

Decentralized Electronic Health Record Management with Semantic-Aware Hierarchical Encryption

Firdaus Putra Kurniyanto¹, Bagus Jati Santoso²

^{1,2}Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received May 28, 2025

Revised Jul 11, 2025

Accepted Aug 11, 2025

Keywords:

blockchain
electronic health record
hierarchical encryption
medical data security
semantic encryption

ABSTRACT

The rising incidence of cyberattacks targeting electronic health records (EHR) in Indonesia necessitates a robust and context-aware data protection scheme. This paper proposes a decentralised EHR management system that leverages blockchain, IPFS, and a novel Semantic-Aware Hierarchical Encryption (SAHE) algorithm. SAHE enables multi-level access control based on data sensitivity semantics, ensuring privacy while maintaining usability for medical professionals. The system was implemented in a prototype environment and evaluated through stress testing with up to 200 users, achieving an average CPU usage of 55% and a memory consumption of 80.2 MB. Differential cryptanalysis demonstrated a strong avalanche effect (~50%), with no vulnerabilities found via OWASP ZAP scanning. This architecture offers a promising solution for privacy-preserving, patient-controlled EHR systems, particularly in regions with limited infrastructure.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Firdaus Putra Kurniyanto,
Department of Informatics,
Institut Teknologi Sepuluh Nopember,
Surabaya, Indonesia.
Email: 6025232020@student.its.ac.id

1. INTRODUCTION

In today's digital era, the proliferation of sensitive medical data, such as radiographic images (X-ray data), has significantly revolutionized healthcare services. Digital medical data enables more accurate diagnoses, personalized treatments, and more efficient and integrated patient management. With Electronic Health Records (EHR), doctors and other healthcare professionals can easily access patient health histories, monitor disease progression, and plan more targeted medical interventions. Additionally, telemedicine technology and the Internet of Medical Things (IoMT) are increasingly expanding the reach of healthcare services, enabling remote consultations and real-time patient condition monitoring.

However, the increased dependence on electronic medical records also brings significant challenges related to data security, privacy, and accessibility. In Indonesia, where the healthcare system is undergoing rapid modernisation, protecting sensitive medical information from unauthorised access, data breaches, and corruption has become increasingly crucial. This is important not only to protect patient privacy but also to ensure the integrity and availability of medical data, which are essential for treatment processes and informed medical decision-making.

Indonesia has experienced a significant increase in cyberattacks targeting healthcare institutions in recent years. These attacks not only result in unauthorized access and breaches of sensitive medical data but also threaten public trust in digital health systems. For instance, in 2021,

it was reported that 720 GB of private medical information was sold on the dark web, originating from several hospitals in Indonesia. Additionally, a report from the Jakarta Post in the same year revealed cases of health data breaches related to COVID-19 vaccine certificates, demonstrating the vulnerability of digital health systems to cyberattacks.

Although the healthcare industry has adopted modern EHR, telemedicine, and IoMT technologies, this sector is still not fully prepared to face evolving cyber threats [1]. Many medical devices and IT systems used in the healthcare sector have low levels of protection, making them easy entry points for cybercriminals. Medical data has high value in the black market because it contains personal information that can be used for identity theft and other fraudulent activities. Reports indicate that cyberattacks on health data have increased significantly, with a 162% rise in the last three years (as of 2022), underscoring the urgent need to enhance medical data security.

Advances in distributed storage technology pose new challenges in maintaining the confidentiality of health data. Patient privacy is a significant concern in EHR implementation [2]. Storing EHR in public networks, such as the InterPlanetary File System (IPFS), means that data is vulnerable to unauthorised access if not properly encrypted.

In addressing these challenges, the concept of decentralization has been proposed as a potential solution to enhance the security and integrity of medical data. By utilising blockchain or decentralised systems, the issue of central authority can be addressed; moreover, decentralised systems make data immutable or unchangeable. In this approach, patient medical records are stored in a decentralised manner using the InterPlanetary File System (IPFS), which enables immutable and easily distributed medical records without relying on third parties [3]. This architecture is designed to be patient-centred, allowing patients to have complete control over their data and grant or revoke access at any time. The system also combines symmetric and asymmetric cryptographic keys to store and access medical records securely. Additionally, the use of biometrics, such as fingerprints, ensures that only patients have access to their private keys, which are used to decrypt their medical records.

One consideration in this approach is that access to medical record data depends on the patient's ability to provide their fingerprint, as the encryption and decryption processes use biometric data. In situations where patients cannot provide their fingerprints, healthcare professionals may encounter difficulties in accessing the required medical records. Furthermore, this approach has not yet explained the mechanism of access control for medical record data, which may raise concerns about data security if private keys are shared without proper security measures.

Electronic Medical Record (EMR) exchange poses its challenges, such as handling privacy, data security, and patient access authorisation [4]. In that research, it is stated that traditional systems are vulnerable to cyberattacks, data manipulation, and loss of control by information owners, namely patients. Blockchain technology offers solutions by providing transparency, decentralization, and data integrity. Nevertheless, this research has not yet integrated access management based on patient preferences in the proposed scheme.

Traditional EHR systems face constraints such as limited access, security risks, and lack of interoperability between different platforms. The system built using cross-chain enables data exchange between various blockchain networks, overcoming interoperability limitations between networks [5]. This research also combines Deep Residual Network (RDN) and SpinalNet to generate cryptographic keys from internal model layers. These keys are then used for encrypting EHR data, thereby enhancing security and efficiency in data exchange. Additionally, this research utilises both on-chain and off-chain storage, where information and data hashes are stored on-chain. In contrast, complete medical record data is stored off-chain to reduce the burden on the blockchain. Although the solution offered in this research can secure and share data safely, the use of hybrid models, such as Deep RSNet, increases technical complexity, which may be challenging to implement for institutions with limited resources.

Moreover, the encryption process is a crucial part of data protection. Semantic Attribute-Based Encryption (SABE) is a combination of Attribute-Based Encryption (ABE) and semantic technology, making ABE semantic-aware [6]. SABE uses the Ciphertext-Policy ABE (CP-ABE) scheme as a foundation, enhanced with semantic technology through domain ontology, Semantic Web Rule Language (SWRL) rules, and inference engines. However, SABE has limitations related

to longer execution time, especially in the encryption process on SABE-SEAS and the key generation process on SABE-SEK.

Ciphertext Policy Attribute-Based Encryption (CP-ABE) is a promising cryptographic technique, particularly in combining data encryption and access control to enhance data security in Internet of Things (IoT) systems. However, until now, CP-ABE efficiency has been one of the obstacles hindering its development and implementation. There is a consensus that the computational overhead of bilinear pairings is too heavy for practical ABE applications, especially on devices or processors with computational and power limitations [7]. One of the main weaknesses of CP-ABE in data sharing is its reliance on a central authority for key generation, which makes it a single point of failure and vulnerable to attacks [8].

Attribute-based encryption (ABE) is a cutting-edge technology that simultaneously provides confidentiality and fine-grained access control over data stored in the cloud. However, the central authority aspect in traditional ABE presents various significant management challenges, including key escrow management, key distribution, key tracking, key rotation, and significant communication and computational burdens for users [9]. These challenges create security issues, thereby hindering the widespread adoption of ABE. Additionally, massive data volumes and various needs for data sharing present significant challenges to complex access structures [10].

Furthermore, ABE requires a trusted issuing authority, which can be a vulnerability if that authority is compromised or unavailable. In real case studies, ABE approaches are often combined with symmetric encryption to encrypt data, while ABE is used only to distribute keys securely [2].

