International Journal of Research in Commerce and Management Studies

ISSN 2582-2292

Vol. 7, No. 02 Mar-Apr; 2025 Page. No. 199-210

To cite this article: Chaitanya Vasamsetty, Sunil Kumar Alavilli, Bhavya Kadiyala, Rajani Priya Nippatla, Subramanyam Boyapati and Purandhar. N (2025). ENSURING SECURE DATA TRANSMISSION AND STORAGE IN CLOUD FOR HEALTHCARE SYSTEMS TO PROTECT PATIENT PRIVACY, International Journal of Research in Commerce and Management Studies (IJRCMS) 7 (2): 199-210 Article No. 358 Sub Id 667

ENSURING SECURE DATA TRANSMISSION AND STORAGE IN CLOUD FOR HEALTHCARE SYSTEMS TO PROTECT PATIENT PRIVACY

Chaitanya Vasamsetty[1], Sunil Kumar Alavilli[2], Bhavya Kadiyala[3], Rajani Priya Nippatla[4], Subramanyam Boyapati[5] and Purandhar. N[6], *

· [1]Elevance Health, Georiga, USA. Email: chaitanyavasamsetty@ieee.org
[2]Sephora, California, USA. Email: sunilkumaralavilli@ieee.org
[3]Parkland Health, Texas, USA. Email : bhavyakadiyala@ieee.org
[4]Kellton Technologies Inc, Texas, USA. Email: rajanipriyanippatla@ieee.org
[5]American Express, Arizona, USA. Email: subramanyamboyapati@ieee.org
[6]Department of CSE (Artificial Intelligence) School of Computers Madanapalle Institute of Technology and Science, Madanapalle
Andhra Pradesh, India. Email: purandhar.n@gmail.com

*Corresponding author name: Purandhar. N. Corresponding author Email: purandhar.n@gmail.com

DOI : https://doi.org/10.38193/IJRCMS.2025.7216

ABSTRACT

More than ever, the healthcare industry witnesses a remarkable transformation with cloud computing becoming a new data-driven era involving healthcare delivery. However, existing methodologies lean heavily on classical encryption algorithms but will alter with emerging and new security threats. This work lies the exhibition of a new paradigm involving NTRU, (N-th degree truncated polynomial ring units), a post-quantum key generation method, and salsa20 for fast data encryption. The entire process involves the generation of keys from the NTRU polynomial-based private-public key pairs and the actual encryption of data using the salsa20 keystream in order for secure data storage inside the cloud environment. The encrypted data is uploaded to the cloud, and decryption is carried out by the authorized receivers using the complete key and the nonce. Results indicated a relationship between the cryptographic processing time, which gradually increased from 2200 ms for a 75-bit key to 3500 ms for a 250-bit key. Further, encryption time went from 2000 ms for a 50-bit key to 16,000 ms for a 300-bit key, while decryption time was observed between 10 ms for a 500-bit key and over 700 ms for a 3500-bit key. Thus, it provides extra strength against quantum attacks yet also balances the performance and makes it viable for future healthcare data protection.

KEYWORDS: Healthcare, Cloud Computing, Salsa20 Stream Cipher, N-th degree truncated polynomial ring units Algorithm, Security, Encryption, Decryption

1. INTRODUCTION

Cloud services have swiftly gained popularity among individuals and enterprises alike, owing to their numerous advantages, such as cost-effectiveness, scalability, and flexibility [1]. Cloud computing, having entered the modern business arena, offers small and medium-sized enterprises (SMEs) enhanced operational efficiencies and strategic capabilities [2]. Hybrid cloud environments are becoming a preferred hosting solution for data-centric operations with the evolution of cloud computing, being synonymous with scalability, flexibility, and cost-saving [3]. These environments facilitate complex data sets' management through access to advanced algorithms and massive computing power [4]. Particularly, the medical landscape has been disrupted by cloud computing adoption and its capabilities, bringing in support for advanced predictive models to cater to growing demands of accuracy and efficiency in medical diagnostics [5]. Added to that, cloud computing also provides safe storage, quick access to patient data, and an expanding infrastructure to deal with one of the latest big data concerning health arising from wearables and IoT gadgets in contemporary healthcare [6].

