the computation of the rank number for the modified inverse mix column step remains the same, because the same key is used for both the process of the modified mix column and the modified inverse mix column.

d) AddRoundKey

This is the last step, which is done in each round of MAES. The AddRoundKey has the great capability to offer more security throughout the process of encryption. The AddRoundKey operation depends on generating the relationship between the cipher text and the key. In this operation, the XOR operation is done among every sub key bytes and state bytes. The process of encryption is over by executing all the steps in each round and the encryption using MAES is mathematically represented as,

$$c^j = MAES\_Enc\left(K, d_1^j, P\right) \tag{13}$$

where, is the encrypted data, which contains the key, patient’s medical details, and the privacy policy. After encryption, along with the encrypted data, break-glass key auxiliary messages and the patient's identity are stored in the storage of the blockchain securely. Figure 8 depicts the workflow of data data-storing phase.

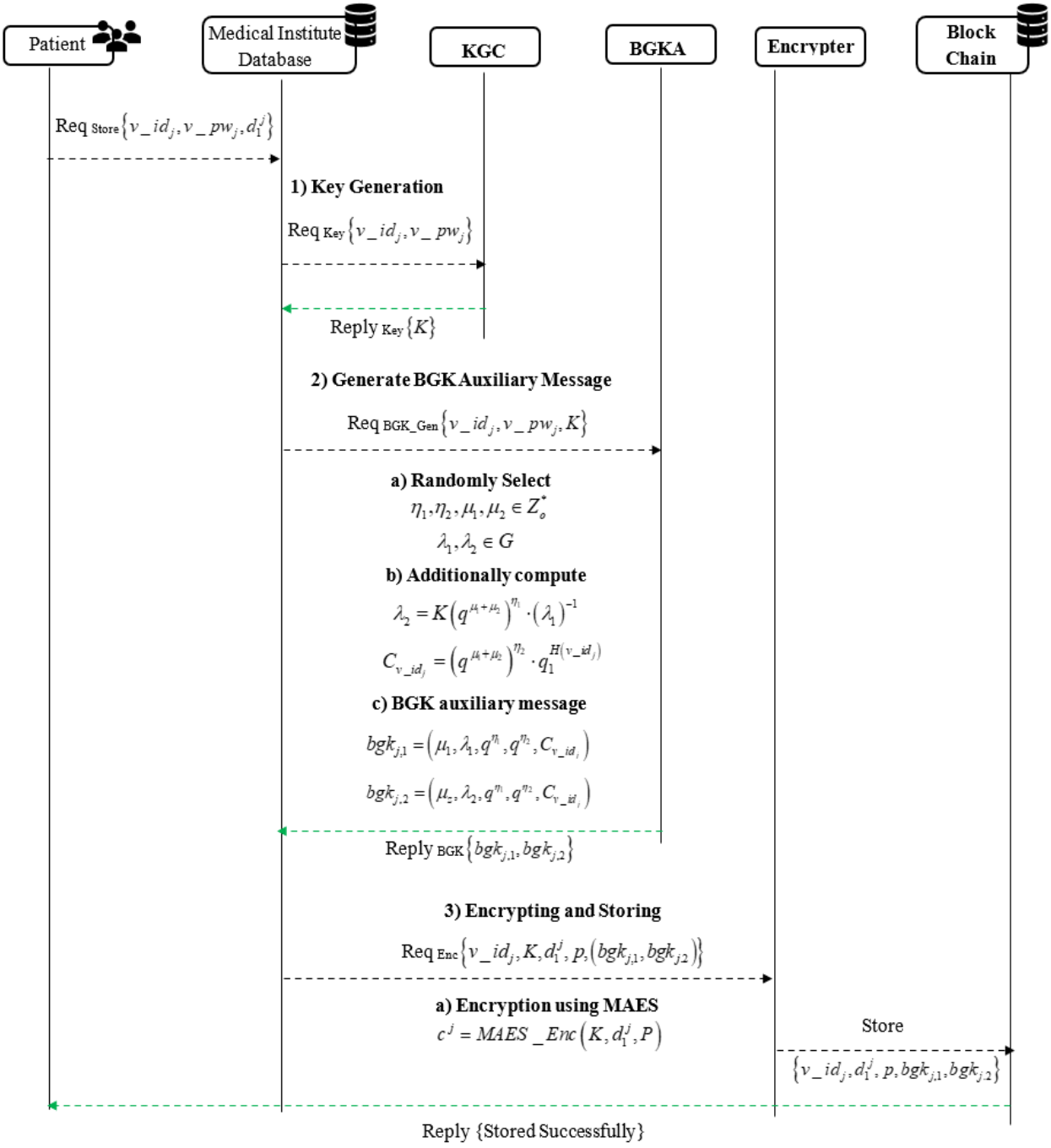


Figure 8. Data Storing Phase

4.4.3 MAES Algorithm security analysis.

The security of the proposed modified advanced encryption standard (MAES) algorithm is discussed so that the alteration that is made in the MixColumns transformation does not compromise the cryptographic power of the original aes structure. The MAES algorithm adheres to the basic design principles of AES such as substitution, permutation as well as key mixing and introduces a rank-based reorganization of the fixed matrix in its MixColumns step. This change will help minimize the computational complexity without compromising the security levels that are necessary to ensure that sensitive health information of patients is safely stored.

Strong confidentiality and diffusion of a very fast encryption algorithm is one of the main security characteristics. The SubBytes transformation used on the MAES algorithm is identical to the nonlinear substitution

box (S-box) that is used by the standard AES algorithm, and, as a result, it guarantees significant nonlinearity between a plaintext and a ciphertext. This nonlinearity is one of the causes of confusion as the relationship between the input data and the encryption key is extremely complicated. What is more, the operation of ShiftRows still spreads the position of bytes in the state matrix and the changedMixColumns operation offers diffusion by spreading minor changes in the input information around several bytes of the condition of the state matrix. The rank-based rearrangement of the MixColumns matrix leads to another variation of the transformation process to distribute the data dependencies throughout the entire block and reinforce the diffusion ability of the algorithm.

A different significant attribute of a secure encryption algorithm is the avalanche effect, in relation to which a slight variation of the input data or encryption key leads to a big and random variation in the output ciphertext. Under MAES algorithm, input changes are carried over to every round of the cipher through joint actions of SubBytes, ShiftRows, Modified MixColumns and AddRoundKey. Because of this difference, any slight alteration of the plaintext or the key propagates quickly through many bits in the encryption messages after several encryption round. This action serves as confirmation that the suggested change keeps the avalanche property in place that would ensure good cryptography security.