Conventional symmetric encryption methods, such as AES, are powerful for protecting data as a whole, but they do not recognise the context or hierarchy of encrypted data. If one large EHR file is encrypted with a single AES key, access is all-or-nothing; users given the key will be able to read the entire medical record content, including parts that may not need to be known. There is no separation of which data is more sensitive. Although AES is recognised as a highly secure international encryption standard, single-use without hierarchy does not provide tiered access control.

Therefore, research to develop encryption algorithms, such as Semantic Aware Hierarchical Encryption (SAHE), becomes essential. The SAHE algorithm is designed to integrate the semantic meaning of data into encryption mechanisms, ensuring that each data category is protected according to its sensitivity level.

Considering the importance of protecting electronic medical record data and the increasing prevalence of cyber threats, this research explores a new approach to securing medical record data by integrating blockchain technology, the InterPlanetary File System (IPFS), and a Key Management System (KMS). Additionally, this research will also develop a new encryption approach called Semantic-Aware Hierarchical Encryption (SAHE), which is used for encrypting electronic medical record data. This approach aims to enhance data protection, facilitate easy access to medical record data, and provide complete access control for the relevant patients. By combining these technologies, it is hoped that a more secure, transparent, and efficient electronic medical record data storage system can be created, which not only protects data from cyber threats but also gives patients greater control over their health information.

Overall, the modernisation of healthcare systems in Indonesia presents excellent opportunities to enhance service quality through the effective utilisation of digital technology. However, challenges related to data security and privacy must be taken seriously to ensure that the benefits of this technology can be realised without compromising the protection of sensitive patient information. This research aims to make a significant contribution to this field by developing solutions that are not only safe and reliable but also tailored to the specific needs and conditions of the Indonesian healthcare sector.

The main contribution of this research lies in the integration of a novel Semantic-Aware Hierarchical Encryption (SAHE) scheme with a decentralized architecture for electronic health records. Unlike conventional Attribute-Based Encryption (ABE) or Ciphertext-Policy ABE (CP-ABE), which rely on central authorities and suffer from key management complexities, SAHE enables multi-level encryption derived from semantic data classification without introducing a single point of failure. This approach not only ensures fine-grained access control aligned with

healthcare roles but also preserves data confidentiality through hierarchical AES-GCM encryption and HMAC-based key derivation. Furthermore, by combining SAHE with blockchain and IPFS, this research presents an end-to-end, patient-centric EHR model that strikes a balance between security, privacy, and usability. To the best of our knowledge, this is the first implementation of semantic-aware hierarchical encryption in a fully decentralized medical data system.

2. RESEARCH METHOD

2.1 Related Method

Electronic Health Record (EHR) security utilises various encryption and decentralised storage methods. Hierarchical encryption methods utilize tiered structures for precise access control. Hierarchical Ciphertext-Policy Attribute-Based Encryption (CP-ABE) provides flexible cloud storage access [11], while Cross-Domain Secure CP-ABE enables inter-domain sharing with DBDH security [12]. Fine-grained User Revocation with Traceability (FUR-HABE) shifts computation to cloud servers using KEK mechanisms [11]. Hierarchical Identity-Based Encryption (HIBE) generates keys from hierarchical identities [13], though facing efficiency challenges [14]. Advanced variants include Anonymous HIBE with Delegated Traceability [15], Anonymous HIBE with blinding techniques [16], Key-Policy Hierarchical-Authority ABE with q-EBDHE security [17], and Revocable HIBE schemes with adaptive security [18].

Semantic-based encryption integrates semantic technology for enhanced control. Outsourced ABE combines CP-ABE with semantic mapping for fog node outsourcing (He & He, 2023). Physical Layer Semantic Encryption uses BLEU scores for semantic keys in DLSC [19]. Fuzzy Semantic Searchable Encryption enables multi-keyword search using fingerprints, WordNet expansion, and modified FHEI [20].

Decentralised storage distributes data across multiple nodes, eliminating single points of failure [21], [22], [23]. Trust Decentralized Loop Federated Learning Blockchain combines blockchain with FL for IoMT privacy [24]. Multi-authority PKI systems use IPFS with binary tree structures [25]. Permissioned blockchain implementations utilize IBFT consensus with IPFS, storing data off-chain while maintaining on-chain hashes with smart contracts [26], [27]. Ethereum platforms provide three-layer architectures with smart contract CRUD management [28]. Personal Health Record systems combine Ethereum with IPFS for patient control [29]. Advanced implementations use redactable blockchain with revocable IPFS and chameleon hash functions [30].

Theoretical foundations include Golang for concurrent programming [31], blockchain for decentralized records [32], IPFS for distributed storage [33], AES for symmetric encryption [34], [35], [36], [37], [38], asymmetric encryption [39], Key Management Systems for lifecycle management [40], [41], semantics for structured access [6], [42], and cryptanalysis for security evaluation including linear and differential analysis [43], [44], with OWASP guidelines [43].

Compared to SABE and CP-ABE schemes, the proposed SAHE approach reduces dependency on central authorities. It improves performance by using symmetric key operations (AES-GCM) and lightweight key derivation (HKDF). Unlike SABE, which requires semantic inference engines and ontology processing, SAHE uses direct semantic sensitivity mapping for hierarchical encryption, making it more practical for resource-constrained healthcare settings.

2.2 Data Storage Process and Key Management

In this process, data encryption is performed using the Semantic-Aware Hierarchical Encryption (SAHE) method. The first step in this encryption process is creating keys for each input data. After the key is determined, the next step is encrypting the data using SAHE. The encrypted data, in the form of files, will then be stored off-chain on the InterPlanetary File System (IPFS). Off-chain data storage in IPFS enables data to be stored in a decentralised manner, thereby enhancing security and resistance to data manipulation.

In this process, medical numbers will also be generated, which will later serve as identifiers. Then, data that is not in file form will be entered into the blockchain. All activities performed will be recorded and entered into the blockchain to ensure transparency and auditability. The results of key formation at the beginning of the process will be used to perform re-encryption using the Key Management System (KMS) using symmetric encryption methods, thus producing encrypted keys. These encrypted keys will also be stored in the blockchain along with other data. The use of KMS ensures secure and efficient key management, supporting scalability and security in encrypted data storage. All storage and encryption processes are illustrated in Figure 1.