Many sectors have made leaps in progression toward data-driven productivity. However, the convergence of cloud computing is substantially transformative for the future of the health industry [7]. Value formation in healthcare refers to the creation and dissemination of value to patients, providers, and other stakeholders [8]. However, with a massive interconnected tapestry of devices, security issues usually become more important as there are multiple doors to enter for cybercriminal acts. Such attacks can lead to interrupting some vital services in healthcare systems, data breaching and unauthorized access to confidential private information of patients [9]. These health issues tend to have a rather negative impact on quality of life among elderly adults while quite a few of these costs are borne by healthcare systems, caretakers, and families [10]. Optimistically, AI and cloud computing are resulting in the development of advanced systems for disease prediction which is eventually expected to open avenues in the healthcare field [11]. The integration of Artificial Intelligence (AI), Big Data Mining, and the Internet of Things (IoT) has revolutionized healthcare applications in the modern medical practice [12].

The paper is systematically divided into the following sections. Literature review on existing techniques is presented in Section 2. Section 3 describes the methodology proposed for the work. Section 4 presents the results that evaluate the performance and security of the proposed system. Finally, Section 5 concludes the paper.

2. LITERATURE SURVEY

The RSA (Rivest-Shamir-Adleman) algorithm is one of the many tools that Yallamelli et al. [13]used in enhancing data security during cloud computing. RSA can secure communications over a network that is not secure by prime factorization complexity in the encryption and decryption procedure. RSA has undoubtedly proved to enhance privacy, integrity and authenticity in digital systems without depending on any other secret shared keys after it was invented in 1977. It displays its versatility in varying application areas, including internet connection and email security protocols.

Symmetric key encryption is added on top of DPOS, as found by Alagarsundaram et al. [14],to ensure that prior to storage data are secured off-the-shelf users may have confidentiality and effective data deduplication for accessing data. The pure use of the storage space was through elimination and identification of duplicate copies of data, while the proof of storage was simplified to help users ascertain the correctness of data without complex decryption. This becomes important for the integrity of data in today's digital environments and in showing the technological hurdles of an integrity auditing protocol in Sec-DPoS.

Recognition of significant challenges in maintaining data management and the way to overcome data security, access, and sharing, was done by Nagarajan et al. [15]. The research hence underscored the game-changing advent of the convergence of cloud computing and GIS, based on in-depth literature review and case study synthesis. The testimony made is for the effect of how disaster management, health studies, environmental risks assessment, sustainable energy, conservation, and engineering geology would take particular applications up.

Deevi et al. [16] proposed a secure mobile healthcare model based on WBANs with a multi-biometric key generation approach. The model used cloud computation to operate on and store data massively in a reliable and flexible manner. Feature extraction from EEG and ECG signals was done using Discrete Wavelet Transform (DWT) because this would assist in key generation and in enhancing security.

Deepa et al. [17] investigated the state-of-the-art progress in applying deep learning (DL) and machine learning (ML) technologies to enhance fraud detection capabilities. The study showed how significant the improvement in detecting fraud is by employing algorithms such as logistic regression, decision tree algorithms, support vector machines, convolutional neural networks (CNN), and recurrent neural networks (RNN) on big data sets.

Valivarthi et al. [18] developed the BBO-FLC and ABC-ANFIS model for increasing the prediction accuracy and monitoring of diseases. The model, incorporated within a dynamic cloud-based scalable architecture, used data acquisition with IoT-enabled sensors, feature optimization using ABC, tuning fuzzy rules via BBO, and disease classification with ANFIS. Therefore, the resulting system demonstrated augmented precision and scalability while significantly improving healthcare applicability in complex disease prediction and monitoring.

