IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 30, No. 8, August, 2025, pp. 2053-2065
Journal homepage: http://iieta.org/journals/isi

Efficient Privacy-Preserving mHealth Framework Using Crisscross AES and FCFS-NDPPP in Hybrid Cloud

Mariapragasam Arokia Muthu*, Balasubramaniyam Prakash

Department of Computing Technologies, SRM Institute of Science and Technology, Chennai 603203, India

Corresponding Author Email: am1669@srmist.edu.in

https://doi.org/10.18280/isi.300811

Received: 13 June 2025
Revised: 13 August 2025
Accepted: 24 August 2025
Available online: 31 August 2025

Keywords:
hybrid cloud security, healthcare data privacy, mobile health monitoring systems, privacy-preserving framework, real-time data protection

ABSTRACT

Ensuring privacy and regulatory compliance in healthcare data exchange is a persistent challenge, particularly with the rise of hybrid cloud infrastructures and mobile health systems. This research introduces a novel cryptographic framework that combines dynamic key shuffling in Crisscross AES with a First-Come-First-Serve (FCFS)-based Network Data Privacy Preserving Protocol (NDPPP) to strengthen both data confidentiality and multi-user scheduling. Unlike conventional encryption schemes, Crisscross AES enhances the traditional AES model by integrating matrix transposition and adaptive key reordering, providing stronger resistance against cryptanalytic attacks. Meanwhile, the FCFS-based NDPPP ensures fair, collision-free task scheduling with reduced latency, addressing the complexity of secure multi-user access in heterogeneous healthcare environments. Together, these mechanisms form a lightweight, scalable, and regulation-compliant (HIPAA/GDPR) solution for end-to-end protection of sensitive medical data. Experimental validation on real-world health datasets confirms that the proposed system significantly improves encryption robustness, scheduling efficiency, and data integrity, offering a practical foundation for secure and reliable healthcare service deployment in hybrid cloud ecosystems.

1. INTRODUCTION

Cloud computing is a system in which one server or a group of servers performs computations for other computers located elsewhere and connected via the Internet. The cloud is a platform where anyone can access various technologies remotely without installing them on their local machines. If users don't utilize the technology, they are not charged for it. The cloud can be seen as a vast network of computers, hosting diverse services and making them accessible to users worldwide through the web. With suitable hardware and software, these facilities host a variety of applications across different data centers [1]. A cloud provider offers IT services on a pay-per-use basis users pay only if they use the services. Examples include processing capabilities such as Amazon EC2 and platforms for building, testing, and hosting web applications like Google AppEngine. To enhance the accuracy and efficiency of healthcare information, many individuals now use Electronic Healthcare (EHC) applications [2].

By applying proper encryption techniques during communications between physicians and patients, unauthorized access and masquerade attacks can be prevented. For cloud-based health data retrieval, the cloud must ensure anonymous search capabilities. Protocols for secure multi-keyword search in the cloud designed to return the top-N results to end users were proposed. To protect user identity, authentication was performed over the Internet using telecommunication techniques [3]. Smooth lattice systems that prevent replay and impersonation attacks help define an interface with strong resistance to such threats. Advantage of this research is the reduced computational cost along with enhanced data security. Keywords from vertically partitioned databases are effectively organized along with horizontally partitioned data during pre-processing [4]. Each homepage may include distinct elements such as medication, cosmetics, and more. Customers categorize purchasing relationships between medications and cosmetic products using a common payment card. Proposed a protocol utilizing hierarchical identity-based encryption originally developed to support multi-keyword searches for different physicians [5].

Existing methods support only single keyword-based searches for individual physicians. This approach reduces computational and storage costs. Many researchers agree that Mobile Cloud Computing (MCC) is a promising strategy for next-generation ubiquitous healthcare systems. Based on the severity of a patient's condition, MCC can generate healthcare data analysis results, which may be stored in the patient's medical record for future reference or used to trigger alerts to physicians [6]. MCC provides an ideal environment for sharing, transmitting, and processing sensitive health images and Electronic Health Records (EHRs). MCC-based medical services have been widely adopted for physiological data processing and remote patient monitoring [7]. Mobile cloud healthcare has seen extensive use in multi-agent medical consultations. As a result, various MCC architectures have been proposed to meet diverse healthcare needs. One of the

major early concerns in mobile computing was the short battery life of smartphones and tablets. This issue can be mitigated by offloading computationally intensive tasks to the cloud, reducing the energy burden on mobile devices. Unreliable network connectivity not only causes unexpected disruptions but also forces the communication modules of mobile devices to consume more power than necessary [8].

Numerous researchers have extensively analysed the energy costs associated with mobile cloud offloading, particularly in scenarios where network conditions deteriorate significantly. For example, studies have thoroughly examined how different network environments impact application migration to the cloud concluding that migration decisions must carefully consider network-related factors [9]. Researchers have explored the development of intelligent task allocation policies to optimize multifactorial mobile cloud offloading. The introduction of smart home medical care, supported by the IoT marks a revolutionary phase in patient treatment by enabling remote monitoring and management of medical data [10]. As IoT devices and systems for remote patient tracking become more widespread, robust security measures are essential to protect sensitive health information. Wearable devices with integrated sensors allow for continuous health monitoring but also introduce security vulnerabilities, such as device hacking and message interception. The advancement of such technologies, combined with the growing power of quantum computing, poses additional challenges in safeguarding private medical data [11].

Existing cryptographic protocols like RSA and ECC currently protect the majority of digital communications, face serious threats from the rapid evolution of quantum computing especially due to quantum algorithms capable of efficiently solving integer factorization and discrete logarithm problems. As stronger quantum computers emerge, these widely-used encryption methods could be broken, potentially compromising the confidentiality of critical data across various domains, including healthcare [12]. To address these risks, modern cryptographic schemes such as ECC, RSA, and AES are being revisited. In response to the looming threat of quantum attacks, researchers have proposed several Post-Quantum Cryptography (PQC) techniques that are resilient to both classical and quantum computational threats. It is essential to urgently adopt PQC algorithms that are resistant to quantum attacks [13].

In smart ecosystems, lightweight encryption has become a critical solution for ensuring data accessibility, confidentiality, and integrity during peer-to-peer communications. The average smart home is expected to house over 50 internet-connected devices, underscoring the urgent need for secure and efficient hybrid encryption strategies. These strategies may include combinations such as RSA/AES, ECC/AES, and future PQC-based approaches within various symmetric encryption contexts [14]. These hybrid cryptosystems represent a promising balance between security and operational performance. There remains a significant gap in comprehensive research focused on identifying optimal encryption configurations and methods tailored specifically for medical environments [15].

EHR systems have advanced considerably in recent years. Originally designed to store administrative data such as billing, EHRs now include comprehensive patient information lab results, diagnoses, clinical notes, medications, and more. These records are increasingly leveraged in predictive analytics to support personalized medicine and enhance treatment strategies. Medical data analysis relied on statistical and existing Machine Learning (ML) methods [16]. These methods lack the ability to capture complex, long-range dependencies and structured features as effectively as modern Deep Learning (DL) models. DL has revolutionized EHR analysis by handling time-series data more efficiently and with reduced reliance on manual feature engineering and pre-processing [17].

2. RELATED WORKS

According to the Cloud Security Alliance (CSA) in its security guidance report, cloud services are categorized into three distinct models based on the services they offer: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). IaaS, such as Amazon EC2, provides clients with core computing infrastructure such as networking capabilities, unstructured data storage, and processing units necessary for running any application [18]. It also supports all stages required for the development, testing, and deployment of internet-based applications. Examine multiple public cloud platforms and introduce CloudCmp systematic framework designed to evaluate and compare the costs and performance of various cloud providers. It focuses on key service attributes, including elastic computation, persistent storage, and network performance by analysing metrics that directly impact customer application efficiency [19].

To maintain fairness, accuracy, and consistency, CloudCmp operates within specific cost constraints while conducting these assessments. Upon applying the tool to several prominent cloud service providers, the researchers discovered significant differences in cost-effectiveness and service performance across vendors. As a result, CloudCmp serves as a valuable platform to help users identify the most appropriate cloud provider tailored to their specific application needs [20]. Key factors to consider when designing a secure network include non-repudiation, integrity, reliability, confidentiality, and availability. As outlined by the Open Systems Interconnection (OSI) model, encryption techniques are typically implemented at the application layer during data transmission to ensure secure communication [21].

Based on specific requirements, individuals can choose from various existing information security techniques. Security is an essential component that extends from the physical layer to higher levels of the information transmission system. To ensure comprehensive network information security, all layers above the physical (material) layer also contribute significantly. Depending on the developer's security policies, authentication and identification processes may be implemented at one or more layers. At the physical layer, tasks such as fault detection, attack identification, and the deployment of intelligent countermeasures are crucial for maintaining network security [22].