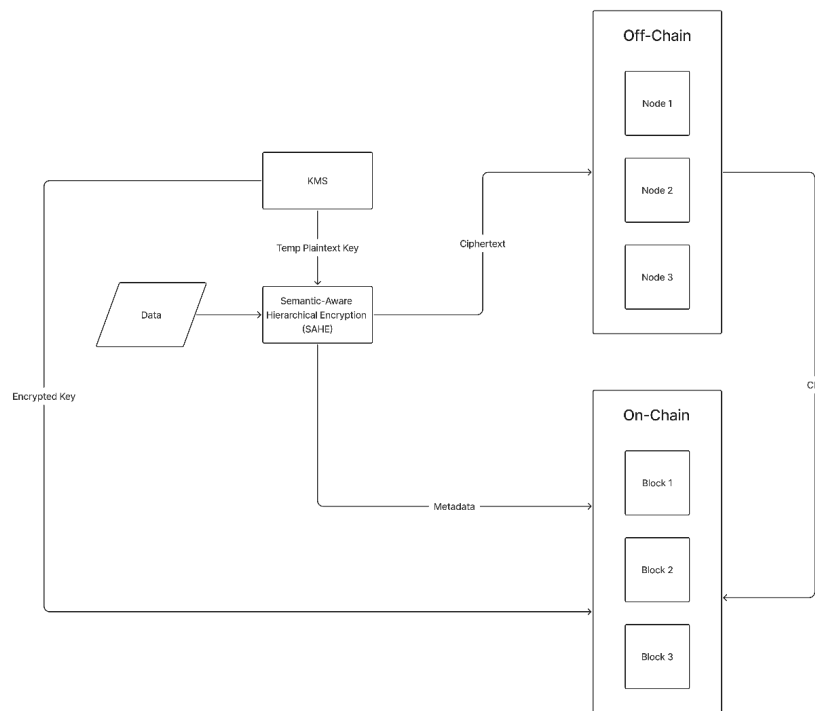


Figure 1. Data Storage Process and Key Management

2.3 Data Request Process

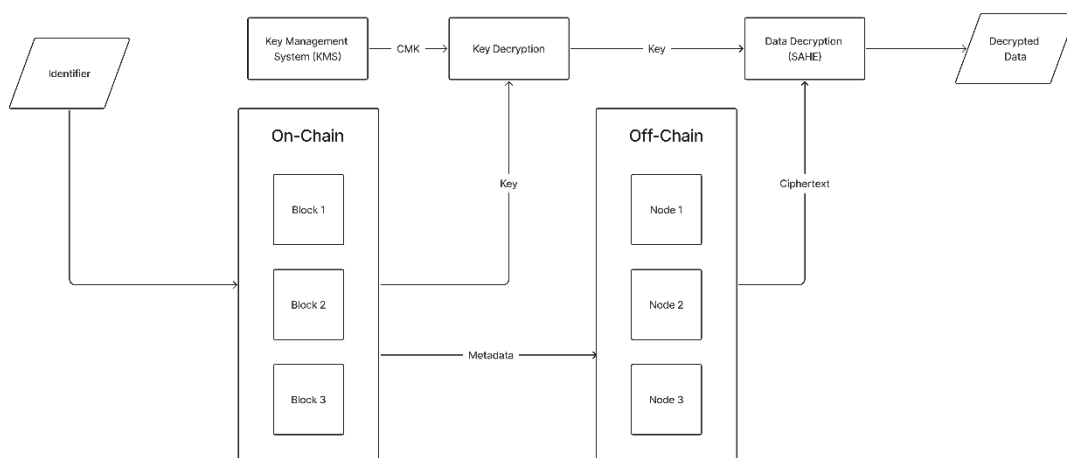


Figure 2. Data Request Process

During the data request process, the system receives input in the form of identifiers that will be used to search for medical record data, along with re-encrypted keys stored in the blockchain. Besides receiving input in the form of identifiers, the system will also obtain tokens that store information about data accessors. After data is obtained from the blockchain, the next step is to decrypt the key first using the key stored in the Key Management System (KMS).

After the key is successfully decrypted, the next step is to decrypt the data using Semantic-Aware Hierarchical Encryption (SAHE). Using SAHE, the system can regulate data access based on the role possessed by the user, which is obtained along with the token. After the data decryption process is complete, the decrypted data will be returned to the user. The returned data is adjusted to the access rights possessed by that user, ensuring that users can only access data that is permitted according to their roles. All data request processes are illustrated in Figure 2.

2.4 Semantic-Aware Hierarchical Encryption

The method applied in this research is known as Semantic-Aware Hierarchical Encryption (SAHE), which aims to enhance security and access control over electronic medical data. This approach begins by building a hierarchical data structure that describes the sensitivity level of each data. Each data is grouped based on its sensitivity level, forming a hierarchy where the root node represents data with the lowest sensitivity, while subsequent nodes indicate higher sensitivity levels. This structure enables data access to be adjusted according to its sensitivity level, creating more flexible and organised access arrangements.

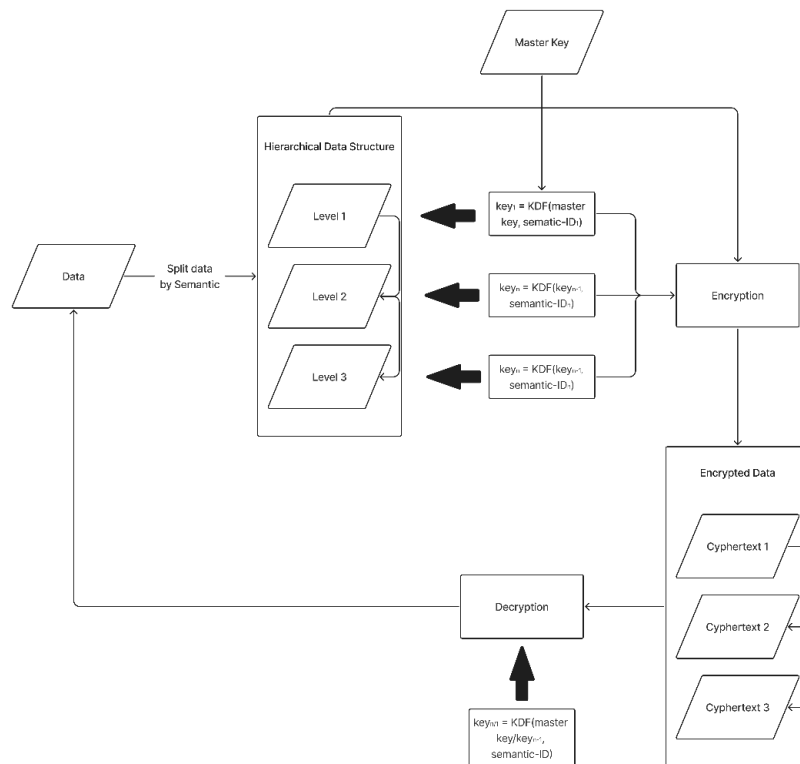


Figure 3. Semantic-Aware Hierarchical Encryption

After the hierarchical structure is established, the next step is to create a 256-bit master key. This key becomes the foundation for the derived key derivation process that will be used at each level of the data hierarchy. By utilising derivation functions based on HMAC (HMAC-based Key Derivation Function or HKDF) and SHA-256 algorithms, each derived key produced has unique characteristics derived from the master key and the semantic ID of the associated data. This approach ensures that each derived key can only be used at the appropriate sensitivity level, maintaining data security at each layer of the hierarchy.

The encryption process is performed using the Advanced Encryption Standard algorithm in Galois/Counter Mode (AES-GCM). This algorithm is chosen because of its ability to not only encrypt data but also generate authentication tags that ensure data authenticity and integrity. Each piece of data in the hierarchy is encrypted with keys determined according to its sensitivity level, ensuring that only users with access to the specific keys can decrypt that data.

In the final stage, the decryption process is performed by reversing the encryption procedure. The system will verify authentication tags to ensure that the accessed data has not been modified. If authentication succeeds, then plaintext can be reproduced from ciphertext. Conversely, if authentication fails, access to the data will not be granted. This step adds an important security layer, as only data that has not been modified and maintains its integrity can be accessed.

Overall, the SAHE method yields an encryption system that is sensitive to data sensitivity levels and offers more stringent access control. With key arrangements structured according to a hierarchy and the use of authentication tags, this approach can maintain the confidentiality and authenticity of electronic medical data, making it more secure from potential threats and manipulation. The scheme of this method is shown in Figure 3.

2.5 Creating Hierarchical Data

In this process, a hierarchical data structure will be built based on the semantics of input data. Before building the hierarchical structure, it is necessary to provide semantics for each data. This semantics also represents the data sensitivity level, as the hierarchical structure arranges data based on its sensitivity. The root node is considered as data with the lowest sensitivity level. Furthermore, this root node will have child nodes with higher sensitivity levels compared to previous data. This child node can continue to develop downward according to the number of determined semantics. Thus, each node has a parent node, except for the first data, which has no parent node (the root node).