2.1 PROBLEM STATEMENT

Existing works has greatly improved cryptographic processes and the security of sensitive information within cloud computing environments but has yet to achieve everything, including requiring hardware acceleration to enhance cryptographic methods, incorporating new algorithms to ward off upcoming threats, and the ability to complicate configuring and employing cryptographic frameworks in an optimal manner [19]. In addition, deploying AES into cloud computing platforms to avoid unauthorized access and cyberattacks is a key challenge [20]. The aim is to overcome these challenges through enhancing cryptographic techniques, optimizing encryption methods and adherence to changing security standards to advance the security and efficiency of cloud-based medical systems while protecting sensitive patient information.

3. PROPOSED METHODOLOGIES

The proposed methodology is to design a secure and effective system to manage sensitive health information through up-to-date cryptography methods. This approach combines post-quantum cryptography with symmetric encryption, with the N-th degree Truncated Polynomial Ring Units algorithm for generating keys and Salsa20 stream cipher for encrypting data. The system provides safe data storage and transmission through the creation of a unique key pair via NTRU, and then encryption of the dataset via Salsa20. The encrypted data is stored in the cloud, keeping it confidential and intact. Only valid receivers, with the proper decryption keys, can decrypt and retrieve the data, providing both privacy and security. This method brings together the quantum resistance of NTRU for key exchange and Salsa20 for secure data encryption quickly, offering a future-resistance solution to the future security threats posed by quantum computing. Overall, the approach is depicted in Figure 1.

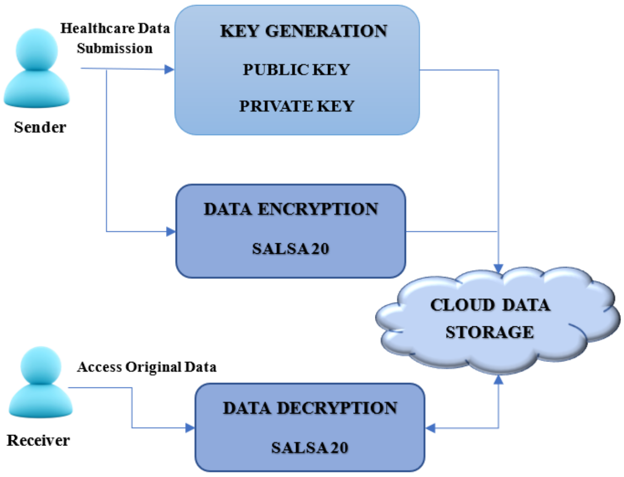

Figure 1: Secure Healthcare Data Transmission in Cloud

3.1 Dataset Submission

The dataset submission is the first step in sending such a dataset to encrypt and secure it for storage. Sensitive healthcare datasets may include patient demographics, medical history and diagnosis information. This dataset is then transferred through secure lines, freeing it from unauthorized access during transmission. Upon ensuring that the dataset is whole, accurate and privacy-preserving, it marks the starting point of securely encrypting and storing the data for future use.

3.2 Key Generation

Key generation is the generation of a pair of cryptographic keys. During key generation, one key is used for encryption and decryption-the private key-whereas the other key is used for securing communication and the transfer of secured data-the public key. Here in the proposed approach, key generation involves the application of NTRU algorithm thereby embedding a post-quantum-security-based framework within the public-private key pair generation. The sender starts the key generation using two polynomials, f and g, with small integer coefficients; the private key is generated and the public key is calculated as $h = g\ f$ mod$q$, where q is a large modulus, whereas the private key is kept secret by the sender, and h is sent to the receiver for use in encryption. The NTRU key generation algorithm provides security using lattices against quantum attacks, thereby enabling a secure transmission of sensitive health data. This forms the basis of a secure key exchange method to be followed by both parties for the secure encryption and decryption of data across the system.