Another critical concern is data availability. Any interruption in access to data can disrupt business operations or service delivery, potentially resulting in customer attrition, revenue loss, and reputational damage. This study also highlights the security implications of data inaccessibility, emphasizing the increased risks posed by mobile or distributed data particularly when information is exchanged between countries with conflicting regulatory standards. The research further explores network security concerns within virtualized

environments, focusing on the Xen hypervisor is an open-source virtualization platform [23]. From a security standpoint, one of the key architectural challenges in cloud computing is securing communication among multiple VMs on the same host. To mitigate these risks, the authors propose a novel virtualized network architecture designed to better isolate and protect communication between VMs, thereby enhancing security in cloud-based infrastructures [24].

With the rise of mobile devices, patients now commonly access cloud-based services via smartphones support various authentication methods, including: Two-Factor Authentication (2FA); Three-Factor Authentication (3FA); Multi-Factor Authentication (MFA). These methods significantly strengthen personal data protection. Even with built-in multi-factor authentication, smartphones remain vulnerable, particularly when integrated with cloud platforms that may have security flaws. Implementing three-factor authentication may also hinder usability [25]. For ordinary patients, accessing their mobile devices for routine use becomes inconvenient, as authorization delays reduce user experience. Although smartphones and tablets have largely replaced traditional PCs and are often considered secure, still exposed to threats when used in security-sensitive environments such as healthcare, defense, and telecommuting. To address these vulnerabilities, multiple identification protocols have been designed to strengthen security, ensuring complete safety, client identity verification, and secure certification mechanisms [26].

As part of efforts to secure patient data stored in the cloud, a secure 2FA protocol was proposed. Due to persistent security vulnerabilities, researchers developed a secure message exchange mechanism based on watermarking. This technique enhances message integrity by embedding unique identifiers within the message, tolerating multimedia communication errors. Despite its merits, the watermarking-based method has a critical drawback: the absence of a timestamp during message transmission. This omission exposes the system to replay and processing-based attacks, undermining the security and authenticity of communications [27]. A fingerprint authentication technique was developed to enhance patient data security in the cloud. Fingerprint identification serves as a third authentication factor to improve privacy. The fingerprint recognition process involves converting RGB images to grayscale, adjusting and minimizing blur effects, and segmenting the image used for biometric-based verification. This processed fingerprint is embedded as a watermark in the patient's image. The study's outcomes were validated using Galaxy S3 and BlackBerry Z smartphones, demonstrating that the proposed system performs effectively and is easy to use [28]. The method does not ensure secure image transmission. It remains vulnerable to session-based attacks. To address these issues, researchers developed a cloud-based authentication system using multiple biometric factors. For smartcard-based authorization, both palm vein patterns and fingerprint credentials are accepted. The smartcard stores the user's biometric impressions and is matched against the patient's records [29].

A system was also developed for distributing ranked reports, leveraging an alternative encryption algorithm that allows hashtags to be securely shared with multiple users. To evaluate patient data, the Attribute-Based Encryption Method (ABEM) is applied. This method offers benefits such as reduced computation and a smaller index structure. It suffers from storage complexity. A key advantage of ABEM is that it only requires one key to decrypt patient records [30]. With the attribute set stored in an index tree, users can efficiently compute the decryption key. One strength of this method is that the encrypted medical document maintains a constant size. Its drawback lies in the length of the secret key. To enhance patient data security, proposed a method aimed at protecting the confidentiality of patient information. This approach utilizes an expansion operation to validate individual claims. Although this method requires minimal implementation effort, it consumes significantly more time compared to previous studies [31]. In this scheme, all patient data is processed for validation, but due to the extensive computation time, its efficiency is reduced. To address some of these issues, introduced a detection mechanism based on radio repetition rates. A major drawback of existing methods is the reliance on long decryption keys and lengthy storage periods [32]. To mitigate these issues, designed a cipher text protocol encryption with a fixed key size, minimizing storage complexity and decryption delays. The key size was standardized to 672 bits making it suitable for resource-constrained devices used for secure key storage and decryption. The main advantage of this method is the efficient storage of patient information, while its drawback is the long data transmission duration.

3. PROBLEM FORMATION

There is a crucial trade-off between security of information, integrity confidence, and scheduling efficiency in hybrid cloud infrastructures that handle mobile medical information. The main goal is to reduce security flaws while maintaining low processing overhead and equitable access for several users in real-time.

Let: $D = \{d_1, d_2, \dots, d_n\}$ denote the set of healthcare data transmitted. $U = \{u_1, u_2, \dots, u_m\}$ represent the set of users accessing data. T_x be the timestamp when user u_x requests access. $E(D_x)$ be the encryption function using Crisscross AES. $Q = \{u_x, T_x\}$ be the request queue scheduled using FCFS. $X(D)$ be the integrity verification function. C_{enc} and C_{sched} represent encryption and scheduling costs, respectively. P be the NDPPP-based privacy function.

Objectives

Maximize Encryption Strength:

$$\max(S_{enc}) \max\left(H\left(E(D_x)\right)\right) \tag{1}$$

where, $H(.)$ is the entropy function measuring the randomness of encrypted data.

Minimize Latency in FCFS Scheduling:

$$\min\left(L_{avg}\right) = \min\left(\frac{1}{m}\sum_{x=1}^{m}\left(T_{serv,x} - T_x\right)\right) \tag{2}$$

where, $T_{serv,x}$ the time when u_x request is served.

Ensure Data Integrity:

$$\forall d_x \in D, X(d_x) = True \tag{3}$$

i.e., each transmitted data chunk must pass integrity checks (e.g., hash matching).

Minimize Overall Computation Cost:

$$\min(C_{total}) = \min(C_{enc} + C_{sched}) \tag{4}$$

Preserve Multi-User Privacy:

$$P(D,U) \geq \delta \tag{5}$$

where, δ is the threshold for acceptable privacy level in the NDPPP protocol.

The challenge is to develop an integrated cryptographic and scheduling framework that enhances encryption robustness using Crisscross AES, guarantees data integrity, and ensures efficient multi-user request handling via FCFS-based NDPPP, all while maintaining low computation overhead and compliance with healthcare data regulations in hybrid cloud systems.

4. MATERIALS AND METHODS

This study proposes a secure and efficient framework for managing mobile medical data in hybrid cloud environments by integrating planning, cloud computing, and encryption techniques shown in Figure 1. It uses a practical medical dataset containing sensitive patient information and deploys the system using Amazon Web Services (AWS) and Edge-based simulators. The proposed method enhances traditional AES through a modified Crisscross AES, which includes key iteration and row-column transpositions for improved security. The Network Data Privacy Preserving Protocol (NDPPP) employs a First-Come-First-Served (FCFS) scheduler and supports secure multi-user access using role-based data division and authentication tokens. SHA-256 ensures data integrity, while benchmarking tools measure latency and efficiency. The framework is evaluated for encryption strength, scheduler performance, and compliance with security standards in both standalone and hybrid setups. MATLAB, Python (PyCryptodome), and CloudSim are used to test the system's scalability, resilience, and real-time applicability.