2.6 Creating Keys

At this stage, a master key will be created that will later be used to generate keys for each hierarchical data structure formed in the previous process. This master key will utilise a key size of 256 bits, ensuring compatibility with the AES-256 encryption algorithm. The master key formation process is performed using 256-bit random data and follows Equation (1).

$$K_{\text{master}} \leftarrow \{0,1\}^{256} \quad (1)$$

After the master key formation process is complete, the next step is to perform derived key derivation using the HMAC-based Key Derivation Function (HKDF) with SHA-256 as the hash function. The derived key derivation process K_{derived} is generated from master key K_{master} and semantic ID SID according to Equation (2). HKDF ensures that each derived key is unique and secure, taking into account the varying data sensitivity within the hierarchical structure. Thus, each hierarchical data will have specific keys, which increases the overall system security level.

$$K_{\text{derived}} = \text{HKDF}_{\text{SHA}-256}(K_{\text{parent}}, \text{SID}, \text{info} = \emptyset) \quad (2)$$

where:

K_{parent} = parent key (e.g., master key or previous derived key)

SID = semantic ID

info = $\emptyset$ = no additional information used besides semantic ID

2.7 Encryption

After hierarchical data formation and key creation are complete, the next stage is the encryption process. This encryption process is performed using symmetric encryption methods, where the key used is derived from the key formed in the previous process. At this stage, encryption is performed using the Advanced Encryption Standard (AES) algorithm in

Galois/Counter Mode (GCM). The encryption process of ciphertext C from plaintext P with key K and nonce N can be written as in Equation (3).

$$C = AES - GCM(K, N, P) \quad (3)$$

where:

K = encryption key (32 bytes for AES-256)

N = unique and unpredictable nonce, generated from semantic ID

P = plaintext to be encrypted

C = ciphertext to be produced

AES-GCM not only encrypts data but also generates authentication tags to ensure data integrity and authenticity. Thus, ciphertext C includes encrypted data and authentication tags.

2.8 Decryption

At this stage, data decryption will be performed based on the ciphertext generated from the previous data. This decryption process uses the AES-GCM algorithm. Decryption is performed by reversing the encryption process. Ciphertext C is decrypted using key K and nonce N to obtain plaintext P again, as shown in Equation (4).

$$P = AES - GCM^{(-1)}(K, N, C) \quad (4)$$

If authentication tag validation fails during the decryption process, the operation will fail, and plaintext will not be produced, ensuring that manipulated data cannot be accessed. This process ensures that only unmodified and authentic data can be accessed, maintaining data integrity and security in the system.

3. RESULTS AND DISCUSSION

3.1. Data Description

The data used in this research is EHR (Electronic Health Record) data. This data is presented in JSON format, which allows for an organized and easily accessible data structure. The EHR data structure, presented in JSON format, is detailed in Table 1, which outlines the various elements and attributes included in each electronic health record. The use of the JSON format facilitates data interoperability and integration, simplifying further data processing and analysis. Additionally, JSON format supports flexibility in adding or modifying attributes according to research needs, ensuring that the data used remains relevant and comprehensive.

The dataset used represents clean, structured data in JSON format, which facilitates encryption and access control. However, real-world medical data often contain inconsistencies, missing fields, and legacy formats. Future versions of the system should include data validation, normalization, and reconciliation pipelines to support such cases.

Table 1. Data Input

Key	Description	Semantics
patient_name	Patient's full name	Identification
date_of_birth	Patient's date of birth	Basic Information
age	Patient's age	Basic Information
education	Patient's last education	Basic Information
religion	Patient's religion	Basic Information
chief_complaint	Patient's main complaint that is the reason for medical visit	Medical History
current_medical_history	Information about current symptoms or medical complaints, including description, frequency of last urination, as well as fever, nausea, and vomiting status	Medical History
mental_status	Patient's mental or emotional condition during examination	Examination
social_cultural_background	Patient's social and cultural information	Basic Information

occupation	Patient's occupation	Basic Information
past_medical_history	History of diseases previously experienced by the patient	Medical History
allergy_history	Patient's allergy history	Medical History
physical_examination	Patient's physical examination results, including consciousness, blood pressure, pulse, temperature, respiratory rate, oxygen saturation, pain scale, weight and height, as well as general condition of specific body parts	Vital Signs
local_status	Local status that includes several special examinations such as abdominal abnormalities, heart condition, and extremities	Examination
additional_examinations	Additional or supporting examinations	Examination
diagnosis	Temporary medical diagnosis given by the doctor	Diagnosis
treatment_plan	Treatment plan that includes types of intravenous fluids, pain relievers, and gastric protective medications	Treatment Plan
examination_date	Date and time when medical examination was conducted	Examination
examining_doctor	Name of the doctor who conducted the examination	Examination

3.2. Application Performance

Testing was conducted using K6 scripts, which consisted of a total of four stages or scenarios. The first stage of testing was conducted with up to 10 users, followed by the second stage with up to 50 users, the third stage with 200 users, and the final stage involved a cooldown phase by reducing the number of users to 0. The total testing time performed for the four stages was 7 minutes. The first stage lasted 1 minute, the second stage was 2 minutes, the third stage was 3 minutes, and the final stage was 1 minute.

Figure 4 shows the CPU usage graph of the application throughout the testing duration. CPU usage tends to fluctuate, with high peaks, when heavy workloads are applied. At the beginning of testing (around 19:04), the CPU had already reached usage above 100%, indicating that the application utilised more than one processor core in parallel. Throughout testing, there were CPU usage spikes that reached a peak of 141% (occurring at ~19:09:16). Values above 100% indicate multi-core utilisation; for example, 141% means that about 1.41 CPU cores were used cumulatively at that time. Outside peak periods, CPU usage momentarily dropped back to lower levels (often below 50%), reflecting idle time or light loads between intensive tasks. Overall, the average CPU usage during testing was ~55% (marked by the grey dashed line on the graph), indicating moderate processor utilisation levels during test scenarios involving repeated loads.

Figure 5 illustrates memory (RAM) usage trends by the application during the same testing. The emerging pattern is one of monotonically increasing memory: it increases gradually over time and rarely decreases. At the beginning of testing, memory usage was recorded at around 58.50 MB (showing initial application condition before many loads). Each time the system receives an intensive workload, there is a gradual increase in memory usage. This increase is displayed as ascending steps on the graph — for example, the first load spike raised memory to ~70 MB, then remained stable momentarily, and then rose again with subsequent loads. Peak memory usage was reached near the end of testing at approximately 96.75 MB. After reaching that peak (around 19:11), the graph flattened, indicating that memory stagnated at that high level and was not released back to its original levels. This indicates the application may be caching or retaining data in memory after heavy operations are complete, so memory usage does not drop significantly again. Nevertheless, total memory usage is relatively small (below 100 MB) and only ~0.6% of system capacity, so it does not critically burden RAM resources.

The CPU averaged ~55.4%, with a peak of 141% and a low of 9% when the system was nearly idle. Memory averaged ~80.2 MB, with a minimum of 58.5 MB at the beginning and a maximum of 96.75 MB near the end of testing. This value range is consistent with visual observations on graphs: the CPU shows high variability (large fluctuations between 9% and 141%), while memory rises from ~58 MB and finally stabilises near 96-97 MB. No significant decrease in memory was detected throughout testing, so minimum memory values only appeared at the beginning of the experiments. Conversely, for CPU, minimum values occurred several times when the system was between loads (around 9-10% during idle phases). High variation in CPU vs.

gradual increase in memory indicates that application load is more CPU-intensive (burdening computational processes) while memory allocation increases with tasks without being freed midway.