3.3.1 Private Key Generation

The private key consists of two polynomials, $f(x)$ and $g(x)$, chosen randomly from sets of polynomials with small integer coefficients. Let f(x) and g(x) be polynomials defined over rational numbers whose equations are given in (1,2).

$$f(x) = \sum_{i=0}^{N-1} f_i x^i \ (\mathrm{mod} q) \tag{1}$$

$$g(x) = \sum_{i=0}^{N-1} g_i x^i \ (\mathrm{mod} q) \tag{2}$$

Where fi and gi are random integers within a specified range, and N is the polynomial degree. The

private key in NTRU consists of two polynomials, $f(x)$ and $g(x)$, chosen randomly with small integer coefficients. These polynomials are employed in the encryption and decryption process; thus, the private key is kept secret by the sender and by creating small coefficients for the polynomials $f(x)$ and $g(x)$, efficiency and security in the encryption process were ensured.

3.3.2 Public Key Generation

The public key $h(x)$ is generated from the private key polynomials through modular arithmetic, as exhibited in equation (3).

$$h(x) = \frac{g(x)}{f(x)} \bmod q \tag{3}$$

The division is performed modulo q, with q being a large modulus. Therefore, the result of such operation would provide the public polynomial $h(x)$, which may in turn be publicly announced for encryption. In this way, it can be seen that the NTRU public and private keys are closely related to each other through this polynomial, and then their security relies on the complexity of recovering the private key from the public key trying to solve the lattice-theoretic shortest vector problem.

3.3 Encryption

The Salsa20 stream cipher has been used for encryption. The sender generates a random 256-bit key and an 8-byte nonce. Then the plaintext dataset is encrypted by XORing with the keystream generated by Salsa20 using the key and nonce, ripping out the ciphertext such that the data is thus transformed into an unreadable state for secure transfer. Another layer of encryption for this data is applied by storing it on the cloud to protect against an eavesdropper when stored and transmitted over a network. The receiver, who possesses the corresponding key and nonce, is able to decrypt the data back to the original form.

The encryption process using Salsa20 can be expressed as equation (4)

$$C = P \oplus K_{\text{Salsa20}}\,(K, N) \tag{4}$$

Where, C is the ciphertext (the encrypted data), P is the plaintext (the original data), $\oplus$ denotes the XOR operation, $K_{\text{Salsa20}}\,(K, N)$ is the keystream generated by the Salsa20 cipher using the 256-bit key K and 8 -byte nonce N.

Keystream Generation: The Salsa20 will generate a keystream from the key K and the nonce N. The keystream is a sequence of pseudo-random bytes. XOR operation: The plaintext P is XORed with the keystream to create the ciphertext C. This is applicable for the transformation of data from a readable format into an unreadable format. Decryption: Decryption is the reverse process in which the same key with the same nonce is used to generate the same keystream and the ciphertext is XORed with that keystream to retrieve the original plaintext.

This equation captures the core of the encryption process, ensuring data confidentiality during transmission and storage.

3.4 Cloud Storage

Cloud storage is used for securing the data such that it is secure at all times, easily accessible, and made highly available. In the process, a dataset is encrypted by the Salsa20 encryption and then uploaded, the ciphertext (the encrypted version of the actual data) to a cloud storage platform for safe storage purposes and scalability. The encrypted data is transferred across very secured channels to prevent unauthorized tapping while in transit to its new home: the cloud service, which will then offer the required storage, scalability, and redundancy such that even with the failure of that system, the encrypted data remains safe. No unauthorized third parties can read this encrypted dataset as it is stored in an encrypted format, hence safeguarding the privacy of sensitive healthcare data.

3.5 Decryption

Decryption is the mechanism through which the receiver retrieves the original dataset from the encrypted data in the cloud. After obtaining the ciphertext from the cloud storage, the receiver employs the same 256-bit key and 8-byte nonce used in the encryption process with Salsa20. Then the receiver performs the Salsa20 decryption algorithm, which is essentially XOR with the ciphertext with the keystream computed with the key and the nonce. So, in a nutshell, it returns the plaintext (health dataset) back to its original state. The process of decryption enables only legitimate users with the right key and nonce to retrieve the original data; that means it protects the confidentiality and integrity of the dataset while being stored and transmitted.