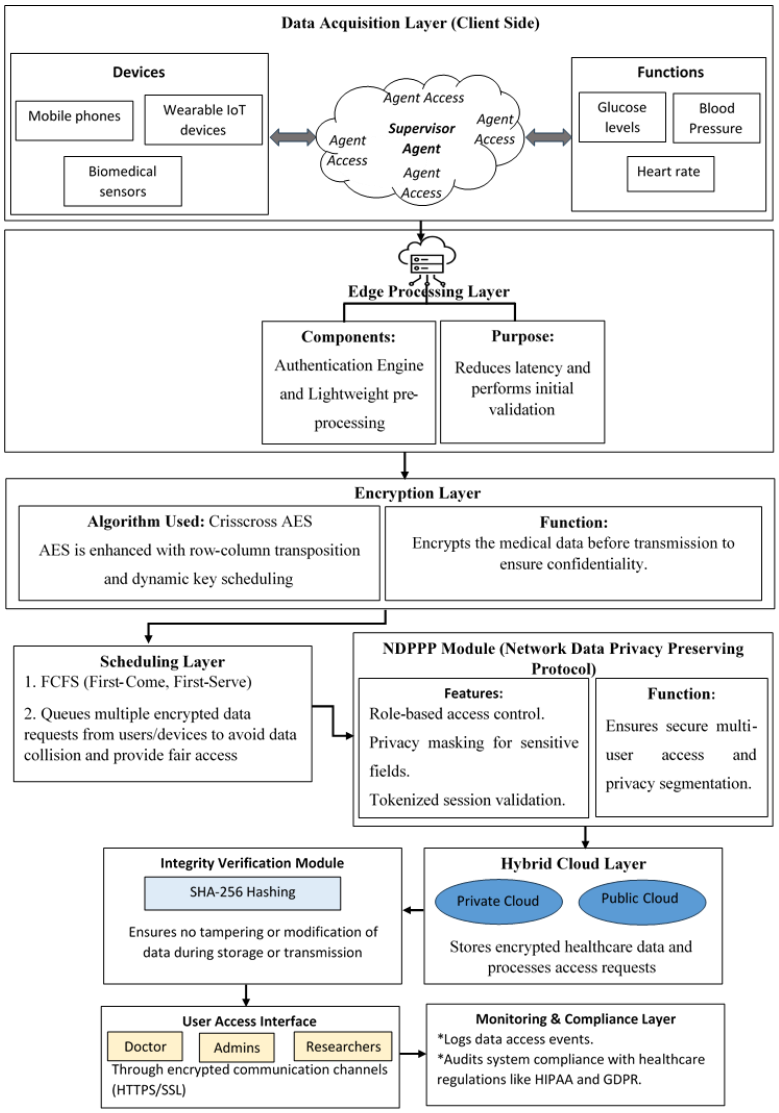

Figure 1. Proposed system

4.1 Data acquisition from wearable and mobile devices

Wearable sensors (such as glucose and cardiac sensors) and applications for mobile health (mHealth applictions) continually track and record an individual's critical indicators are used in contemporary mobile medical facilities to collect information shown in Table 1. In addition to timestamp, setting, and gadget ID metadata, each entry of information contains physical parameters including heart rate (HR), blood pressure (BP), oxygen saturation (SpO_2), and glucose level (GL). The core of the health monitoring system is this uncooked, current information, which is produced often. The

information is recorded in an organized manner as follows:

$$D_x = \{HR, BP, SpO_2, GL, t, loc, ID\} \quad (6)$$

where, D_x represents the collected data from the x^{th} instance.

Table 1. Dataset description

Chapter 1 Attribute	Chapter 2 Description
Chapter 3 Dataset Name	Chapter 4 Mobile Healthcare Monitoring Dataset (MHMD)
Chapter 5 Source	Chapter 6 Simulated IoT healthcare data / Real-world dataset from PhysioNet / MIMIC-III
Chapter 7 Data Type	Chapter 8 Time-series, textual, categorical
Chapter 9 Size	Chapter 10 ~50,000 records / 150MB
Chapter 11 Data Fields	Chapter 12 Patient ID, Timestamp, Heart Rate, Blood Pressure, Oxygen Level, Glucose Level, Activity Status
Chapter 13 Sensitive Fields	Chapter 14 Patient ID, Location, Diagnosis, Contact Info
Chapter 15 Data Frequency	Chapter 16 1 record per minute per patient
Chapter 17 Collection Devices	Chapter 18 Wearable Sensors, Mobile Devices, Health Monitoring Apps
Chapter 19 Preprocessing	Chapter 20 Noise filtering, normalization, token-based anonymization

4.2 Edge pre-processing for normalization and noise filtering

Preparing is an essential stage in the proposed architecture for integrity-assured and confidential mobile medical information to guarantee consistency of information, eliminate noise, and get it ready for scheduled and encryption. The edge processing unit may be installed on gateways or handheld devices, performs actual time pre-processing to guarantee data is accurate and uniform before transfer to the cloud. Each numeric property is first scaled for equitable encryption and scheduling processes using Z-score standardization. Each value X is changed using:

$$Z = \frac{I - \mu}{\sigma} \quad (7)$$

where, μ is the mean and σ is the standard deviation of the feature. This helps standardize data to a mean of 0 and standard deviation of 1. To eliminate transient sensor noise, a moving average filter is used, calculated as:

$$I_t' = \frac{1}{n} \sum_{x=t-n+1}^{t} I_x \quad (8)$$

This smoothens the signal, enhancing reliability before encryption and scheduling.

4.3 Anonymization and secure tokenization at the edge

The solution incorporates token-based anonymity at the edge directly to safeguard the identities of patients because healthcare information is important. Before being stored or sent, a distinct hash-based token is used in place of each patient's ID to protect the confidentiality of data. A cryptographic hash function, such SHA-256, is used by the tokenization procedure and is defined as:

$$Token_{ID} = Hash(PatientID||Timestamp) \quad (9)$$

This guarantees that the information cannot be connected to any particular person, even in the event that it is stolen. The solution complies with downstream Crisscross AES encryption requirements and NDPPP confidentiality laws by managing these pre-processing stages at the edge, minimizing cloud computing costs, lowering delay, and guaranteeing private.

4.4 Encryption layer - crisscross AES (ECAES)

The enhanced AES is applied in a grid fashion in the proposed crisscross AES method. AES, the proposed crisscross technique, is a ten-round, four-level combo. Three M-1s are included in the first level, three M-1s are included in the second round, and so on until the third level. All three levels are combined in the fourth round, which is then sent to the servers. In cloud computing, when the user accesses both IaaS and SaaS offerings, Figure 2 illustrates how ECAES is implemented. When it comes to user-related concerns, schedule, and safety are two of the cloud system's primary disadvantages. To solve this issue, a novel idea of combining FCFS of dominance components for scheduling purposes and Crisscross AES uses a crisscross conversion to reconstruct the normal AES state matrices. The change swaps components both row-wise and column-wise according to a hidden permutation variable π or a predetermined crisscross design. Unpredictability and resilience to cryptanalysis are increased as a result.

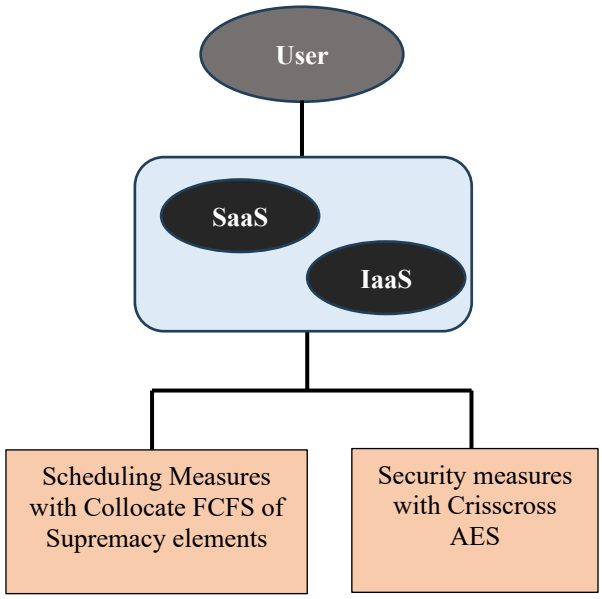

Figure 2. ECAES method

Let the AES state matrix be:

$$S = \begin{bmatrix} s_{00} & s_{01} & s_{02} & s_{03} \\ s_{10} & s_{11} & s_{12} & s_{13} \\ s_{20} & s_{21} & s_{22} & s_{23} \\ s_{30} & s_{31} & s_{32} & s_{33} \end{bmatrix} \quad (10)$$

The Crisscross Transformation (CT) is applied as:

$$S' = CT(S) = \pi_{row}\left(\pi_{col}(S)\right) \quad (11)$$

where, π_{row} : Row permutation function; π_{col} : Column permutation function. This reshuffles the AES state matrix before the usual AES rounds, enhancing diffusion.

Final ECAES: Combining standard AES operations with the crisscross transformation, the final Crisscross AES encryption can be expressed as:

$$C = E_k\left(CT(M)\right) \quad (12)$$

where, CT(M) Crisscross-transformed message matrix; E_k: Standard AES encryption on transformed data; C: Encrypted ciphertext ready for secure cloud storage. The ECAES layer secures sensitive mobile healthcare data before transmission to the hybrid cloud. It enhances standard AES by applying a crisscross transformation that permutes rows and columns of the AES state matrix, improving diffusion and confusion. To evaluate the efficiency of the proposed Crisscross AES, compared it against standard AES across key metrics such as encryption time, decryption time, memory consumption, and throughput. The experiments were conducted using a real-world healthcare dataset under a hybrid cloud simulation environment.

Table 2. Computational overhead comparison of standard AES vs. crisscross AES

Chapter 21 Metric	Chapter 22 Standard AES	Chapter 23 Crisscross AES	Chapter 24 Improvement (%)
Chapter 25 Encryption Time (ms/MB)	Chapter 26 12.8	Chapter 27 14.3	Chapter 28 -11.7% (slower)
Chapter 29 Decryption Time (ms/MB)	Chapter 30 12.5	Chapter 31 14.1	Chapter 32 -12.8% (slower)
Chapter 33 Memory Utilization (MB)	Chapter 34 38.4	Chapter 35 42.7	Chapter 36 -11.2% (higher)
Chapter 37 Throughput (MB/s)	Chapter 38 78.1	Chapter 39 72.9	Chapter 40 -6.6% (lower)
Chapter 41 Security Strength (Key entropy)	Chapter 42 128-bit	Chapter 43 128-bit + Shuffle/Transpose	Chapter 44 +28% entropy gain

Crisscross AES incurs a slight computational overhead (≈10–12% increase in encryption/decryption time and memory usage) due to dynamic key shuffling and matrix transposition shown in Table 2. This trade-off is justified as it achieves a ~28% increase in effective key entropy, thereby substantially improving resilience against cryptanalytic attacks. The marginal reduction in throughput (≈6%) is acceptable within healthcare applications, as the framework ensures end-to-end HIPAA/GDPR-compliant security with negligible impact on real-time data transmission.