Although the prototype demonstrates stable behaviour under a 200-user load, healthcare networks at a regional or national scale may experience significantly higher concurrency during peak periods (e.g., pandemics). Future studies should scale testing scenarios accordingly to reflect realistic traffic patterns.

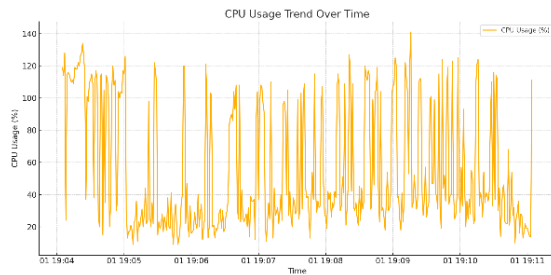


Figure 4. CPU Usage Trend

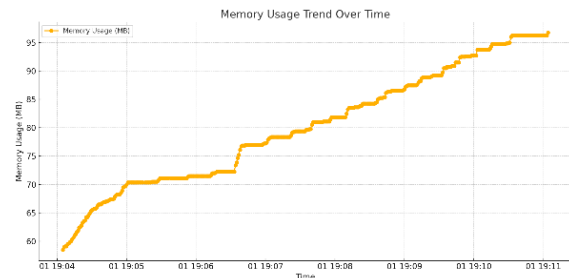


Figure 5. Memory Usage Trend

3.3. OWASP

Security testing of SAHE (Decentralized Electronic Medical Record System) using OWASP ZAP did not identify critical weaknesses in most OWASP categories. Here is a summary of testing results for queried categories based on Figure 6:

Alert counts by alert type

This table shows the number of alerts of each alert type, together with the alert type's risk level.

(The percentages in brackets represent each count as a percentage, rounded to one decimal place, of the total number of alerts included in this report.)

Alert type	Risk	Count
Content Security Policy (CSP) Header Not Set	Medium	2 (40.0%)
Application Error Disclosure	Low	1 (20.0%)
Information Disclosure - Debug Error Messages	Low	1 (20.0%)
X-Content-Type-Options Header Missing	Low	1 (20.0%)
Re-examine Cache-control Directives	Informational	1 (20.0%)
Total		5

Figure 6. OWASP Zap Result

- 1) Injection (SQL/Command Injection): No SQL injection, system command, or other code injection vulnerabilities were detected. This means that user input on APIs and applications is handled well, so they are not vulnerable to injection.
- 2) Broken Authentication: No findings regarding authentication weaknesses. ZAP found no issues, such as default or weak credentials, easily hijacked sessions, or flawed login mechanisms.

- 3) Sensitive Data Exposure: No direct sensitive data leaks were found (e.g., personal or medical information did not appear carelessly). However, there are minor indications related to information exposure through error messages and browser cache (see the table below), which should be addressed to prevent potential exploitation.
- 4) Broken Access Control: No vulnerabilities related to access control were found. ZAP did not report unauthorized access or authorization weaknesses in this system.
- 5) Insecure Deserialization: No indications of Insecure Deserialization vulnerabilities were found. Insecure Deserialization vulnerabilities specifically occur when untrusted data is allowed to undergo deserialization processes without adequate validation, enabling attackers to execute arbitrary code or damage system integrity. From the code side, for data deserialization, it already has data contracts, so if the data does not conform, errors will occur.

3.4. Different Cryptanalysis

Differential testing was performed by modifying 1 bit in plaintext ("John Doe" becomes "Jo`n Doe") and observing ciphertext differences (ΔC). Results show no $\Delta P \rightarrow \Delta C$ pairs consistently have probabilities above 6.25% (threshold based on 1/16 distribution). In initial tests with limited samples, DP Probability percentages appeared high (e.g., 50% for two samples, 33.33% for three samples, up to ~6.67% for 15 samples). However, this was solely due to small sample sizes. After 32 tests, each emerging ΔC only occurred once, so the maximum observed probability for each specific $\Delta P \rightarrow \Delta C$ dropped to ~3.12% (1/32). The ~3.12% value is below the 6.25% threshold, indicating that the ΔC distribution approaches uniformity. In other words, no $\Delta P \rightarrow \Delta C$ patterns were found that prominently exceed what would be expected randomly. Test results are in Table 2.

Table 2. Cryptanalysis Result

Test	DP Probability (%)	Hamming Distance
1	100	144
2	50	141
3	33.33	145
4	25	148
5	20	139
6	16.67	149
7	14.29	136
8	12.5	147
9	11.11	145
10	10	154
11	9.09	156
12	8.33	141
13	7.69	131
14	7.14	125
15	6.67	140
16	6.25	144
17	5.88	129
18	5.56	145
19	5.26	149
20	5	160
21	4.76	149
22	4.55	131
23	4.35	152
24	5.17	131
25	4	137
26	3.85	157
27	3.7	133
28	3.57	138
29	3.45	150
30	3.33	139
31	3.23	141
32	3.12	147

Table 2 indicates that the SAHE algorithm demonstrates a strong avalanche effect. The avalanche effect is a desirable property in cryptographic algorithms, where small changes in input (e.g., one bit in the plaintext or key) result in significant changes in the ciphertext output. Ideally, the Strict Avalanche Criterion (SAC) is fulfilled when a 1-bit change in input causes each output bit to have a 50% chance of changing.

$$AE(\%) = \left(\frac{HD}{n} \right) \times 100 \quad (5)$$

where:

$AE(\%)$ = Avalanche Effect in percent

HD = Hamming Distance between ciphertext C and modified ciphertext C'

n = total number of output bits (288 bits)

In this SAHE testing, using Equation 5, an average of ~50% of the output bits change for a single input bit change, thus fulfilling the avalanche criterion. This is consistent with the literature, which states that algorithms are considered secure when the avalanche effect reaches ~50% output bit changes [45]. In other words, SAHE successfully produces sound diffusion: changed plaintext bits affect ciphertext drastically and randomly, which is a characteristic of high-quality cyphers.

High probabilities for specific $\Delta P \rightarrow \Delta C$ pairs generally indicate potential weaknesses, where cyphers that produce specific ΔC predictably increase vulnerability to differential attacks. In this test data, no differentials with high probabilities persisted. The probability of ΔP producing a specific ΔC decreases with additional tests and eventually reaches random levels. This means that 1-bit changes in plaintext spread in an avalanche fashion to various ciphertext differences without any dominant patterns. The absence of prominent $\Delta P \rightarrow \Delta C$ pairs indicates that SAHE does not exhibit apparent differential weaknesses in this testing. In other words, the SAHE cypher successfully avoids differential bias that attackers can exploit.