The decryption process using Salsa20 can be mathematically represented as equation (5)

$$P = C \oplus K_{Salsa20}(K, N) \tag{5}$$

Where, P is the plaintext (the original data after decryption), C is the ciphertext (the encrypted data), $\oplus$ denotes the XOR operation $K_{\text{Salsa20}}\ (K, N)$ is the keystream generated by the Salsa20 cipher using the 256 -bit key K and 8 -byte nonce N.

Creation of Keystream: Keystream is produced by means of Salsa20 algorithm with the same secret K and nonce N as during the encryption. Performing XOR Operation: The ciphertext C is then XOR-joined up with keystream $K_{\text{Salsa 20}}(K, N)$, resulting in original plaintext P. Reversibility: Since encryption and decryption both happen with the same key and nonce hence appropriate XOR operation results in reversal of process that is original data gets restored. The equation is such that the encrypted data would be able to decrypted successfully by the receiver, as long as the provided there is the right key with nonce.

4. RESULT

The result indicates a positive correlation between key length and time taken for key generation, encryption and decryption. As key length increases, times for both encryption and decryption grow by a considerable amount because of the increased complexity of computation as well as the increased power required for processing. Thus, there is a cost-benefit in terms of having longer keys: enhanced security but then increased computational overhead, which can thus affect the efficiency of cryptographic systems.

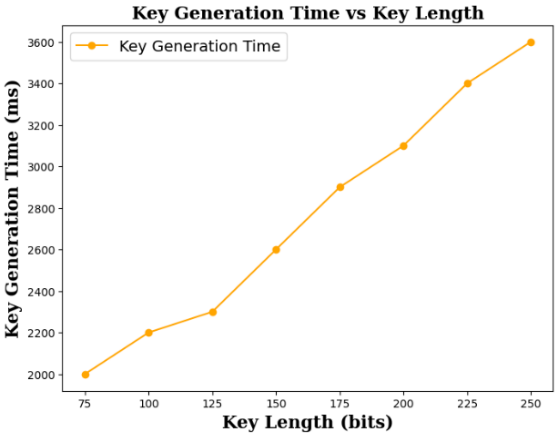

Figure 2: Key generation

From figure 2, a direct trend of variation can quite clearly be observed between the time required to generate a key and its length. The time taken to generate such keys is approximately doubled as the length of the keys increased from 75 bits and 250 bits. Which ultimately means that keys with longer bit-length values will require higher computation resources. For example, key generation varies about 2200 ms for the 75-bit-long key to some 3500 ms for the 250-bit-long key. This trend indicates that there exists a trade-off between security and performance since longer keys produce higher cryptographic strength but also take more computational resources, affecting the performance of cryptographic systems.

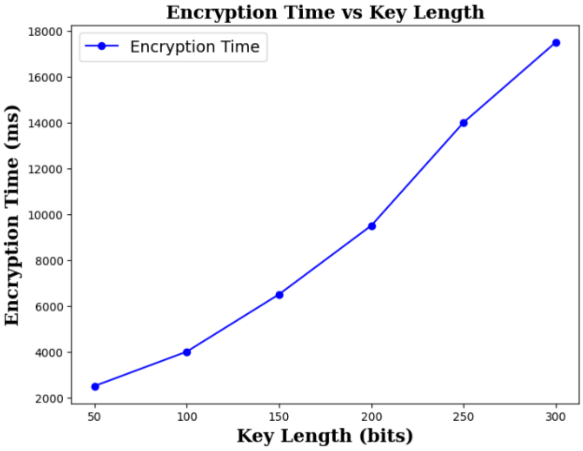

Figure 3: Encryption Time

The relationship between Encryption Time and Key Length is positively correlated as indicated by Figure 3. For increasing key lengths between 50 bits and 300 bits, we get a corresponding progressive increase in encryption time. For example, at 50 bits, encryption time was approximately 2000 ms, and gradually increased with an increase in the key length: for 100 bits, it was approximately 4000 ms; for 150 bits, it became 6000 ms; for 200 bits, it was 8000 ms, and for 250 bits, 12000 ms; finally, for 300 bits, it became 16000 ms. With this, the increase is also seen in the computational complexity in encryption with longer keys, thus emphasizing the trade-off between security and efficiency. Longer keys imply more security but also require more processing time for their encryption.