The vulnerabilities of the MAES algorithm to general cryptographic attacks are also taken into account. Because MAES has the same block size and key size hierarchy as AES, with 128-bit, 192-bit, and 256-bit keys also supported, it is by default so resistant to brute-force attacks: the key search space is so large. Moreover, the nonlinear S-box transformation process and multi-round transformation is employed in order to offer defense against differential and linear cryptanalysis, which seeks to leverage statistical knowledge between plaintext and ciphertext. The reordering of the MixColumns matrix dynamically additional upsets predictable pattern of transformations, and thus, attackers find it harder to extract a significant statistical correlation.

Computationally, the change in the MixColumns step decreases unnecessary operations, as well as increases processing efficiency but does not discard the needed transformation scheme of AES. These optimization results in quicker encryption and decryption processes which is very useful especially in the medical setting where bulk of sensitive data on patients requires to be safely handled and sent. As it can be seen in the performance analysis section, the experiment proved that the time taken by the MAES algorithm is less than the encryption and decryption times of multiple available methods and allows keeping the data secure.

4.5 Data Accessing Phase

Once an emergency arises, the ECP gives a request to the Emergency Access Control (EAC) to access the patient's medical data with their identity, password, biometric, patient identity, and privacy policy, that is $\left\{f\_id_j, f\_pw_j, f\_bio_j, v\_id_j, P\right\}$. Then the authentication check is done to validate whether the ECP is valid or not by checking whether the requested credentials belong to the data already stored in the database. If the authentication is not valid, reject the request with the reject reply, but if it is valid the access request is continued and further the request is passed into the Blockchain. Then the blockchain gives a request to the MRAC mechanism to check the privacy policy.

4.5.1 Modified Rule-based Access Control Mechanism

The MRAC mechanism is one of the effective algorithms to regulate who can access patient data. In this mechanism, the rule is defined by an attribute and the role. The attributes consist of the time of the request, the location of the requester (ECP), the role of the requester, and the IP address of the requester. Additionally, in MRAC mechanism, the Bell-LaPadula (BLP) model is also incorporated to check the privacy policy. Consider, that the request contains $\text{Req}_{\text{Data}} \left\{v\_id_j, ECP_{Detail}, Rule-attribute\right\}$, where the rule attribute consists of location, time, IP address, and the role of the requester that is a doctor or nurse. The ECP details consist of the identity, password, and biometrics of the requesters. The authorization check is the initial check of the authorization process and it is done with the details of ECPs or requesters and it is the process of checking whereas the requester has the suitable rights to access the patient's medical data, while policies are the attributes and the guidelines are utilized to determine access to the patient's medical data. The MRAC mechanism is used for the second check, which is based on the rule, that is attribute and role of the ECPs. The BLP model integrates the multilevel security through the policy of discretionary access control and also BLP model applies both the policy of discretionary access control and multi-level security to ensure the flexibility in access control. The access permission is granted by the BLP model, if and only if the policy should have the higher level of clearance. In BLP model, a system comprises of set of users,

which is represented as R, set of data is denoted as T and the set of security level is denoted as, α. The relation of domination among the set α is defined by $\leq_a$. The group of triples $\left(L_R, L_T, L_g\right)$ is denoted by L, where L_R and L_g indicates the mapping function of users to their highest clearance level and the present clearance level. The mapping function of data to its security level is represented as L_T. The group of access control is represented by $E = \{rd, wr, ap\}$, which includes writing, appending and reading respectively. The group of present access policy is denoted by $J = R \times T \times E$ and the group of discretionary access control is represented as $N \subseteq R \times T \times E$. The BLP model is defined as quadruple model $\left(g, a, l, p\right)$, where $g \subseteq J$ comprises the user's present access control, $a \subseteq N$ comprises the policy of present discretionary access control, security level of triple function includes $l = \left(l_R, l_T, l_g\right) \in L$ and the hierarchy of data is denoted as p. The BLP model checks the privacy policy, if they satisfy the properties of simple security policy (ss-policy), star policy and discretionary security policy (ds-policy), it grant permission to access the data.

- The property of ss-policy $\left(r,t,e\right) \in \left(R \times T \times E\right)$ is satisfied if the privacy policy holds one of the following rules;
  - $e = ap$
  - $e = rd\,(or)\ e = wr\ \ and\ \ l_T\left(T\right) \leq_a l_g\left(R\right)$
- The property of star policy (*-policy) $\forall\left(r,t,e\right) \in \left(R \times T \times E\right)$ is satisfied if the privacy policy holds one of the following rules:
  - $e = ap\ \ and\ \ l_g\left(r\right) \leq_a l_T\left(T\right)$
  - $e = wr\ \ and\ \ l_g\left(r\right) = l_T\left(T\right)$
  - $e = rd\ and\ \ l_T\left(T\right) \leq_a l_g\left(R\right)$
- The property of ds-policy is satisfied, if the privacy policy holds one of the following rules:

$$\left(r,t,e\right) \in g$$

$$\left(r,t,e\right) \in a$$

The doctor has full access to privacy, that is he can read, write, and edit the medical files. However, the nurse has a partial access policy, that is he can only read the medical files. Table 1 describes the working of MRAC.

Table 1. Working of MRAC mechanism

Modified Rule-based Access Control Mechanism

Modified_Rule_based_Access (Rule)
{
Check(Request.Location ∈ MedicalInstitute.Location ∩
Request.time ∈ Range(Role.time)
∩ Request.IP_address ∈ MedicalInstitute.IP_address ∩ CheckAccessPrivacy(Request.Role))
}
CheckAccessPrivacy(Request.Role)
{
If (Request.Role==Nurse)
If (Request.Accesspolicy ≤ Role. Access policy) ∧ (Valid_Doctor-Ack)
Validate (Bell-LaPadula_Model);
Else
Return Reject "No"
Else
Validate (Bell-LaPadula_Model);

Return Grant permission “Yes”
}

If all the rules are validated, permission is granted to access the patient’s medical files or otherwise access is denied. After granting permission to access, the blockchain gives a request with the identity of the patient and the BGK auxiliary messages to the BGKA mechanism for getting the key to decrypt the medical data. The key is extracted using the attribute-based break glass key extraction mechanism, which is clearly explained in the following section.