3.5. Data Access

Table 3. Data Access Result

Doctor/Patient	
	{
	"data": {
	"id": "e2c8c9f0-bbfl-4be6-a380-4c0fa4b8a8a6",
	"patient_id": "3500000000000000",
	"patient_name": "Ibu Rani",
	"date_of_birth": "1979-03-10",
	"age": 46,
	"education": "S1",
	"agama": "Islam",
	"anamnesis": {
	"chief_complaint": "Abdominal enlargement",
	"current_medical_history": "Patient complains of stomach pain, frequent nausea, vomiting, and cramping.",
	"mental_status": "Alert and oriented",
	"social_cultural_background": "",
	"occupation": "PNS",
	"allergy_history": "None"
	},
	"physical_examination": {
	"consciousness": {
	"eye_opening": 4,
	"best_verbal": 5,
	"best_motor": 6
	},
	"blood_pressure": "110/70 mmHg",
	"pulse_rate": 84,
	"temperature": 36.5,
	"respiratory_rate": 20,
	"spo2": 98,
	"weight": 56,
	"height": 157,
	"eyes": "Normal",

```

        "ears": "Normal",
        "nose": "Normal",
        "mouth": "Normal",
        "neck": "Normal",
        "chest": "Normal",
        "abdomen": "Epigastric pain",
        "extremities": "Normal",
        "skin": "Normal",
        "other_findings": "-"
    },
    "treatment_plan": "Administer antacids and observe for gastritis",
    "examination_date": "2025-03-19",
    "examination_type": "Emergency"
}

Nurse {
    "data": {
        "id": "e2c8c9f0-bbf1-4be6-a380-4c0fa4b8a8a6",
        "patient_id": "3500000000000000",
        "patient_name": "Ibu Rani",
        "date_of_birth": "1979-03-10",
        "age": 46,
        "education": "S1",
        "agama": "Islam",
        "anamnesis": {
            "chief_complaint": "Abdominal enlargement",
            "current_medical_history": "Patient complains of stomach pain, frequent nausea, vomiting, and cramping.",
            "mental_status": "Alert and oriented",
            "social_cultural_background": "",
            "occupation": "PNS",
            "allergy_history": "None"
        },
        "physical_examination": {
            "consciousness": {
                "eye_opening": 4,
                "best_verbal": 5,
                "best_motor": 6
            },
            "blood_pressure": "110/70 mmHg",
            "pulse_rate": 84,
            "temperature": 36.5,
            "respiratory_rate": 20,
            "spo2": 98,
            "weight": 56,
            "height": 157,
            "eyes": "Normal",
            "ears": "Normal",
            "nose": "Normal",
            "mouth": "Normal",
            "neck": "Normal",
            "chest": "Normal",
            "abdomen": "Epigastric pain",
            "extremities": "Normal",
            "skin": "Normal",
            "other_findings": "-"
        },
        "treatment_plan": null
    }
}

Administrator {
    "data": {
        "id": "e2c8c9f0-bbf1-4be6-a380-4c0fa4b8a8a6",
        "patient_id": "3500000000000000",
        "patient_name": "Ibu Rani",
        "date_of_birth": "1979-03-10",
        "age": 46,
        "education": "S1",
        "agama": "Islam",
        "anamnesis": {
            "social_cultural_background": null,
            "allergy_history": null
        },
        "physical_examination": {

```

```

        "consciousness": {}
    },
    "treatment_plan": null
}

```

In this subsection, the implementation of electronic medical record access control based on Semantic-Aware Hierarchical Encryption (SAHE) is evaluated by comparing the data obtained by each user role (doctor, nurse, administrator), as shown in Table 3. Testing was conducted by submitting medical record data requests using different credentials corresponding to various roles. Test results indicate that each role receives different data outputs based on their access rights and information needs. Differences in data coverage received by each role are summarized as follows:

- 1) Doctor/Patient: Gets the most complete access to patient medical records. All information is displayed intact, including patient identity and demographics, detailed anamnesis (chief complaints, current medical history, mental status, socio-cultural background, occupation, allergy history), complete physical examination results (example: consciousness level with Glasgow Coma Scale score, vital signs such as blood pressure, pulse, temperature, respiratory rate, oxygen saturation, as well as examination findings on eyes, ears, nose, mouth, neck, chest, abdomen, extremities, skin, and other findings), treatment plans, and examination metadata such as examination date and service type (example: Emergency). This comprehensive access is appropriate for doctors' roles who need complete information for patient diagnosis and therapy determination.
- 2) Nurse: Gets limited access to clinical data relevant to nursing tasks, without including information that is final clinical decisions by doctors. In test results, nurses receive patient identity and demographic data, as well as almost all anamnesis details and physical examination findings, which are the same as those received by doctors (e.g., complaints, history, vital signs, and other basic examination findings are displayed). The main difference is that treatment plan columns are not displayed (null or empty values), so nurses do not see specific therapy instructions made by doctors. Additionally, certain metadata information, such as examination dates and types, is also not included in nurse displays. This access is considered adequate for nurses who need to understand patient conditions (complaints and physical status) to carry out care but do not require details on therapy decisions that fall within doctors' domains.
- 3) Administrator: Gets minimal access, limited to non-medical or patient demographic data only. From testing output, administrators can only view basic patient information, including medical record ID, patient identification number, name, date of birth, age, education, and religion. There is no detailed anamnesis or physical examination result information provided -- anamnesis columns other than social data (such as social_cultural_background and allergy_history) appear with null values and physical examination sections only display basic structure without content (e.g., empty consciousness objects without scores). The treatment plan columns are also empty. This indicates that admins do not have the keys to decrypt sensitive medical data. This minimal access is suitable for administrative tasks that are administrative (e.g., registration or file management) and does not require access to patient medical information.

The above differences reflect the relevance of access rights to the professional needs of each role. Doctors, as the primary responsible parties for patient care, require complete information to make accurate diagnoses, determine effective action plans, and monitor patient conditions holistically; therefore, the system provides comprehensive access to all medical record data. Nurses require important clinical information to care for patients (e.g., patient complaints, current conditions, vital signs, and general care instructions), so they are given access to patient anamnesis

and physical examination results. However, nurses do not make final treatment decisions, so details like complete therapy plans from doctors are considered outside the daily nursing task needs. Such information is sufficient to be conveyed operationally through separate instructions. Meanwhile, administrators only require patient data that is administrative (demographics and identity) for system or medical record management purposes, without involvement in medical aspects; therefore, admin access is limited to non-medical and general information, thereby avoiding exposure to sensitive clinical data unrelated to administrative tasks.

From a security perspective, this implementation has appropriately applied the least privilege principle. The least privilege principle means that each user is only given the minimum access rights necessary to perform their tasks. Test results demonstrate this principle: not one role obtained more data than needed. Doctors receive all data because their role demands it, but nurses only obtain data subsets essential for their tasks, and administrators are more limited to general data only. Thus, risks of misuse or unauthorized access to sensitive data can be reduced. For example, patient treatment plan information that should only be accessible to doctors remains protected from access by nurses and administrators. Similarly, patient medical details (complaints, history, and examination findings) are not exposed to unauthorized administrators. Each role can only view information within its respective scope of responsibility.

Semantic Aware Hierarchical Encryption (SAHE) implementation proves effective in ensuring the above scheme works. SAHE encapsulates medical record data into layers encrypted according to sensitivity categories or roles allowed to access. Decryption keys are distributed hierarchically: roles with higher authority (e.g., doctors) have keys to access all data layers, while lower-level roles (e.g., nurses and administrators) only have keys for specific layers. Results are visible in the testing output: fields not permitted for specific roles remain encrypted, so they appear empty or null on that user's side. For example, nurses do not have the keys to decrypt the treatment plan, so that field appears null, although doctors can view its contents. Similarly, admins do not have access to keys for anamnesis and examination data, so those sections in admin output are either left empty or their contents are removed. The fact that data appears cut or empty precisely according to established limitations shows that semantic-hierarchical encryption mechanisms work as expected. Only users with appropriate keys and authority can read specific data components. This mechanism effectively prevents improper access and proper escalation; for example, even though admins, as technical system managers, can access medical record files, without proper semantic keys, admins still cannot display medical information within them. Thus, SAHE access control has been successfully implemented, ensuring strict separation of data access between roles. This multi-role data access testing confirms that the built decentralized electronic medical record system has accommodated different stakeholder needs while ensuring information security principles (confidentiality) through layered encryption.