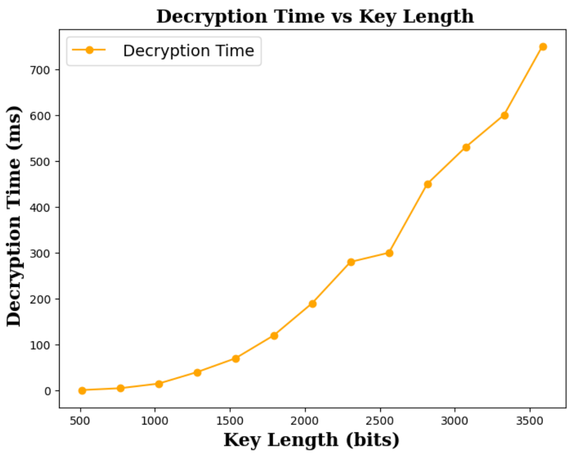

Figure 4: Decryption Time

Figure 4 publishes the chart for Decryption Time and Key Length positively correlated to each other. As the key length rises from 500 bits (nearly 10 ms) to 3500 bits (more than 700 ms), the decryption time rises steadily. For instance, it takes around 30 ms to decrypt a message with a key length of 1000 bits, while it takes 60 ms at a key length of 1500 bits and 250 ms at a key length of 2500 bits. Hence, more processing time is required to decrypt with larger key lengths. This is the practical manifestation of the trade-off between security and computational efficiency in cryptographic systems.

5. CONCLUSION

With the above considerations, it is interesting that the key generation, encryption, and decryption times clearly indicate trade-offs between security and computational efficiency. Higher key lengths consider more computational time for any of the processes. Key generation time, encryption time, and decryption time grew positively concerning key length: that is, for larger key lengths, the cryptographic processes become complicated. The key generation time increases from 2200 ms for

75-bit keys to 3500 ms for a 250-bit key, and encryption time for the 50-bit key goes from 2000 ms to 16000 ms for a 300-bit key. Likewise, decryption time ranges from about 10 ms for 500 bits to over 700 ms for 3500 bits. Such trends confirm the additional computational effort needed as key lengths increase; this, in turn, implies a higher degree of cryptographic strength but adds to the load on computational efforts. This factor is particularly important in types of applications like cloud computing, where the performance of encryption and decryption processes affects global system efficiency. This should guarantee optimal lengths for keys selected, fulfilling very high security requirements, and performance criteria as well. Such a balance will be important for having solid functioning implementations of cryptographic systems in real-world scenarios.

REFERENCE

[1] V. S. B. H. Gollavilli, “Securing Cloud Data: Combining SABAC Models, Hash-Tag Authentication with MD5, and Blockchain-Based Encryption for Enhanced Privacy and Access Control,” International Journal of Engineering Research Science and Technology, XVIII (3), pp. 149–165, Aug. 2022.

[2] A. R. G. Yallamelli, “Cloud Computing and Management Accounting in SMEs: Insights from Content Analysis, PLS-SEM, and Classification and Regression Trees,” International Journal of Engineering, XI (3), 2021.

[3] J. Bobba, “Enterprise Financial Data Sharing and Security in Hybrid Cloud Environments: An Information Fusion Approach for Banking Sectors,” International Journal of Engineering, XI (3), 2021.

[4] S. Narla, S. Peddi, and D. T. Valivarthi, “Optimizing Predictive Healthcare Modelling in a Cloud Computing Environment Using Histogram-Based Gradient Boosting, MARS, and SoftMax Regression,” International Journal of Management Research Business Strategy, XI (4), pp. 25–40, Nov. 2021.