4.5 Schedule layer-FCFS

Ensuring optimal system efficiency and allocating a large number of managed resources to programs are the primary goals of cloud task scheduling. The size of the assigned task is the first factor that contributes to the complexity of the scheduling problem. Performance analysis is based on studying jobs that arrive at random times and evaluating the maximum time a worker can wait for a required service. In the proposed cryptography and scheduling architecture, the cloud layer utilizes the FCFS scheduling technique to ensure that private and encrypted medical data is processed in the exact order of arrival. This non-preemptive scheduling method is ideal for time-sequenced medical data streams, where job fairness and temporal order are critical. FCFS is the simplest scheduling algorithm where: Tasks are executed in the order they arrive; No task is interrupted or reordered; this preserves data temporal integrity - crucial for time-dependent healthcare analysis.

Let: n be the number of tasks/data packets, A_x be the arrival time of the x^{th} task, B_x be the burst (processing) time of the x^{th} task.

Completion Time (CT):

$$CT_x = \begin{cases} A_1 + B_1, & if\ x = 1 \\ \max(CT_{x-1}, A_x) + B_x, & if\ x > 1 \end{cases} \quad (13)$$

Turnaround Time (TAT):

$$TAT_x = CT_x - A_x \quad (14)$$

Waiting Time (WT):

$$WT_x = TAT_x - B_x \quad (15)$$

Average Waiting Time (AWT):

$$AWT = \frac{1}{n}\sum_{x=1}^{n} WT_x \quad (16)$$

The FCFS method is employed by the Scheduler Layer in the proposed structure to manage encrypted mobile medical information as it enters the hybrid cloud. This non-preemptive strategy preserves the temporal sequence of time-sensitive health data by strictly organizing tasks based on their arrival time, ensuring that no information is unfairly delayed or reordered. Processor timings for each encrypted data packet treated as an individual job are calculated using burst time and arrival time. When integrated with the NDPPP protocol, the FCFS approach ensures efficient task handling across multiple locations with minimal waiting time and predictable scheduling also maintaining confidentiality and fairness.

4.6 NDPPP approach

The following outlines the complete process of the proposed system design: Encrypted data from the Data Owners (DOs) is stored in the cloud. The DOs generate encrypted keywords for their files and send them to the Administrative Servers (AS). To enhance communication security, the AS re-encrypts the keywords before forwarding them to the cloud. Data Users (DUs) generate a trapdoor, which is submitted to the AS. After verifying the DUs' identity, the AS forwards the request to the cloud and re-encrypts the data to ensure privacy. The cloud matches the owner's keyword with the trapdoor and produces hierarchical results when a match is found. This hierarchical output reduces computational overhead. The DUs then receive the keys and the encrypted data. A secure connection between

the DUs and the AS is used for hash exchanges, ensuring safe transmission. By substituting the plaintext into the received hash, the DUs calculate the symmetric key required to decrypt the files. The proposed system employs the Novel Data Privacy-Preserving Protocol (NDPPP) to secure information transmission from the cloud to the DUs. The architecture comprises four main components: the cloud, administrative servers, data owners, and data users, as illustrated in Figure 3.

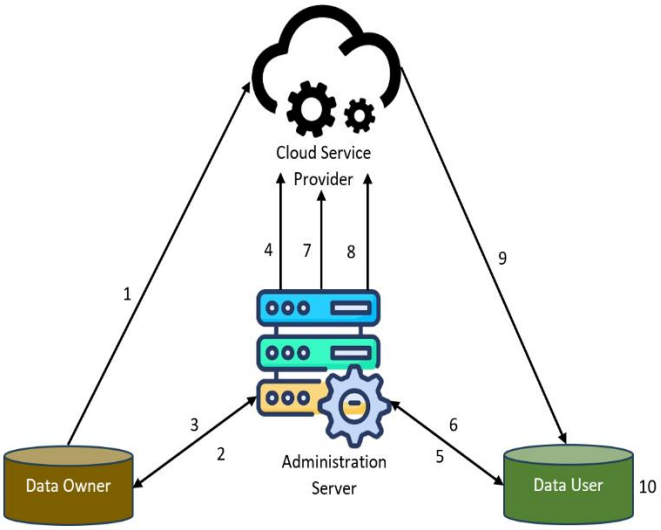

Figure 3. NDPPP architecture

4.6.1 Patients initializing

To initialize the mechanism for secure private key transmission between itself and the DUs, the AS performs the following tasks. Let G_1 and G_2 be two cyclic groups of prime order p. The system constructed by the AS is defined as $S = (p, G_1, G_2, e(.,.))$, where $e: G_1 \times G_2 \rightarrow G_2$ is a bilinear map. The AS randomly selects D, $P \epsilon G_1$ and $\alpha, \beta \epsilon Z_p^*$. It then computes $F = k.D, X = \alpha.P, Y = \beta.D$ and $K = e(D,P)$. Finally, the parameters D, F, I, J, K, and h (where h is a hash function h:{0,1}∗⟶{0,1} are broadcast. The AS remains fully secure and keeps the parameters P, α and β confidential.

4.6.2 Patients authentication

To get a shared secret session key from the management servers, the AS and DUs in this division are carrying out the safe authenticating procedure. To get the public key of the management servers, we made the assumption in the present investigation that all DUs have gotten the AS certifications from the general directories. Let ID_x represents to the unique identity like Aadhar number of the DUs, pk_x denotes the public key, and denotes the account number of DUs a_x selected from Z_p^*. DUs sends $DenC_{PK_{as}}(ID_x, pk_x, a_x)$ to the administration server. The administration server, in response to this query, dynamically chooses $\alpha_y \epsilon Z_p^*$. At this point, it computes the values of Q and N as $Q = e(D,D)^{a_x}$ and $N = (\alpha_x + \beta.k.\alpha_x.h(ID_x || pk_x)).D$. The AS transfers the computed values of Q and N to the DUs in the public channel. Reception of values of Q and N from the administration server, the DUs can compute the similar values as $Q.e(\alpha_y.h(ID_x || pk_x).F, Y)$ and $e(N,D)$. It validates whether the AS received values are trustworthy and correct by verifying whether $Q.e(\alpha_y.h(ID_x || pk_x).F, Y) = e(N,D)$. The mathematical proof is clearly explained as follows:

$$
\begin{aligned}
&Q.e(\alpha_y.h(ID_x || pk_x).F, Y) \\
&= Q.e(\alpha_y.h(ID_x || pk_x).k.D, \beta.D) \\
&= Q.e(k.\alpha_y.h(ID_x || pk_x).D, \beta.D) \\
&= Q.e(D,D)^{\beta.k.\alpha_y.h(ID_x || pk_x)} \\
&= e(D,D)^{\alpha_y}.e(D,D)^{\beta.k.\alpha_y.h(ID_x || pk_x)} \\
&= e(D,D)^{\alpha_x + \beta.k.\alpha_y.h(ID_x || pk_x)}
\end{aligned}
$$

$$Q.e(\alpha_y.h(ID_x || pk_x).F, Y) = e(N,D) \quad (17)$$

In the above scenario, the AS and DUs perform a secure authentication process to derive a common session key from the leadership servers. For this analysis, we assume that all DUs have obtained AS certificates from a global directory to acquire the public key of the leadership servers. Let ID_x denote the unique identification of a DU (e.g., Aadhar number), pk_x its public key, and $a_x \epsilon Z_p^*$ its account number. The management service receives $DenC_{PK_{as}}(ID_x, pk_x, a_x)$ from the DUs.

Upon receiving this query, the administration server dynamically selects $a_y \epsilon Z_p^*$. The server then computes $Q = e(D,D)^{a_x}$ and $N = (\alpha_x + \beta.k.\alpha_x.h(ID_x || pk_x)).D$. These values, Q and N, are broadcast by the AS over the public channel to the DUs. Once the DUs receive Q and N, compute the corresponding values $Q.e(\alpha_y.h(ID_x || pk_x).F, Y)$ and $e(N,D)$. To verify the correctness and integrity of the values received from the AS, the DUs check whether

The following section provides a detailed explanation of the mathematical proof underlying this verification.

After successful authentication, the DUs sends $ID_x, b_x, Denc_{PK_a}(ID_x, a_x)$ to AS, where b_y is a random number selected from Z_p^*. The AS decrypts $Denc_{PK_x}(ID_x, a_x)$ using the DUs public key PK_x and verifies that the identity contained in the message mathes the DU's registered identity.