4.5.2 Attribute-based Break Glass Key Extraction Mechanism

If the patient faces an emergency medical condition, the attribute-based break glass key mechanism uses the patient's identity and breaks glass key auxiliary messages to extract the key for decrypting the medical files. Initially, choose the random number, s which belongs to the Z_o^* and compute α, then choose the random integer b_1, b_2 which belongs to the Z_o^* and compute Q_1, Q_2. These calculations are represented in the following equations.

$$\alpha = q^s \cdot q_1^{H(v\_id_i)} \tag{14}$$

$$Q_1 = \left(q^{\eta_2}\right)^{b_1} \tag{15}$$

$$Q_2 = \left(q^{\eta_2}\right)^{b_2} \tag{16}$$

Then compute the value of X_1, X_2 and B_1, B_2, where $B_1 = q^{b_1}$, and $B_2 = q^{b_2}$. These values are computed using the following equations.

$$X_1 = \lambda_1 \left(C_{v\_idi_j} \cdot \alpha^{-1}\right)^{b_1} \cdot \left(q^{\eta_1} \cdot Q_1 \cdot Q_2\right)^{-\mu_1} \tag{17}$$

$$X_2 = \lambda_2 \left(C_{v\_idi_j} \cdot \alpha^{-1}\right)^{b_2} \cdot \left(q^{\eta_2} \cdot Q_1 \cdot Q_2\right)^{-\mu_2} \tag{18}$$

Finally, the BGK is generated using the following equation

$$K = \left(X_1 \cdot X_2\right) \times \left(B_1 \cdot B_2\right)^s \tag{19}$$

After generating the BGK, the BGKA mechanism sends the key as a reply to the blockchain to decrypt the patient’s medical file. Then the decryption is done through the MAES algorithm to decrypt the medical file. The decryption process is the same as the encryption process, which is clearly explained in section 3.4.2 but the decryption process is an inverse of the encryption process. The decryption using the MAES algorithm is mathematically represented as

$$d_l^i = MAES\_Dec\left(K, c^j\right) \tag{20}$$

After decrypting, the patient’s medical data is forwarded to the ECPs, who need to access the patient’s medical data to review the condition of the patient. The utilization of various authentication mechanisms and effective encryption processes makes the patient’s medical data secure and it can’t able to accessed by an unauthorized person. Figure 9 depicts the workflow of data data-accessing phase.

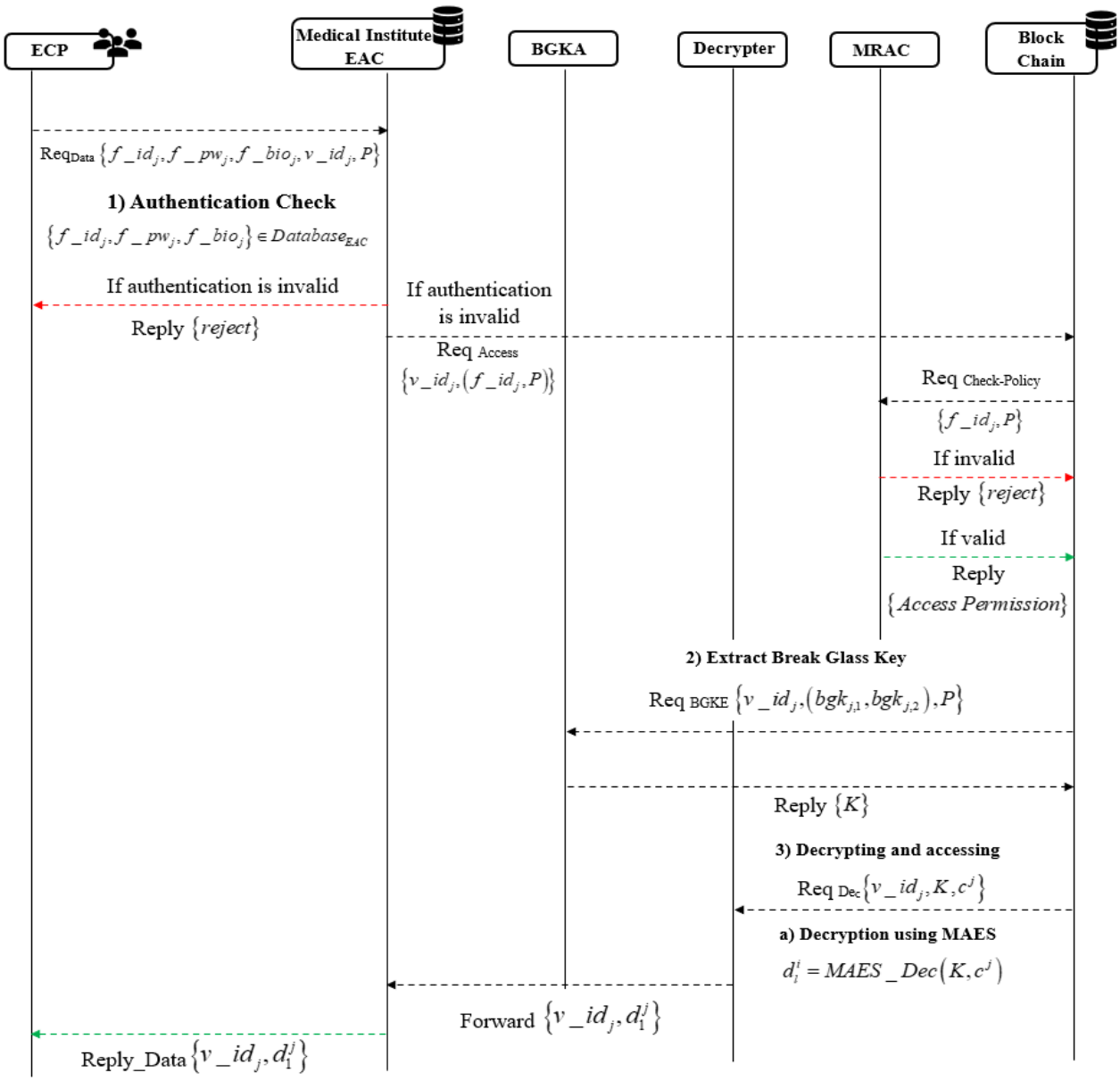


Figure 9. Data Storing Phase

5. Results and Discussion

This section describes the efficiency of the MAES-RAC framework, by comparing it with the existing approaches.

5.1 Experimental Setup

The MAES-RAC framework is executed on the latest version of PyCharm with Windows 11 OS and RAM of 16GB, ROM of more than 100GB, and a CPU with a speed of 1.7 GHz.

5.2 Dataset Description

The Healthcare dataset [25] and the Healthcare Management System dataset [26] are utilized in this research. The Healthcare dataset comprises 10,000 records and all records represent an artificial patient medical record. It consists of numerous parameters, such as basic information about the patient, admission details of the patient, and the patient's medical conditions. The Healthcare Management System dataset includes various tables, such as

patients table, doctors table, appointments table, medical procedure table, billing table, and demo table. Each table stores data about the patients, and healthcare providers, including their contact details and name.