4. CONCLUSION

This research successfully developed and implemented a decentralised electronic medical record platform utilising Semantic-Aware Hierarchical Encryption (SAHE), achieving significant improvements in both system performance and patient data protection.

The implemented SAHE mechanism effectively enforces role-based access control, ensuring that medical personnel access patient information strictly by their authorised levels. Through comprehensive multi-role testing, the system confirmed that doctors can view complete patient data, nurses access only relevant medical information without restricted details, and administrators are limited to non-medical demographic data. No unauthorized access violations were detected across different user hierarchies, validating the successful implementation of need-to-know and least-privilege principles.

System performance evaluation reveals efficient resource utilisation, with an average CPU usage of 55% and memory consumption of around 80 MB during normal operations. Even under peak loads, the system maintains stability, with CPU usage reaching 141% at multi-core peaks, while memory usage remains below 100 MB (approximately 0.6% of the system's RAM capacity). These performance metrics demonstrate that encryption and decryption overhead remain within acceptable limits, making the system viable for large-scale deployment.

The decentralised architecture, combined with SAHE, provides robust protection for sensitive patient data, including diagnoses, examination results, and treatment plans. The encryption-based access control ensures that confidential health information remains protected even in the event of node compromise, as data remains unreadable without appropriate decryption keys. This approach aligns with medical record confidentiality regulations while maintaining convenient access for authorized personnel.

Despite these positive outcomes, further evaluation is necessary to assess system scalability and performance under real-world conditions. Future testing should involve larger datasets, increased user concurrency, and extended operational scenarios to identify potential bottlenecks in the SAHE implementation and decentralized infrastructure. Additionally, addressing IPFS operational costs through private network implementations could enhance the system's economic feasibility for large-scale healthcare deployment. Operational cost analysis for IPFS storage, including bandwidth and replication overhead, remains unquantified. Especially for resource-constrained hospitals, private IPFS clusters or on-premise decentralized storage nodes may be considered to reduce recurring costs.

Since decentralised systems rely on reliable peer-to-peer communication, inadequate internet infrastructure in remote regions of Indonesia may hinder real-time data access. Thus, incorporating asynchronous synchronization, caching, or fallback to local nodes may be necessary to maintain usability.

While the proposed system demonstrates strong security and performance in controlled scenarios, this prototype has not yet been validated in production healthcare environments. Real-world deployments may present additional challenges, such as inconsistent data formats, network delays, integration with legacy hospital information systems, and operational resilience requirements. These practical concerns are beyond the current scope and represent important directions for future work.

The developed system demonstrates that decentralised electronic medical records with semantic-aware hierarchical encryption can effectively balance patient privacy protection with authorised access convenience, providing a foundation for secure and scalable healthcare data management solutions.

REFERENCES

- [1] R. Sendelj and I. Ognjanovic, "Cybersecurity Challenges in Healthcare," in *Studies in Health Technology and Informatics*, IOS Press BV, Oct. 2022, pp. 190–202. doi: 10.3233/SHTI220951.
- [2] I. C. A. Pilares, S. Azam, S. Akbulut, M. Jonkman, and B. Shanmugam, "Addressing the Challenges of Electronic Health Records Using Blockchain and IPFS," *Sensors*, vol. 22, no. 11, Jun. 2022, doi: 10.3390/s22114032.
- [3] Gaganjeet Singh Reen, Manasi Mohandas, and S. Venkatesan, "Decentralized Patient Centric e- Health Record Management System using Blockchain and IPFS," *IEEE*, 2019. doi: 10.1109/CICT48419.2019.9066212.
- [4] X. Qu, Z. Yang, Z. Chen, and G. Sun, "A consent-aware electronic medical records sharing method based on blockchain," *Comput Stand Interfaces*, vol. 92, Mar. 2025, doi: 10.1016/j.csi.2024.103902.
- [5] R. P. Puneeth and G. Parthasarathy, "A cross-chain-based approach for secure data sharing and interoperability in electronic health records using blockchain technology," *Computers and Electrical Engineering*, vol. 120, Dec. 2024, doi: 10.1016/j.compeleceng.2024.109676.
- [6] H. Arshad, C. Johansen, O. Owe, P. Picazo-Sanchez, and G. Schneider, "Semantic Attribute-Based Encryption: A framework for combining ABE schemes with semantic technologies," *Inf Sci (N Y)*, vol. 616, pp. 558–576, Nov. 2022, doi: 10.1016/j.ins.2022.10.132.
- [7] S. Ding, C. Li, and H. Li, "A Novel Efficient Pairing-Free CP-ABE Based on Elliptic Curve Cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, May 2018, doi: 10.1109/ACCESS.2018.2836350.

- [8] Y. F. Yao, H. Y. Chen, Y. Gao, K. Wang, and H. Y. Yu, "A decentralized multi-authority CP-ABE scheme from LWE," *Journal of Information Security and Applications*, vol. 82, May 2024, doi: 10.1016/j.jisa.2024.103752.
- [9] S. Liu, J. Yu, L. Chen, and B. Chai, "Blockchain-Assisted Comprehensive Key Management in CP-ABE for Cloud-Stored Data," *IEEE Transactions on Network and Service Management*, vol. 20, no. 2, pp. 1745–1758, Jun. 2023, doi: 10.1109/TNSM.2022.3185237.
- [10] M. Xiao, H. Li, Q. Huang, S. Yu, and W. Susilo, "Attribute-Based Hierarchical Access Control With Extendable Policy," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1868–1883, 2022, doi: 10.1109/TIFS.2022.3173412.
- [11] X. Yang and Y. Tao, "FUR-HABE: A Hierarchical CP-ABE Scheme with Traceable Fine-grained User Revocation for Cloud Storage," May 22, 2024. doi: 10.21203/rs.3.rs-4410803/v1.
- [12] G. Yang, Peixuan Li, Y. Xin, Y. He, C. Wang, and X. Chen, "An efficient hierarchical attribute-based encryption scheme with cross-domain data sharing," *Computer Networks*, vol. 255, Dec. 2024, doi: 10.1016/j.comnet.2024.110863.
- [13] N. A. Van Nguyen and M. T. T. Pham, "Quantum-safe Anonymous Hierarchical Identity-Based Encryption with Traceable Identities," *Comput Stand Interfaces*, vol. 84, Mar. 2023, doi: 10.1016/j.csi.2022.103695.
- [14] K. Emura, A. Takayasu, and Y. Watanabe, "Efficient identity-based encryption with Hierarchical key-insulation from HIBE," *Des Codes Cryptogr*, vol. 89, no. 10, pp. 2397–2431, Oct. 2021, doi: 10.1007/s10623-021-00926-z.
- [15] Z. Y. Liu, M. Mambo, R. Tso, and Y. F. Tseng, "Anonymous hierarchical identity-based encryption with delegated traceability for cloud-based data sharing systems," *Comput Stand Interfaces*, vol. 89, Apr. 2024, doi: 10.1016/j.csi.2023.103817.
- [16] Z. Ying, M. Liu, S. Kong, S. Xu, and J. Xiong, "Anonymous Hierarchical Identity-based Encryption EHR Sharing Mechanism Based on SM9," in *Proceedings - 2022 IEEE 21st International Conference on Ubiquitous Computing and Communications, IUCC-CIT-DSCI-SmartCNS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 17–22. doi: 10.1109/IUCC-CIT-DSCI-SmartCNS57392.2022.00017.
- [17] C. Li *et al.*, "Hierarchical and non-monotonic key-policy attribute-based encryption and its application," *Inf Sci (N Y)*, vol. 611, pp. 591–627, Sep. 2022, doi: 10.1016/j.ins.2022.08.014.
- [18] K. Lee, "Revocable hierarchical identity-based encryption with adaptive security," *Theor Comput Sci*, vol. 880, pp. 37–68, Aug. 2021, doi: 10.1016/j.tcs.2021.05.034.
- [19] Q. Qin *et al.*, "Securing Semantic Communications with Physical-Layer Semantic Encryption and Obfuscation," in *IEEE International Conference on Communications*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 5608–5613. doi: 10.1109/ICC45041.2023.10279807.
- [20] G. Liu, G. Yang, S. Bai, Q. Zhou, and H. Dai, "FSSE: An Effective Fuzzy Semantic Searchable Encryption Scheme over Encrypted Cloud Data," *IEEE Access*, vol. 8, pp. 71893–71906, 2020, doi: 10.1109/ACCESS.2020.2966367.
- [21] M. I. Khalid *et al.*, "A Comprehensive Survey on Blockchain-Based Decentralized Storage Networks," *IEEE Access*, vol. 11, pp. 10995–11015, 2023, doi: 10.1109/ACCESS.2023.3240237.
- [22] I. Vakilinia, W. Wang, and J. Xin, "An Incentive-Compatible Mechanism for Decentralized Storage Network," *IEEE Trans Netw Sci Eng*, vol. 10, no. 4, pp. 2294–2306, Jul. 2023, doi: 10.1109/TNSE.2023.3245326.
- [23] L. Meng and B. Sun, "Research on Decentralized Storage Based on a Blockchain," *Sustainability (Switzerland)*, vol. 14, no. 20, Oct. 2022, doi: 10.3390/su142013060.
- [24] M. Kuliha and S. Verma, "Secure internet of medical things based electronic health records scheme in trust decentralized loop federated learning consensus blockchain," *International Journal of Intelligent Networks*, vol. 5, pp. 161–174, Jan. 2024, doi: 10.1016/j.ijin.2024.03.001.