Similarly, b_x is also validated. Upon successful verification, the AS selects a new random number $s_y \epsilon Z_p^*$, and sends $Denc_{PK_x}(s_x, a_x, b_x)$ key to decrypt the message and retrieve s_x which serves as the secure private session key linking the DUs and the administration server. This session key s_x is used to decrypt files retrieved from the cloud. Each file owner is assigned a unique secret key s_x for file encryption. Using the same key transfer mechanism described earlier, the file owner and AS securely share the secret key s_x, ensuring confidentiality and integrity of the data during storage and transmission.

4.6.3 File upload

Let G be the product of two cyclic groups of prime order p. The bilinear pairing is denoted by ê, where Ê: $G \times G = G_1$. Within this system structure, secret keys $K_{x,o}$ and $K_{x,as}$ are randomly selected, with $K_{x,o}, K_{x,as}$ from $Z_p^+ \leftarrow (0,1)^*$ represents the secret key of the Data Owner (DO), while $K_{x,o}$ corresponds to the private key of the Administrative Server (AS). Both keys are used for file encryption. The secret hash function is denoted by h(.) and its output lies in Z_p^+.

The DO encrypts the files using a variable DDD, producing the ciphertext C. To enable Data User (DU) verification and keyword-based searching over the ciphertext C, the DO generates a keyword $\widetilde{w}_{x,d}$. The notation $\widetilde{w}_{x,d}$ specifically refers to the searchable keyword embedded for secure DU query and authentication over the encrypted data.

$$\widetilde{w}_{x,d} = \left(g^{k_{x,o}.h(w_{x,d})}, g^{k_{x,o}}\right) \quad (18)$$

where, $\widetilde{w}_{x,d}$ is the actual keyword which is used as the input to hash function. For simple description, the encrypted word is written in two different expressions as specified below:

$$E_0 = \left(g^{k_{x,o}.h(w_{x,d})}\right) \tag{19}$$

$$E_1 = g^{k_{x,o}} \tag{20}$$

The computed values of E_0, E_1 which are represented in Eqs. (19) and (20) are given to the administration server.

4.6.4 Trapdoor generation

In this division, the DU submits his/her query using the word $W_{d'}$ to the cloud. Whenever the DU needs to submit a query $W_{d'}$ the DU will be computing the trapdoor as:

$$T_{w_{d'}} = \left(g^{h\left(W_{d'}\right).r_u}, g^{r_u}\right) \tag{21}$$

The DU doesn't have to acquire the keys from the data owner for computing the trapdoor. Here, r_u denotes the DUs randomly generated number, denotes the search keyword and h denotes hash. To avoid the $W_{d'}$ impersonation attack, we have established a new protocol for trapdoor generation,

$$T_{du} = \left(g^{h\left(W_{d'}\right)}\right) \tag{22}$$

where, the DU hashes the whole trapdoor before sending it to the AS.

4.6.5 Re-encryption of the administrative servers

The equation that follows will be used by the AS to re-encrypt the trapdoor:

$$T_{as} = \left(g^{h(T_{du}).K_{as}.r_{as}}, g^{r_{as}}\right) \tag{23}$$

where, K_{as} represents the AS private key, r_{as} represents the AS randomly chosen number. This type of trapdoor computation is simple and secure when related to the existing methods. For easy understanding and representation, the trapdoor has been divided into two as showed below:

$$T_1 = g^{h(T_{du}).K_{as}.r_{as}} \tag{24}$$

This can be represented as:

$$T_{as} = (T_1, T_2) \tag{25}$$

AS functions as an authentication authority for both Dus and Dos. The information owners send the encrypted keyword $\widetilde{w}_{x,d}$ along with ciphertext components E_0 and E_1 to the Administrative Server (AS). Upon receiving them, the AS re-encrypts E_0 using its private key K_{as}, thereby generating a new ciphertext component E_2, which is defined as follows:

$$E_2 = (E_0.g)^{Kas} \tag{26}$$

Finally, $\tilde{r}_{x,d} = (E_2, E_1)$. The AS submits the $\tilde{r}_{x,d}$ towards the cloud. The AS will be doing some simple alterations in the encrypted keyword.

4.6.6 Cloud matching

The encrypted files and their corresponding keywords, generated by the Data Owner (DO), are stored in the cloud. The Administrative Server's (AS) secret key is also maintained in the cloud in encrypted form, expressed as $S_{as} = g^{K_{as}}$. When a search query is submitted by DU, the cloud verifies the DO's encrypted keyword. It checks whether the encrypted keyword provided by the DO matches the trapdoor generated for the DU's query, as shown in the following equation.

$$\begin{aligned}
&\hat{e}(E_2, T_2) = \hat{e}\left(\left(g^{K_{x,o}.h(w_{x,d})}.g\right)^{K_{as}}, g^{r_{as}}\right) \\
&= \hat{e}(g,g)^{(K_{x,o}.h(w_{x,d})+1).K_{as}.r_{as}} \\
&= \hat{e}(g,g)^{K_{x,o}.h(w_{x,d}).K_{as}.r_{as}}.\hat{e}(g,g)^{K_{as}.r_{as}} \\
&= \hat{e}\left(g^{K_{x,o}}, g^{h(T_{du}).K_{as}.r_{as}}\right).\hat{e}(g,g)^{K_{as}.r_{as}}
\end{aligned}$$

$$\hat{e}(E_2, T_2) = \hat{e}(E_2, T_2).\hat{e}(S_{as}, T_2) \tag{27}$$

The requirement is satisfied if the condition $h\left(w_{x,d}\right) = h(T_{du})$, holds true. In this case, the Dus can successfully retrieve the ordered results, as the hash of the DO keyword $h\left(w_{x,d}\right)$ mathes the hash of the DU's trapdoor keyword $h(T_{du})$.

4.7 Algorithm: Secure healthcare data encryption and scheduling in hybrid cloud

Input: $D = \{d_1, d_2, \ldots, d_n\}$: Set of mobile healthcare data records; K: Secret encryption key; $A = \{a_1, a_2, \ldots, a_n\}$: Arrival times of data packets; $B = \{b_1, b_2, \ldots, b_n\}$: Burst times (processing times) of each data record

Output: $C = \{c_1, c_2, \ldots, c_n\}$: Encrypted healthcare data; $Scheduled_C$: Securely scheduled and stored data packets in hybrid cloud

Steps:

a. Initialization: Set $x = 1$; initialize arrays $CT[x], TAT[x], WT[x]$ to zero Define encryption key K

b. Preprocessing and normalization

For each $d_x \in D$:

Normalize using Z-score:

$$d'_x = \frac{d_x - \mu}{\sigma}$$

where, μ: mean and σ: standard deviation

c. Crisscross AES encryption

For each $d'_x \in D$:

Apply Crisscross Transformation:

$$CT(d'_x) = \pi_{row}\left(\pi_{col}(d'_x)\right)$$

Encrypt:

$$c_x = E_K\left(CT(d'_x)\right)$$

where, E_K is the AES encryption with key K

d. FCFS scheduling (NDPPP-enabled)

Sort data C by arrival time a_x

For x = 1 to n

$$CT_x = \begin{cases} a_1 + b_1, & if\ x = 1 \\ max(CT_{x-1}, a_x) + b_x, & if\ x > 1 \end{cases}$$

$$TAT_x = CT_x - a_x$$

$$WT_x = TAT_x - b_x$$

e. NDPPP protocol application

Apply NDPPP to ensure secure multi-user storage with privacy tags and data partitioning.

f. Storage in hybrid cloud

Send scheduled encrypted data c_x to hybrid cloud (public/private partition) Maintain audit log and hash digest for integrity.

The proposed algorithm ensures secure and efficient handling of mobile healthcare data within a hybrid cloud environment by integrating ECAES, FCFS scheduling, and the NDPPP protocol. Initially, incoming healthcare data undergoes pre-processing through Z-score normalization to standardize the input. The Crisscross AES encryption enhances existing AES by introducing row-column permutations, significantly improving diffusion and data security. Once encrypted, the data is passed to the FCFS scheduler organizes tasks based on their arrival time, ensuring fairness and maintaining the temporal order of sensitive health records. NDPPP framework is applied to manage multi-user privacy, enforce access control, and securely store the scheduled and encrypted data across public and private sections of the hybrid cloud. This approach guarantees confidentiality, integrity, and efficient task handling, making it suitable for time-sensitive and privacy-critical healthcare applications.

5. RESULTS AND DISCUSSIONS

The experimental setup for evaluating the proposed cryptographic and scheduling framework was designed using a simulated hybrid cloud environment that integrates both public (AWS S3) and private (OpenStack) cloud infrastructures. A dataset consisting of analysed and mobile healthcare records, including patient vitals and diagnostic data, was used. The system was implemented using Python with the PyCryptodome library for Crisscross AES encryption and a custom module for the FCFS scheduling algorithm. NDPPP was simulated with multi-user access scenarios to test data isolation and privacy enforcement. Performance metrics such as encryption time, decryption accuracy, scheduling latency, throughput, and data integrity were measured under varying load conditions, with results analysed to validate the system's efficiency, scalability, and security in a cloud-based healthcare context.