5.3 Evaluation Metrics

The evaluation metrics, such as decryption time, genuine user rate, encryption time, memory usage, and responsiveness matrices are utilized to estimate the efficiency of the proposed MAES-RAC framework

Encryption Time: The encryption time measure is utilized to estimate the total period taken through the proposed MAES-RAC framework to convert the original data to the encoded data, which is mathematically represented as,

$$Enc_T = \frac{Enc_{Data}}{T} \tag{21}$$

where Enc_T represents the encryption time, Enc_{Data} indicates the encrypted data, and T denotes the time.

Decryption Time: The decryption measure is utilized to estimate the total period taken by the proposed MAES-RAC framework to convert the encoded data into the original data, which is mathematically represented as,

$$Dec_T = \frac{Dec_{Data}}{T} \tag{22}$$

where Dec_{Data} indicates the encrypted data, Dec_T represents the encryption time, and T denotes the time.

Genuine User Rate: It is the proportion of the total amount of valid users to the total number of users and it is mathematically represented as,

$$gur = \frac{t_{Valid}}{t_{users}} \tag{23}$$

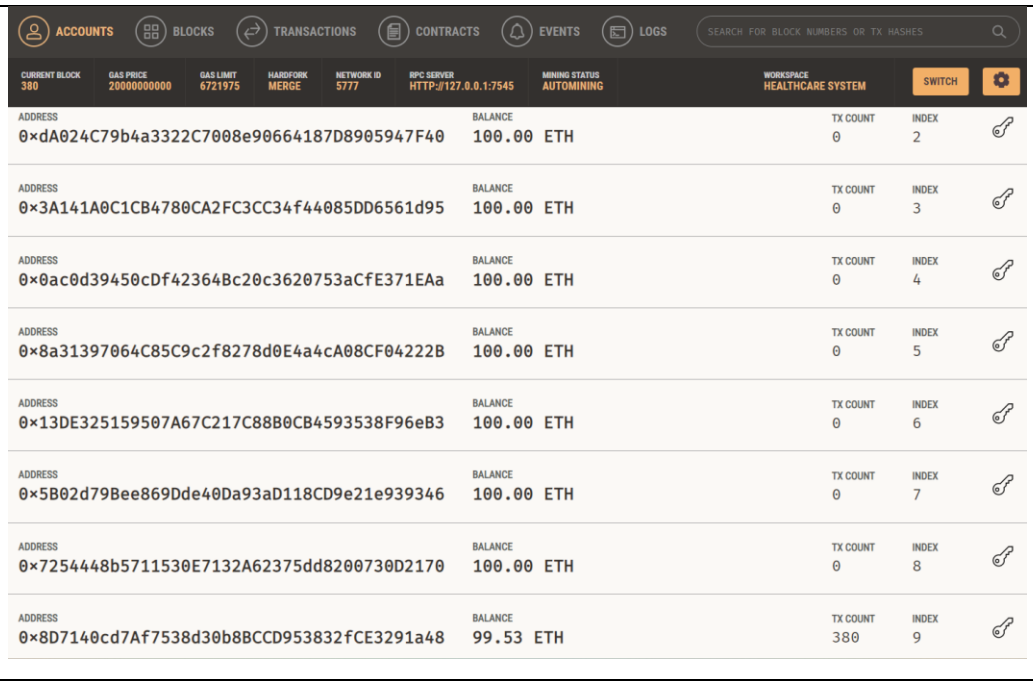
where, t_{users} represents the total number of users, and t_{Valid} is the total amount of valid users.

Memory Usage: It is the measure used to estimate the total volume of memory utilized by the system for implementing both the process of decryption and the encryption respectively.

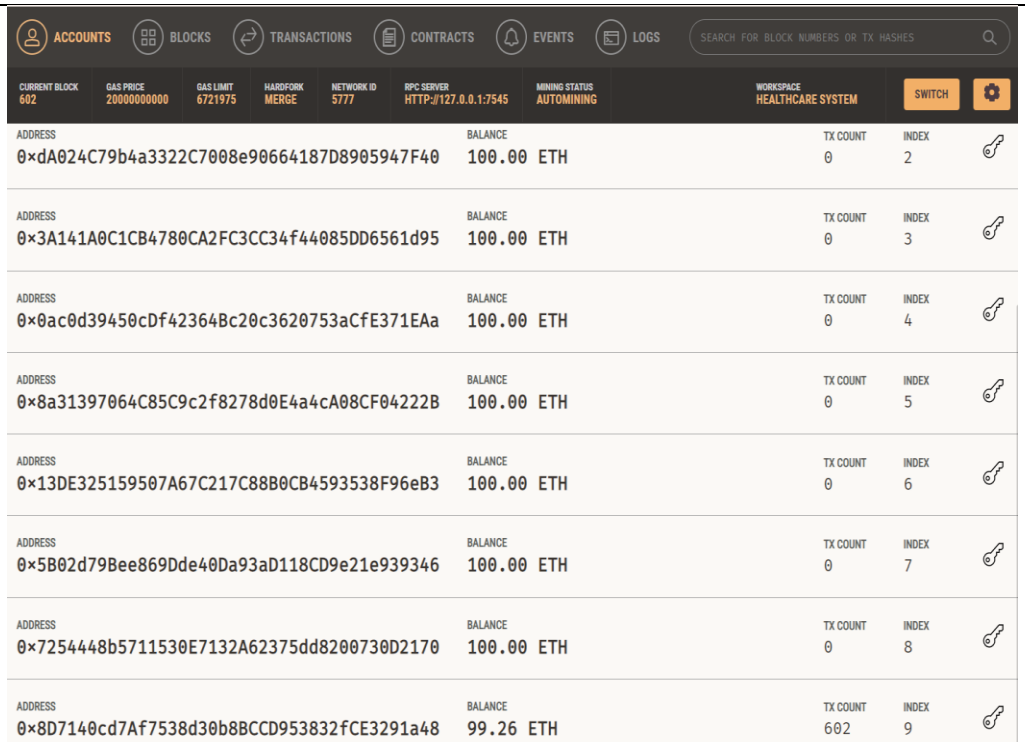
Responsiveness: A responsiveness measure is used to estimate the speed of the framework to process the request.

5.4 Simulation model

Figure 10 displays the simulation model of the MAES-RAC framework.