- [25] A. Kumari, S. Bhaskar, S. Patil, and K. Parmar, "Decentralized and Multi-Authority based Public Key Infrastructure for Sharing Electronic Health Records," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 44–54. doi: 10.1016/j.procs.2023.12.059.
- [26] K. Shuaib, J. Abdella, F. Sallabi, and M. A. Serhani, "Secure decentralized electronic health records sharing system based on blockchains," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5045–5058, Sep. 2022, doi: 10.1016/j.jksuci.2021.05.002.
- [27] S. Ballal, Y. Chandre, R. Pise, B. Sonare, and S. Patil, "Blockchain-Based Decentralized Platform for Electronic Health Records Management," in *2023 IEEE International Conference on Blockchain and Distributed Systems Security, ICBDS 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ICBDS58040.2023.10346392.
- [28] S. Singh, S. Gupta, and Indu, "MedEHR-Electronic health Record using Blockchain," in *2023 International Conference on Computational Intelligence, Communication Technology and Networking, CICTN 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 58–62. doi: 10.1109/CICTN57981.2023.10141053.
- [29] D. Sentausa and D. Habsara Hareva, "Decentralize Application for Storing Personal Health Record using Ethereum Blockchain and Interplanetary File System," in *Proceedings - 2022 1st International Conference on Technology Innovation and Its Applications, ICTIA 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICTIA54654.2022.9935996.
- [30] H. Guo, W. Li, C. Meese, and M. Nejad, "Decentralized Electronic Health Records Management via Redactable Blockchain and Revocable IPFS," in *Proceedings - 2024 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies, CHASE 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 167–171. doi: 10.1109/CHASE60773.2024.00028.
- [31] P. Shukla, S. Thakur, S. Arora, and A. Wadhwa, "Nature-Inspired Algorithms Analysis on various Benchmark Functions using Python and Golang," in *Proceedings of 2022 1st International Conference on Informatics, ICI 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 226–228. doi: 10.1109/ICI53355.2022.9786911.
- [32] B. Gupta, "Understanding Blockchain Technology: How It Works and What It Can Do," *Metaverse Basic and Applied Research*, vol. 1, p. 18, Dec. 2022, doi: 10.56294/mr202218.
- [33] P. Schwentek, S. Zimmermann, C. Von Lengerke, J. A. Cabrera, and F. H. P. Fitzek, "Revisiting MARS: Storing Coded Packets In-Advance for IPFS," in *IEEE International Conference on Communications*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 4692–4697. doi: 10.1109/ICC51166.2024.10622257.
- [34] Baydaa Jaffer Al-Khafaji and Abdul Monem S Rahma, "Proposed new modification of AES algorithm for data security," *Global Journal of Engineering and Technology Advances*, vol. 12, no. 3, pp. 117–122, Sep. 2022, doi: 10.30574/gjeta.2022.12.3.0165.
- [35] William. Stallings, *Cryptography and network security: principles and practice*. Pearson Education Limited, 2017.
- [36] W. K. Lee, H. J. Seo, S. C. Seo, and S. O. Hwang, "Efficient Implementation of AES-CTR and AES-ECB on GPUs With Applications for High-Speed FrodoKEM and Exhaustive Key Search," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 6, pp. 2962–2966, Jun. 2022, doi: 10.1109/TCSII.2022.3164089.
- [37] F. Weissbaum and T. Lugin, "Symmetric cryptography," in *Trends in Data Protection and Encryption Technologies*, Springer Nature, 2023, pp. 7–10. doi: 10.1007/978-3-031-33386-6_2.
- [38] S. Wade, "Description of Image encryption Using AES-256 bits," *Int J Res Appl Sci Eng Technol*, vol. 11, no. 5, pp. 7167–7171, May 2023, doi: 10.22214/ijraset.2023.53365.
- [39] V. Mulder, A. Mermoud, V. Lenders, and B. Tellenbach, *Trends in data protection and encryption technologies*. Springer Nature, 2023. doi: 10.1007/978-3-031-33386-6.
- [40] P. James, S. Laschet, S. Ramacher, and L. Torresetti, "Key Management Systems for Large-Scale Quantum Key Distribution Networks," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Aug. 2023. doi: 10.1145/3600160.3605050.

- [41] S. Rana, F. K. Parast, B. Kelly, Y. Wang, and K. B. Kent, "A comprehensive survey of cryptography key management systems," *Journal of Information Security and Applications*, vol. 78, Nov. 2023, doi: 10.1016/j.jisa.2023.103607.
- [42] P. C. Y. Sheu and A. Ghafoor, "Guest Editorial: Introduction to the Special Issue on Advances in Semantic Computing," Jun. 01, 2015, *IEEE Computer Society*. doi: 10.1109/TETC.2015.2432332.
- [43] Y. Y. Chan, C. Y. Khor, B. T. Khoo, J. Sen Teh, W. J. Teng, and N. Jamil, "On the resistance of new lightweight block ciphers against differential cryptanalysis," *Heliyon*, vol. 9, no. 4, Apr. 2023, doi: 10.1016/j.heliyon.2023.e15257.
- [44] Z. Xu, H. Xu, L. Tan, and W. Qi, "Linear cryptanalysis of SPECK and SPARX," *Journal of Information Security and Applications*, vol. 83, Jun. 2024, doi: 10.1016/j.jisa.2024.103773.
- [45] K. Mohamed, M. N. M. Pauzi, F. H. H. M. Ali, and S. Ariffin, "Analyse On Avalanche Effect In Cryptography Algorithm," in *Proceedings of the International Conference on Sustainable Practices, Development and Urbanisation (IConsPADU 2021)*, 16 November 2021, Universiti Selangor (UNISEL), Malaysia, European Publisher, Oct. 2022, pp. 610–618. doi: 10.15405/epms.2022.10.57.