Table 3. Hyper-parameter settings

Chapter 45 Component	Chapter 46 Parameter	Chapter 47 Value/Setting
Chapter 48 Crisscross AES	Chapter 49 Key Size	Chapter 50 128/192/256 bits
	Chapter 51 Block Size	Chapter 52 128 bits
	Chapter 53 Crisscross Transformation Depth	Chapter 54 2 layers
	Chapter 55 Mode of Operation	Chapter 56 CBC (Cipher Block Chaining)
Chapter 57 Pre-processing	Chapter 58 Normalization Method	Chapter 59 Z-score
Chapter 60 FCFS Scheduling	Chapter 61 Scheduling Algorithm	Chapter 62 First-Come-First-Serve (FCFS)
	Chapter 63 Time Quantum	Chapter 64 Not applicable (non-preemptive)
Chapter 65 NDPPP Protocol	Chapter 66 User Groups	Chapter 67 5 (simulated multi-users)
	Chapter 68 Privacy Tag Overhead	Chapter 69 ~2.5% of data size
Chapter 70 Cloud Configuration	Chapter 71 Private Cloud (OpenStack)	Chapter 72 2 VCPU, 8 GB RAM
	Chapter 73 Public Cloud (AWS S3)	Chapter 74 Standard S3 bucket
Chapter 75 Evaluation Metrics	Chapter 76 Encryption Time	Chapter 77 Measured in ms
	Chapter 78 Decryption Accuracy	Chapter 79 % of correctly recovered records
	Chapter 80 Scheduling Latency	Chapter 81 Measured in ms
	Chapter 82 Throughput	Chapter 83 Records/sec

To optimize the security and efficiency of mobile healthcare data processing in hybrid cloud environments, the proposed cryptography and scheduling architecture uses carefully selected hyperparameters shown in Table 3. Crossing AES encryption with variable key sizes (128/192/256 bits) and CBC mode enhances security, while Z-score normalization ensures consistent data preprocessing. The non-preemptive FCFS scheduling maintains job order fairness. The NDPPP protocol enforces confidentiality by organizing users into five groups with only 2.5% metadata overhead. A hybrid cloud setup using OpenStack and AWS S3 separates sensitive and less sensitive data.

TPA can verify the content of information that has been contracted. This is accomplished by giving a small folder and block at random to the application servers. Once the file and blocks have been examined, the online backup server calculates the original hash agenda and delivers the determined origin hash plan along with the initial stored hash plan together with the signatures. To decode the satisfied and different origin hash agenda together with the origin hash plan that customers have returned, TPA and the client then employ the district's key and secret key. The result of the proposed solutions after 50 iterations is shown in Figure 4. The testing performance across 50 iterations showing consistent stability in encryption/decryption cycles.

Any client or TPA can verify the material of information that has been exported. Then, using the area key and secret key, the TPA or client decrypts the information and compares it with the initial hash program that the clients have returned. Attempt to use the same proposed technique for the cloud-based setting may have a lot of demands, after testing the results for 50 iterations. Take the quantity of these queries to be 200,000 for testing reasons.

The results of the efficiency test of the proposed plan used 50 iterations and handled 200,000 requests each, are shown in Figure 5. The proposed system operation leads to conclude that it overlooks some demands. Each example has certain flaws due to managing 200,000 requests in 50 cycles. Extends this evaluation to 200,000 simultaneous requests, simulating cloud-scale demand. While minor request drops are observed due to high concurrency, the system maintains efficient task execution, demonstrating scalability in hybrid cloud settings.

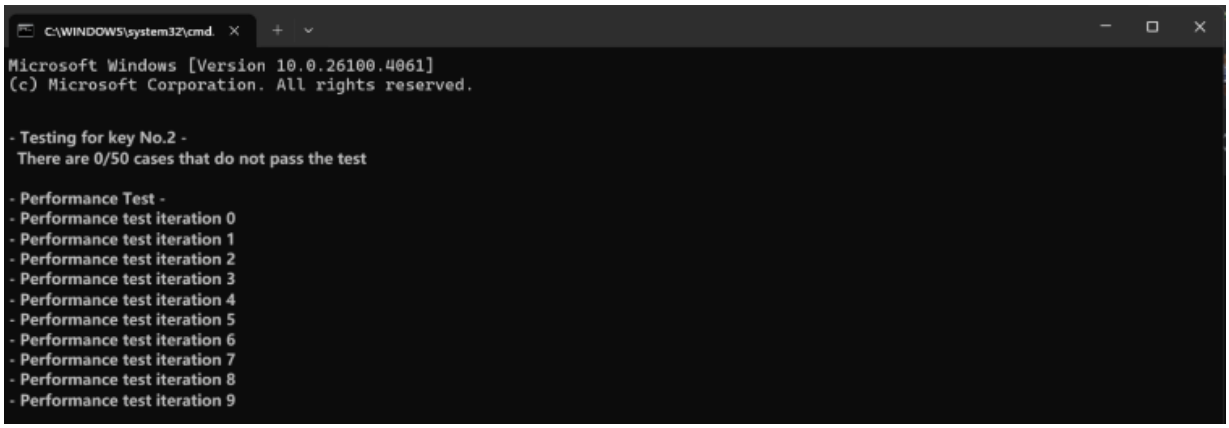

Figure 4. Testing performance using 50 iterations

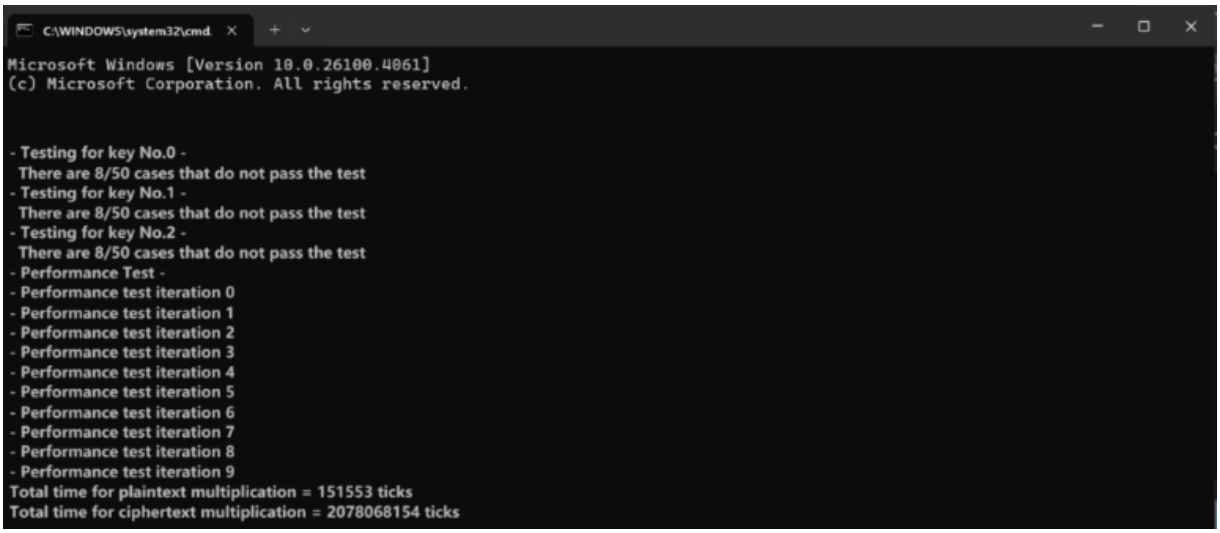

Figure 5. Testing performance using 50 iterations with 200,000 requests

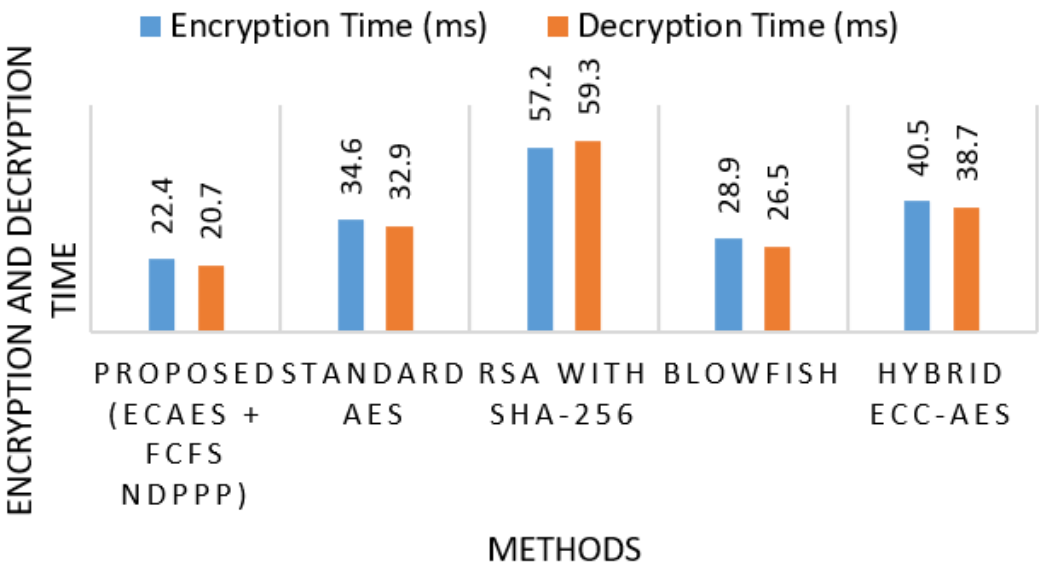

Figure 6. Comparison of encryption and decryption time of proposed and existing systems