(a)



(b)

Figure 10. Simulation Model

5.5 Performance Analysis

Figure 11 shows the performance of the MAES-RAC framework using both the Healthcare and Healthcare Management System datasets in terms of genuine user rates with varying key sizes, such as 128, 192, and 256. The

genuine user rate is evaluated for varying numbers of users as 50, 100, 150, 200, and 250 with different key sizes. For the Healthcare dataset, the MAES-RAC framework achieved a rate of 0.9352 for the key size 128, 0.9358 for the key size 192, and 0.948 for the key size 256 of the 250 users. For the Healthcare Management System dataset, the MAES-RAC framework achieved the rate of 0.913 for the key size 128, 0.921 for the key size 192, and 0.931 for the key size 256 of the 250 users. It shows that the genuine user rate upsurges with the increasing volume of users as well as key size respectively.

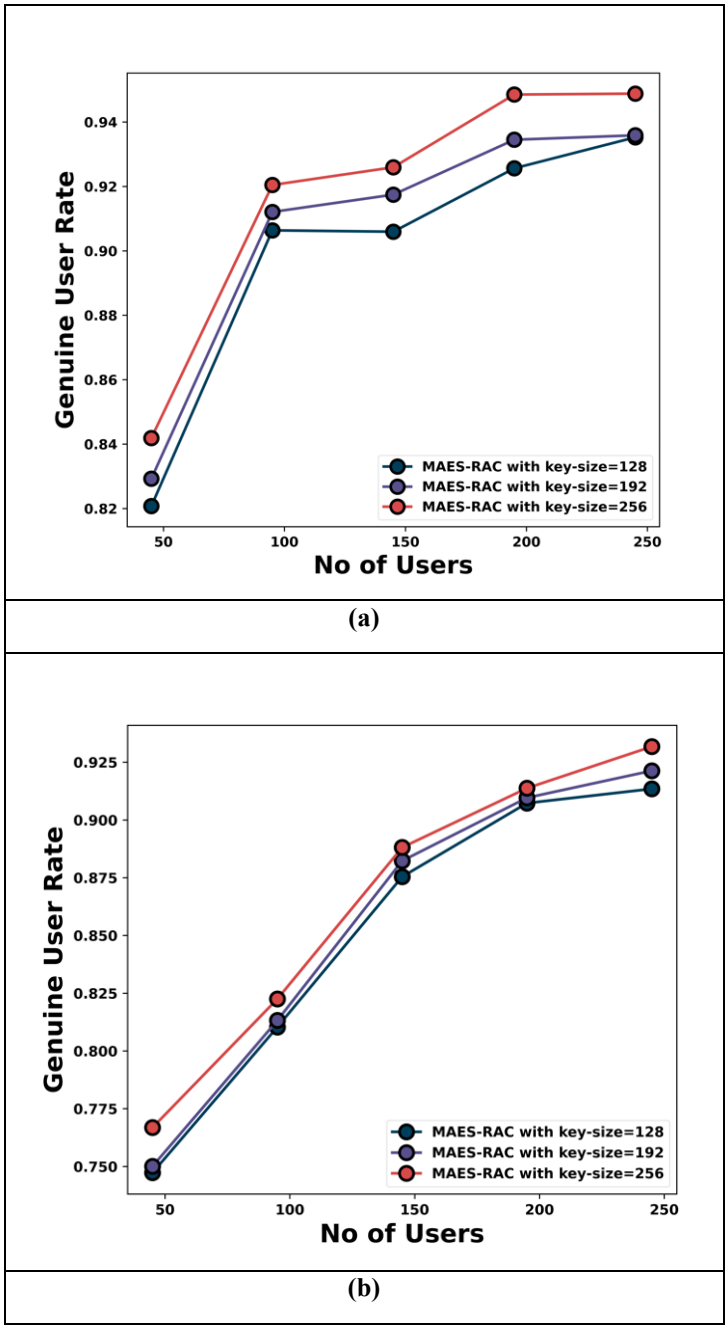


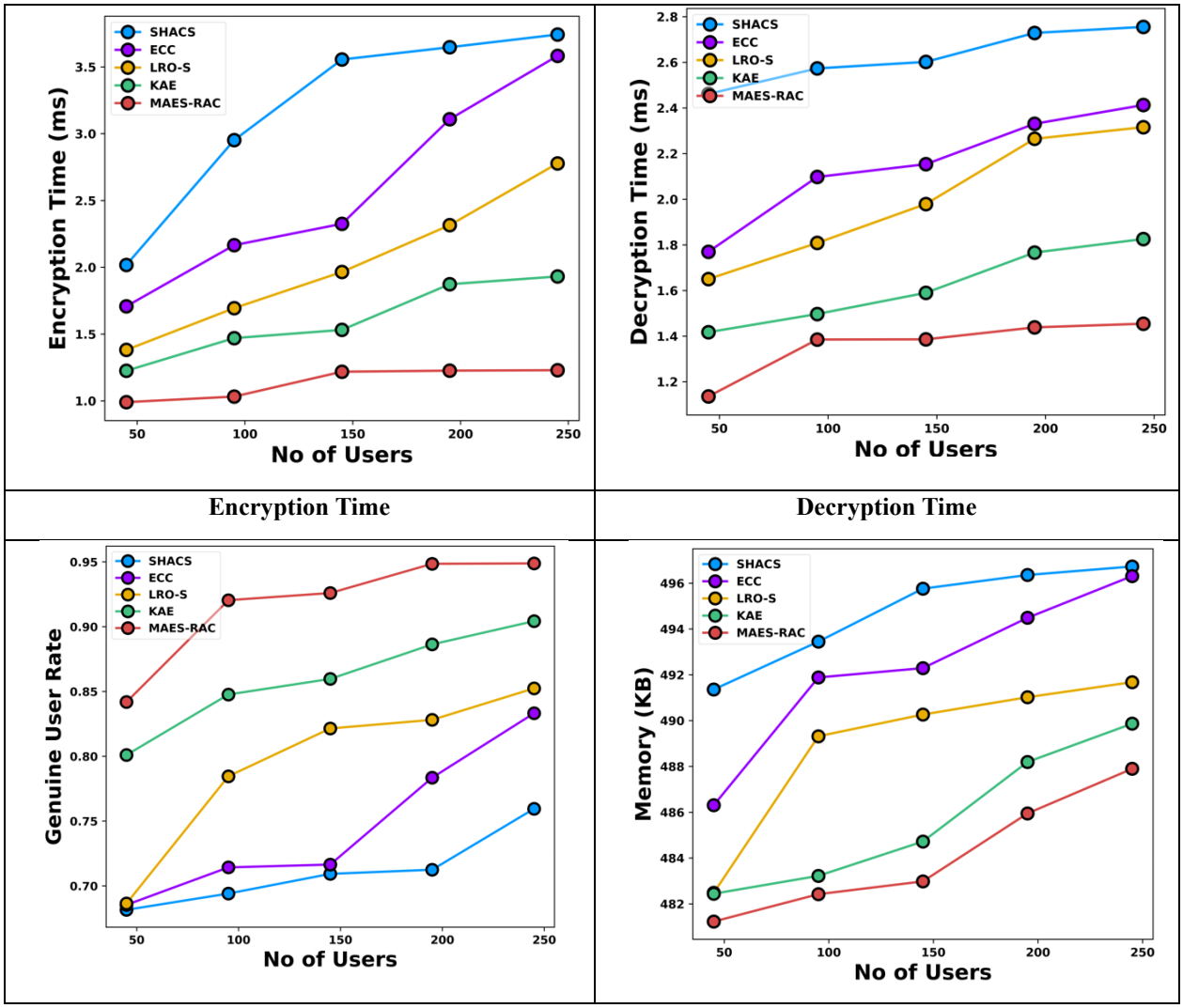
Figure 11. (a) Performance in terms of Genuine User Rate using Healthcare dataset (b) Performance in terms of Genuine User Rate Healthcare Management System dataset