[5] D. T. Valivarthi, S. Peddi, and S. Narla, “Cloud Computing with Artificial Intelligence Techniques: BBO-FLC and ABC-ANFIS Integration for Advanced Healthcare Prediction Models,” International Journal of Information Technology Computer Engineering, IX (3), pp. 167–187, Aug. 2021.

[6] S. Narla, D. T. Valivarthi, and S. Peddi, “Cloud Computing with Healthcare: Ant Colony Optimization-Driven Long Short-Term Memory Networks for Enhanced Disease Forecasting,” *International Journal of HRM and Organizational Behavior*, VII (3), pp. 12–26, Sep. 2019.

[7] S. R. Sitaraman, “AI-Driven Healthcare Systems Enhanced by Advanced Data Analytics and Mobile Computing,” *International Journal*, XII (2), 2021.

[8] S. R. Sitaraman, “AI-Driven Value Formation in Healthcare: Leveraging the Turkish National AI Strategy and AI Cognitive Empathy Scale to Boost Market Performance and Patient Engagement,” *International Journal*, XIV (3), 2023.

[9] T. Ganesan, “Securing IoT Business Models: Quantitative Identification of Key Nodes in Elderly Healthcare Applications,” *International Journal*, XII (3), 2022.

[10] S. Peddi, S. Narla, and D. T. Valivarthi, “Harnessing Artificial Intelligence and Machine Learning Algorithms for Chronic Disease Management, Fall Prevention, and Predictive Healthcare Applications in Geriatric Care,” *International Journal of Engineering Research and Science Technology*, XV (1), pp. 1–15, Feb. 2019.

[11] S. Narla, “Cloud Computing with Artificial Intelligence Techniques: GWO-DBN Hybrid Algorithms for Enhanced Disease Prediction in Healthcare Systems,” *Current Science*, XX (X), 2020.

[12] R. Budda, “Integrating Artificial Intelligence and Big Data Mining for IoT Healthcare Applications: A Comprehensive Framework for Performance Optimization, Patient-Centric Care, and Sustainable Medical Strategies,” *Journal Name*, XI (1), 2021.

[13] A. R. G. Yallamelli, “Improving Cloud Computing Data Security with the RSA Algorithm,” *Journal Name*, IX (2), 2021.

[14] P. A. Poovendran Alagarsundaram, “Symmetric Key-Based Duplicable Storage Proof for Encrypted Data in Cloud Storage Environments: Setting Up an Integrity Auditing Hearing,” *International Journal of Engineering Research Science and Technology*, XVIII (4), pp. 73–86, Oct. 2022, doi: 10.62643/ijerst.2022.v18.i04.pp73-86.

[15] H. Nagarajan, “Streamlining Geological Big Data Collection and Processing for Cloud Services,” *Journal Name*, IX (9726), 2021.

[16] D. P. Deevi, “Improving Patient Data Security and Privacy in Mobile Health Care: A Structure Employing WBANs, Multi-Biometric Key Creation, and Dynamic Metadata Rebuilding,” *International Journal of Engineering Research Science and Technology*, XVI (4), pp. 21–31, Dec. 2020.

[17] N. K. R. Panga, "Financial Fraud Detection in Healthcare Using Machine Learning and Deep Learning Techniques," *International Journal*, X (3), 2021.

[18] D. T. Valivarthi, S. Peddi, and S. Narla, "Cloud Computing with Artificial Intelligence Techniques: BBO-FLC and ABC-ANFIS Integration for Advanced Healthcare Prediction Models," *International Journal of Information Technology and Computer Engineering*, IX (3), pp. 167–187, Aug. 2021.

[19] D. T. Valivarthi, "Implementing the SHA Algorithm in an Advanced Security Framework for Improved Data Protection in Cloud Computing via Cryptography," *International Journal*, X (3), 2022.

[20] P. Alagarsundaram, "Implementing AES Encryption Algorithm to Enhance Data Security in Cloud Computing," *International Journal*, VIII (2), 2020.