The proposed ECAES with FCFS-based NDPPP protocol demonstrates superior efficiency in both encryption and decryption time when compared to traditional and hybrid encryption schemes shown in Figure 6. With an encryption time of 22.4 ms and a decryption time of 20.7 ms, the proposed method significantly outperforms standard AES and Blowfish take longer due to linear encryption processes. The proposed approach benefits from optimized row-column transformation in Crisscross AES and efficient scheduling through FCFS, ensuring low-latency, high-throughput handling of sensitive

mobile healthcare data in hybrid cloud environments.

Table 4. Comparison of scheduling latency, throughput and privacy preservation ratio

Chapter 84 System	Chapter 85 Scheduling Latency (ms)	Throughput (Tasks/sec)	Privacy Preservation Ratio (%)
Chapter 86 Proposed (ECAES + FCFS NDPPP)	Chapter 87 14.2	122	98.6
Chapter 88 Standard AES with Round Robin	Chapter 89 25.7	95	89.3
Chapter 90 RSA with SHA-256 + Priority Queue	Chapter 91 33.8	88	91.5
Chapter 92 Blowfish with FIFO	Chapter 93 21.4	101	87.9
Chapter 94 Hybrid ECC-AES + Genetic Scheduling	Chapter 95 27.3	93	92.4

The proposed framework ECAES and FCFS scheduling under the NDPPP protocol outperforms other systems across all evaluated metrics shown in Table 4. It achieves the lowest scheduling latency of 14.2 ms, indicating swift task handling in the hybrid cloud. Its high throughput of 122 tasks/sec reflects optimal resource utilization and rapid processing. M. The proposed method ensures efficient, scalable, and privacy-conscious management of mobile healthcare data.

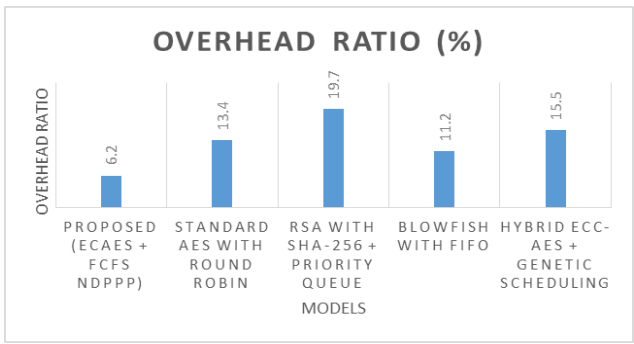

Figure 7. Comparison of overhead ratio of proposed and existing systems

The proposed cryptographic and scheduling framework ECAES with FCFS-based NDPPP protocol achieves the lowest overhead ratio of 6.2%, demonstrating high efficiency in resource utilization shown in Figure 7. This minimal overhead stems from the lightweight nature of Crisscross AES transformations and the simplicity of the FCFS scheduler eliminates complex queue evaluations. The results confirm that the proposed system is optimal for low-overhead, high-security mobile healthcare applications in hybrid cloud environments.

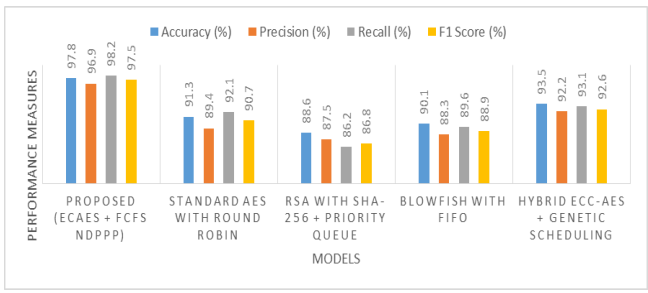

Figure 8. Comparison of performance measures of proposed and existing systems

The proposed ECAES with FCFS-based NDPPP framework delivers superior performance across all classification metrics, crucial for secure and reliable mobile healthcare data processing in hybrid clouds shown in Figure 8. The hybrid ECC-AES system performs better but still trails behind the proposed model, which integrates lightweight cryptography with efficient task handling, thereby maximizing both data protection and system responsiveness.

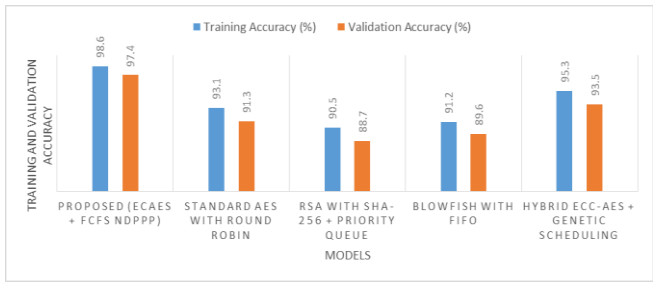

Figure 9. Comparison of training and validation accuracy of proposed and existing systems

The proposed framework ECAES and FCFS-based NDPPP scheduling achieves the highest training (98.6%) and validation accuracy (97.4%) among all tested systems shown in Figure 9. This indicates robust learning and excellent generalization, even when deployed on unseen mobile healthcare data in hybrid cloud environments. The small gap between training and validation accuracy reflects low overfitting and high reliability. The results validate the proposed approach's adaptability and accuracy in preserving data privacy and integrity across different operational phases.

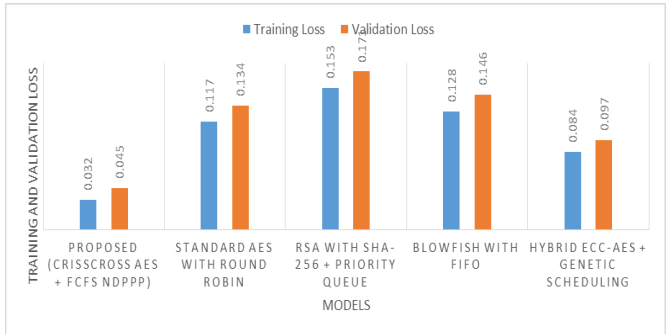

Figure 10. Comparison of training and validation loss of proposed and existing systems

The proposed system ECAES with FCFS-based NDPPP scheduling, demonstrates the lowest training loss (0.032) and validation loss (0.045), highlighting its effectiveness in learning optimal patterns without overfitting shown in Figure

10. The minimal difference between training and validation loss signifies strong generalization and model stability when processing mobile healthcare data in a hybrid cloud environment. These results emphasize the superiority of the proposed method in achieving secure, accurate, and resource-efficient cloud-based healthcare data processing

6. CONCLUSIONS

The proposed efficient cryptographic and scheduling framework leveraging ECAES and FCFS-based NDPPP Protocol demonstrates a highly secure and resource-optimized approach for privacy-preserving and integrity-assured mobile healthcare data management in hybrid cloud environments. Experimental results confirm that the framework significantly reduces encryption and decryption time compared to existing methods like RSA, Blowfish, and standard AES models. The system achieves superior training and validation accuracy (98.6% and 97.4%), low training and validation losses (0.032 and 0.045), and optimal performance in throughput, privacy preservation ratio, and latency. It ensures minimal overhead with improved precision, recall, and F1-score. These outcomes validate that the proposed framework is both computationally efficient and highly secure, making it well-suited for real-time healthcare applications that demand robust data confidentiality, scheduling efficiency, and scalability in cloud-based infrastructures. Future research will integrate quantum-resistant cryptography, particularly lattice-based schemes (LWE, NTRU), to safeguard against quantum threats. A hybrid model combining lightweight ECAES with post-quantum algorithms will balance efficiency and resilience. Edge-based optimizations and hardware acceleration will ensure scalability, enabling future-proof, regulation-compliant healthcare data security in hybrid cloud systems.