5.6 Comparative Methods

The existing approaches, such as SHACS [3], ECC [6], LRO-S [7], and KAE [4] are compared with the MAES-RAC framework to evaluate the efficacy of the proposed MAES-RAC framework.

5.6.1 Comparative Analysis using Healthcare dataset

Figure 12 shows the comparison of the proposed MAES-RAC and the existing SHACS, ECC, LRO-S, and KAE approaches using the Healthcare dataset. The time duration taken by the existing SHACS, ECC, LRO-S, and KAE approaches to encrypt the medical files of 250 users are 3.74ms, 3.58ms, 2.77ms, and 1.93ms. However, the MAES-RAC framework takes 1.22ms to encrypt the medical files, comparatively which is lesser than the other approaches. Similarly, the time taken to decrypt the medical file using the proposed MAES-RAC for 250 users is 1.45ms, but time taken for the existing SHACS is 2.75ms, ECC is 2.41ms, LRO-S is 2.31ms and KAE is 1.82ms. This comparison shows the MAES-RAC framework takes less time to decrypt the medical files than the existing approaches. With 250 users, existing approaches achieved a genuine user rate of 0.75 for SHACS, 0.833 for ECC, 0.852 for LRO-S, and 0.90 for KAE. However, the proposed MAES-RAC framework attained a rate of 0.948, which is greater to the other compared current methods. The proposed MAES-RAC framework takes the responsiveness of 4.39ms for 250 users and the existing SHACS, ECC, LRO-S, and KAE approaches take 3.24ms, 3.97ms, 4.12ms, and 4.27ms respectively, which shows the responsiveness of the MAES-RAC framework is higher than the existing approaches. The memory required or utilized for 250 users through the MAES-RAC framework is 487.8KB, although the memory utilized by the existing SHACS, ECC, LRO-S, and KAE are 496.7KB, 496.3KB, 491.6KB, and 489.8KB, which clearly shows the memory utilized by the MAES-RAC framework lower than the existing methods. These improvements over the proposed MAES-RAC framework are due to the various authentication mechanisms and the modified encryption algorithm to validate the authorized user and encrypt the medical files effectively and securely.



Encryption Time | **Decryption Time**

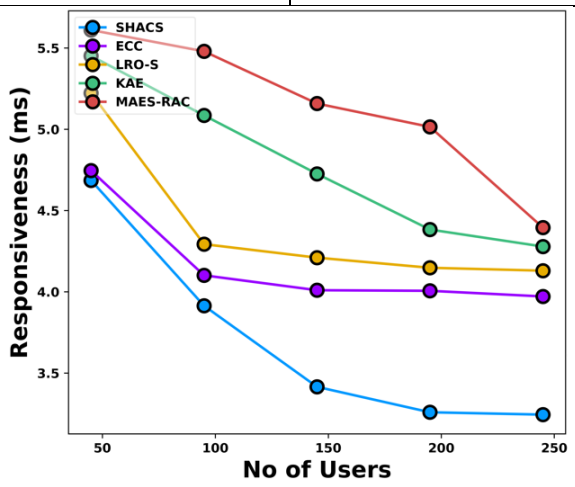
Genuine User Rate	Memory Usage
	
Responsiveness	

Figure 12. Comparative Analysis using Healthcare dataset

5.6.2 Comparative Analysis using Healthcare Management System dataset

Figure 13 illustrates a comparison of the proposed MAES-RAC mechanism with existing SHACS, ECC, LRO-S, and KAE approaches, utilizing the Healthcare Management System dataset. The encryption time for a medical file with 250 users using the proposed MAES-RAC is 1.44ms, while the existing SHACS, ECC, LRO-S, and KAE require 3.83ms, 3.72ms, 3.38ms, and 2.83ms, respectively. This evaluation demonstrates that the MAES-RAC framework encrypts medical files more rapidly than the existing approaches. Similarly, the decryption times for 250 users' medical files using the existing SHACS, ECC, LRO-S, and KAE approaches are 2.52ms, 2.25ms, 2.19ms, and 1.91ms, respectively. In contrast, the MAES-RAC framework achieves a decryption time of 1.23ms, which is comparatively faster than the other approaches. The responsiveness of the existing SHACS, ECC, LRO-S, and KAE approaches are 3.02ms, 3.42ms, 3.78ms, and 4.05ms, respectively, while the proposed MAES-RAC framework exhibits a responsiveness of 4.25ms for 250 users, indicating superior responsiveness compared to the existing approaches. The memory utilization for 250 users using the MAES-RAC framework is 394.6KB, whereas the existing SHACS, ECC, LRO-S, and KAE utilize 407.6KB, 405.7KB, 402.4KB, and 396.8KB, respectively. This clearly indicates that the MAES-RAC framework requires less memory than the existing frameworks. For 250 users, the existing approaches achieved genuine user rates of 0.75 for SHACS, 0.833 for ECC, 0.852 for LRO-S, and 0.90 for KAE. However, the proposed MAES-RAC framework attained a rate of 0.948, which is superior to the other compared existing frameworks. These results demonstrate that the efficacy of the MAES-RAC framework is higher than the existing approaches, achieved through the integration of various authentication mechanisms and the modified encryption algorithm for secure storage and access of patient medical files.

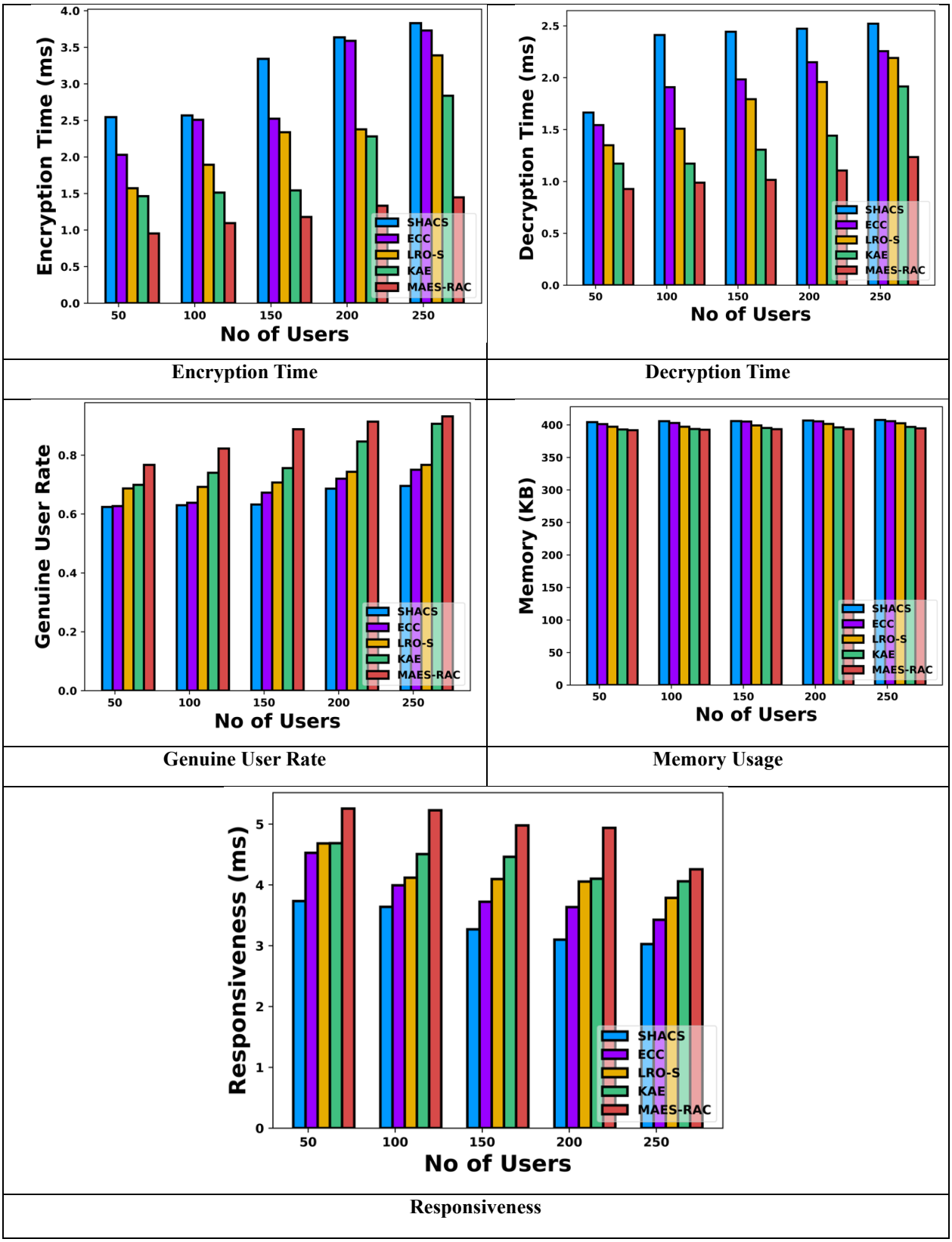


Figure 13. Comparative Analysis using Healthcare Management System dataset

5.7 Comparative Discussion

This section describes the comparison discussion over the proposed MAES-RAC framework and the existing SHACS, ECC, LRO-S, and KAE approaches. The compared existing approaches revealed the significant advantages in storing and accessing medical files, but still due to the presence of certain limitations in the existing approaches they can't perform well in securing the patient's medical files. The suggested SHACS approach can't be executed over the huge distributed system of healthcare due to its low computing resources. Due to the utilization of the homomorphic encryption algorithm in the suggested KAE approach, the computational load occurred on every unit. In the suggested ECC approach, medical data leakage happens, due to the usage of a manual system to encrypt and store the medical files. The security level is decreased in the suggested LRO-S, due to the absence of integration of blockchain technology in the LRO-S approach. The proposed MAES-RAC framework overcomes the limitations of the compared existing approaches by integrating the MAES algorithm to encrypt the medical files securely and the computational load of the existing approach is conquered by the MAES algorithm. Additionally, the mixing of blockchain in the MAES-RAC framework enhances the overall security of the MAES-RAC approach in storing the patient's medical data. Moreover, the MAES-RAC approach can execute on the huge distributed system of healthcare, due to the existence of the MRAC mechanism and high computing resources in the MAES-RAC approach. These advantages make the MAES-RAC framework more secure in terms of storing and accessing the patient's medical files. The outcomes achieved by the MAES-RAC framework and existing approaches using both datasets are shown in Table 2

Table 2. Comparative Discussion of MAES-RAC Framework

Methods/Metrics		**SHACS**	**ECC**	**LRO-S**	**KAE**	**MAES-RAC**
Healthcare Dataset	**Encryption Time (ms)**	3.742	3.582	2.777	1.930	**1.229**
	Decryption Time (ms)	2.755	2.412	2.315	1.825	**1.454**
	Genuine User Rate	0.759	0.833	0.852	0.904	**0.948**
	Memory (KB)	496.7	496.3	491.6	489.8	**487.8**
	Responsiveness (ms)	3.244	3.971	4.129	4.278	**4.394**
Healthcare Management System Database	**Encryption Time (ms)**	3.830	3.729	3.389	2.836	**1.446**
	Decryption Time (ms)	2.520	2.255	2.190	1.915	**1.235**
	Genuine User Rate	0.695	0.750	0.766	0.906	**0.931**
	Memory (KB)	407.6	405.7	402.4	396.8	**394.6**
	Responsiveness (ms)	3.026	3.425	3.786	4.058	**4.255**

6. Conclusion

In this study, the MAES-RAC framework is proposed to enable secure storage and access to medical data, thereby enhancing patient privacy, protecting against unauthorized access, and improving the overall integrity of healthcare data. The utilization of the MAES algorithm minimizes computation and accelerates both encryption and decryption processes. Moreover, the implementation of the MRAC mechanism, incorporating the BLP model and EAC mechanism, verifies policies and authentication protocols, restricting unauthorized access to patients' medical files. Additionally, the employment of blockchain technology to store medical data enhances the scalability and interoperability of the MAES-RAC framework. Overall, the MAES-RAC framework effectively secures sensitive healthcare data and ensures that only authorized personnel or ECPs can access patients' medical files. The MAES-RAC framework was evaluated using metrics such as decryption time, genuine user rate, encryption time, responsiveness, and memory usage, achieving values of 1.23ms, 0.931, 1.44ms, 4.255ms, and 394.6KB, respectively, when using a healthcare management system dataset. For the healthcare dataset, the framework obtained values of 1.45ms, 1.22ms, 0.948, 4.39ms, and 487.8KB, respectively. Future research directions will

include an optimization-driven framework for a secure key generation mechanism to facilitate faster and more secure data storage and access.