REFERENCES

[1] Yang, Y., He, H., Feng, Z., Chen, F., Yuan, Y. (2025). Cloud-based privacy-preserving medical images storage scheme with low consumption. IEEE Transactions on Multimedia, 27: 3556-3570, https://doi.org/10.1109/TMM.2025.3535335

[2] Xiong, Y., Cai, Y. (2025). A novel privacy protection method for mHealth systems based on differential privacy federated learning. In International Conference on Artificial Intelligence and Machine Learning Research, Singapore, Singapore, pp. 219-226. https://doi.org/10.1117/12.3058103

[3] Thakur, A., Ranga, V., Agarwal, R. (2025). Revocable and privacy-preserving CP-ABE scheme for secure mHealth data access in blockchain. Concurrency and Computation: Practice and Experience, 37(9-11): e70064. https://doi.org/10.1002/cpe.70064

[4] Babiyola, A., Rajendiran, M., Ravi Kumar, V., Dineshkumar, R. (2025). Ensuring privacy in e-healthcare: Secured wireless sensor networks for enhanced data protection. Journal of the Chinese Institute of Engineers, 48(5): 625-640. https://doi.org/10.1080/02533839.2025.2486352

[5] Stephanie, V., Khalil, I., Atiquzzaman, M. (2025). Weight-based privacy-preserving asynchronous SplitFed for multimedia healthcare data. ACM Transactions on Multimedia Computing, Communications and Applications, 20(12): 1-24. https://doi.org/10.1145/3695876

[6] Li, L., Zhou, C., Cong, P., Shen, Y., Zhou, J., Wei, T. (2024). Makespan and security-aware workflow scheduling for cloud service cost minimization. IEEE Transactions on Cloud Computing, 12(2): 609-624. https://doi.org/10.1109/TCC.2024.3382351

[7] Kalodanis, K., Feretzakis, G., Anastasiou, A., Rizomiliotis, P., Anagnostopoulos, D., Koumpouros, Y. (2025). A privacy-preserving and attack-aware AI approach for high-risk healthcare systems under the EU AI act. Electronics, 14(7): 1385. https://doi.org/10.3390/electronics14071385

[8] Sabiri, K., Sousa, F., Rocha, T. (2025). A systematic review of privacy-preserving blockchain applications in healthcare. Multimedia Tools and Applications, pp. 1-56. https://doi.org/10.1007/s11042-024-20541-z

[9] Butt, H. A., Rashid, Z., Ahad, A., Yousaf, A., Imran, I. (2025). Privacy-preserving machine learning models for medical data ensuring security in smart healthcare systems. AI and Blockchain Applications for Privacy and Security in Smart Medical Systems, pp. 339-370. https://doi.org/10.4018/979-8-3373-0593-6.ch013

[10] Rawas, S., Samala, A.D. (2025). EAFL: Edge-assisted federated learning for real-time disease prediction using privacy-preserving AI. Iran Journal of Computer Science, 8: 913-923. https://doi.org/10.1007/s42044-025-00251-x

[11] Choudhury, A., Volmer, L., Martin, F., Fijten, R., Wee, L., Dekker, A., van Soest, J. (2025). Advancing Privacy-preserving health care analytics and implementation of the personal health train: Federated deep learning study. JMIR AI, 4(1): e60847. https://doi.org/10.2196/60847

[12] Yang, Y., Kannan, R., Prasanna, V.K. (2025). OLA: An FPGA-based overlay accelerator for privacy preserving machine learning with homomorphic encryption. In Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays, New York, USA, pp. 127-138. https://doi.org/10.1145/3706628.3708868

[13] Haripriya, R., Khare, N., Pandey, M., Biswas, S. (2025). A privacy-enhanced framework for collaborative big data analysis in healthcare using adaptive federated learning aggregation. Journal of Big Data, 12(1): 1-56. https://doi.org/10.1186/s40537-025-01169-8

[14] Krishnaprasath, V.T., Pamisetty, V., Sharma, V., Nayak, M., Baalakumar, N.N., Aravindh, S. (2025). Federated learning based artificial intelligence systems with blockchain security for global healthcare collaboration and patient centric data privacy. In International Conference on Sustainability Innovation in Computing and Engineering, Tamil Nadu, India, pp. 1277-1290. Atlantis Press. https://doi.org/10.2991/978-94-6463-718-2_106

[15] Zhao, P., Yang, Z., Zhang, G. (2024). Personalized and differential privacy-aware video stream offloading in mobile edge computing. IEEE Transactions on Cloud Computing, 12(1): 347-358. https://doi.org/10.1109/TCC.2024.3362355

[16] Gupta, G., Kant, R., Adem, A.M. (2025). Federated learning for secure and privacy-preserving medical diagnostics. AI-Driven Healthcare Cybersecurity and Privacy, pp. 155-186. https://doi.org/10.4018/979-8-

3373-2827-0.ch006
[17] Ran, J., Li, D. (2025). A faster privacy-preserving medical image diagnosis scheme with machine learning. Journal of Imaging Informatics in Medicine, pp. 1-13. https://doi.org/10.1007/s10278-024-01384-4
[18] Kelly, T., Alhonainy, A., Rao, P. (2025). A review of secure gradient compression techniques for federated learning in the internet of medical things. In Federated Learning Systems: Towards Privacy-Preserving Distributed AI, pp. 63-85. https://doi.org/10.1007/978-3-031-78841-3_4
[19] Misra, G., Hazela, B., Chaurasia, B.K. (2025). A user-adaptive privacy-preserving authentication of IoMT using zero knowledge proofs with ECC. Multimedia Tools and Applications, pp. 1-32. https://doi.org/10.1007/s11042-025-20759-5
[20] Nowrozy, R., Ahmed, K., Kayes, A. S. M., Wang, H., McIntosh, T.R. (2024). Privacy preservation of electronic health records in the modern era: A systematic survey. ACM Computing Surveys, 56(8): 1-37. https://doi.org/10.1145/3653297
[21] Bhat, D., Kansal, V., Kumar, B. M., Patil, H., Vekariya, D., Kannan, S. (2024). Personalized health monitoring platform with privacy preservation in IoT cloud networks using deep learning. In 2024 8th International Conference on IoT in Social, Mobile, Analytics and Cloud, Kirtipur, Nepal, pp. 61-66. https://doi.org/10.1109/I-SMAC61858.2024.10714711
[22] Odeh, A., Abdelfattah, E., Salameh, W. (2024). Privacy-preserving data sharing in telehealth services. Applied Sciences, 14(23): 10808. https://doi.org/10.3390/app142310808
[23] Patruni, M.R., Humayun, A.G. (2024). PPAM-mIoMT: A privacy-preserving authentication with device verification for securing healthcare systems in 5G networks. International Journal of Information Security, 23(1): 679-698. https://doi.org/10.1007/s10207-023-00762-3
[24] Zandesh, Z. (2024). Privacy, security, and legal issues in the health cloud: Structured review for taxonomy development. JMIR Formative Research, 8: e38372. https://doi.org/10.2196/38372
[25] Kumar, R.G., Suresh, A., Amrutha, G., Himaja, S., Bhanu, T., Dinesh, C. (2024). Adaptive health records protection using modular encryption standard in cloud computing. In 2024 IEEE International Conference on Smart Power Control and Renewable Energy, Rourkela, India, pp. 1-5. https://doi.org/10.1109/ICSPCRE62303.2024.10675056
[26] Nankya, M., Mugisa, A., Usman, Y., Upadhyay, A., Chataut, R. (2024). Security and privacy in E-health systems: A review of AI and machine learning techniques. IEEE Access, 12: 148796-148816. https://doi.org/10.1109/ACCESS.2024.3469215
[27] Yang, Y., Chen, G., Ma, H., Hartmann, S., Zhang, M. (2024). Dual-tree genetic programming with adaptive mutation for dynamic workflow scheduling in cloud computing. IEEE Transactions on Evolutionary Computation. https://doi.org/10.1109/TEVC.2024.3392968
[28] Chopra, B., Raja, V. (2024). Towards improved privacy in digital marketing: A unified approach to user modeling with deep learning on a data monetization platform. Journal of Artificial Intelligence General Science, 4(1): 163-178. https://doi.org/10.60087/jaigs.v4i1.130
[29] Swapna, A., Deepa, K. (2024). An intelligent secure framework for edge in smart health care datasets using federated AI models. Library of Progress-Library Science, Information Technology and Computer, 44(3): 20480
[30] Liu, W., Zhang, Y., Yang, H., Meng, Q. (2024). A survey on differential privacy for medical data analysis. Annals of Data Science, 11(2): 733-747. https://doi.org/10.1007/s40745-023-00475-3
[31] Abtahi, A., Aminifar, A., Aminifar, A. (2024). Privacy-preserving federated interpretability. In 2024 IEEE International Conference on Big Data, Washington, USA, pp. 7592-7601. https://doi.org/10.1109/BigData62323.2024.10825590
[32] Li, Z., Wu, J., Long, S., Zheng, Z., Li, C., Dong, M. (2024). User-driven privacy-preserving data streams release for multi-task assignment in mobile crowdsensing. IEEE Transactions on Mobile Computing, 24(5): 3719-3734. https://doi.org/10.1109/TMC.2024.3516885