Reference

1. Ali, A., Ali, H., Saeed, A., Ahmed Khan, A., Tin, T.T., Assam, M., Ghadi, Y.Y. and Mohamed, H.G., 2023. Blockchain-powered healthcare systems: enhancing scalability and security with hybrid deep learning. *Sensors*, *23*(18), p.7740.
2. Sonkamble, R.G., Bongale, A.M., Phansalkar, S., Sharma, A. and Rajput, S., 2023. Secure data transmission of electronic health records using blockchain technology. *Electronics*, *12*(4), p.1015.
3. Sangeetha, S.B., Selvarathi, C., Mathivanan, S.K., Cho, J. and Easwaramoorthy, S.V., 2024. Secure Healthcare Access Control System (SHACS) for Anomaly Detection and Enhanced Security in Cloud-Based Healthcare Applications. *IEEE Access*, vol. 12, pp. 164543-164559, 2024,
4. Oh, J., Son, S., Kwon, D., Kim, M., Park, Y. and Park, Y., 2024. Design of secure and privacy-preserving data sharing scheme based on key aggregation and private set intersection in medical information system. *Mathematics*, *12*(11), p.1717.
5. Kala, M.K. and Priya, M., Smart IoT-Blockchain Security to secure Sensitive Personal Medical Data using Shuffled Random Starvation Link Encryption, *IEEE Access*, vol. 12, pp. 168182-168196, 2024.
6. Ryu, J. and Kim, T., 2025. Enhancing Hospital Data Security: A Blockchain-Based Protocol for Secure Information Sharing and Recovery. *Electronics*, *14*(3), p.580.
7. Almalawi, A., Khan, A.I., Alsolami, F., Abushark, Y.B. and Alfakeeh, A.S., 2023. Managing security of healthcare data for a modern healthcare system. *Sensors*, *23*(7), p.3612.
8. Reegu, F.A., Abas, H., Gulzar, Y., Xin, Q., Alwan, A.A., Jabbari, A., Sonkamble, R.G. and Dziyauddin, R.A., 2023. Blockchain-based framework for interoperable electronic health records for an improved healthcare system. *Sustainability*, *15*(8), p.6337.
9. Da Costa, L., Pinheiro, B., Cordeiro, W., Araújo, R. and Abelém, A., 2023. Sec-Health: A blockchain-based protocol for securing health records. *IEEE Access*, *11*, pp.16605-16620.
10. Kumar, R. and Tripathi, R., 2021. Scalable and secure access control policy for healthcare system using blockchain and enhanced Bell–LaPadula model. *Journal of Ambient Intelligence and Humanized Computing*, *12*, pp.2321-2338.
11. Pournaghi, S.M., Bayat, M. and Farjami, Y., 2020. MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption. *Journal of Ambient Intelligence and Humanized Computing*, *11*(11), pp.4613-4641.
12. Shen, B., Guo, J. and Yang, Y., 2019. MedChain: Efficient healthcare data sharing via blockchain. *Applied sciences*, *9*(6), p.1207.
13. Raghav, N. and Bhola, A., 2021. Blockchain based privacy preservation in healthcare: a recent trends and challenges. *Psychol. Educ. J*, *58*, pp.5315-5324.
14. Siedlecka-Lamch, O., 2023. Secure Medical Data Storage with Blockchain Technology. *Procedia Computer Science*, *225*, pp.961-968.
15. Kruse, C.S., Stein, A., Thomas, H. and Kaur, H., 2018. The use of electronic health records to support population health: a systematic review of the literature. *Journal of medical systems*, *42*(11), p.214.
16. Kasula, B.Y., 2023. Framework development for artificial intelligence integration in healthcare: optimizing patient care and operational efficiency. Transactions on Latest Trends in IoT, 6(6), pp.77-83.
17. Ryu, J., Kang, D., Lee, H., Kim, H. and Won, D., 2020. A secure and lightweight three-factor-based authentication scheme for smart healthcare systems. *Sensors*, *20*(24), p.7136.
18. An, Q., Rahman, S., Zhou, J. and Kang, J.J., 2023. A comprehensive review on machine learning in healthcare industry: classification, restrictions, opportunities and challenges. *Sensors*, *23*(9), p.4178.
19. Anitha, C., Komala, C.R., Vivekanand, C.V., Lalitha, S.D. and Boopathi, S., 2023, February. Artificial Intelligence driven security model for Internet of Medical Things (IoMT). In *2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)* (pp. 1-7). IEEE.
20. Gupta, D.S., Mazumdar, N., Nag, A. and Singh, J.P., 2023. Secure data authentication and access control protocol for industrial healthcare system. *Journal of Ambient Intelligence and Humanized Computing*, *14*(5), pp.4853-4864.
21. Gordon, W.J. and Catalini, C., 2018. Blockchain technology for healthcare: facilitating the transition to patient-driven interoperability. *Computational and structural biotechnology journal*, *16*, pp.224-230.
22. Abdullah, A.M., 2017. Advanced encryption standard (AES) algorithm to encrypt and decrypt data. *Cryptography and Network Security*, *16*(1), p.11.

23. De Oliveira, M.T., Michalas, A., Groot, A.E., Marquering, H.A. and Olabarriaga, S.D., 2019, October. Red Alert: break-glass protocol to access encrypted medical records in the cloud. In *2019 IEEE International Conference on E-health Networking, Application & Services (HealthCom)* (pp. 1-7). IEEE.
24. Ramareddy, S. K. (2025). Blockchain-Based Security Solutions: A Review. International Journal of Recent Advances in Engineering and Technology, 14(2s), 256–260. https://doi.org/10.65521/intjournalrecadvengtech.v14i2s.1466
25. Dumbre, S. P., & Dere, K. D. (2025). An Attribute-Based Access Control Mechanism for Cloud Storage using Blockchain Technology. International Journal of Recent Advances in Engineering and Technology, 14(1s), 79–84. https://doi.org/10.65521/intjournalrecadvengtech.v14i1s.249
26. Healthcare dataset, https://www.kaggle.com/datasets/prasad22/healthcare-dataset, Accessed on March 2025.
27. Healthcare Management System Dataset, https://www.kaggle.com/datasets/anouskaabhisikta/healthcare-management-system?select=Patient.csv, accessed on March 2025