Indian Journal of Information Sources and Services
ISSN: 2231-6094 (P) Vol.16, No.1, 2026, pp.120-130

DOI: https://doi.org/10.51983/ijiss-2026.16.1.13

Lightweight Homomorphic Encryption Algorithm for Secure Cloud Storage in Bibliographic Control Systems

Liaqat Ali[1], Taher M. Ghazal[2*], Abdelrahman H. Hussein[3], Amjed Abbas Ahmed[4] and Syed Muqtar Ahmed[5]

[1]Department of Cybersecurity, College of Computer, Qassim University, Buraydah, Saudi Arabia
[2*]Faculty of Computing and IT, Sohar University, Oman; Department of Networks and Cybersecurity, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan; Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor, Malaysia
[3]Department of Networks and Cybersecurity, Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman, Jordan
[4]Center for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi, Selangor, Malaysia; Department of Computer Techniques, Engineering Imam AlKadhum College (IKC), Baghdad, Iraq
[5]Software Engineering Department, College of Engineering, University of Business and Technology, Jedda, Saudi Arabia
E-mail: [1]l.ali@qu.edu.sa, [2]taher.ghazal@ieee.org , [3]a.husein@ammanu.edu.jo, [4]amjedabbas@alkadhum-col.edu.iq, [5]syedahmed@ubt.edu.sa
ORCID: [1]https://orcid.org/0000-0003-3024-3014, [2]https://orcid.org/0000-0003-0672-7924, [3]https://orcid.org/0000-0001-6536-7485, [4]https://orcid.org/0000-0001-6069-2967, [5]https://orcid.org/0000-0003-1636-3618

(Received 23 October 2025; Revised 22 November 2025, Accepted 05 December 2025; Available online 05 January 2026)

Abstract - **Empirical research conducted over the past few years suggests that the rapid expansion of digital bibliographic data in cloud-based library management and data retrieval systems has heightened concerns regarding confidentiality and privacy. Classical methods of encryption protect stored data, but they do not permit efficient access to encrypted data. This lack of functionality makes the system nearly unusable. This study develops a homomorphic encryption technique for safe bibliographic cloud storage and retrieval in order to address these problems. In order to accomplish security and access control goals without sacrificing the effectiveness of cataloging and retrieval systems, partially homomorphic encryption is utilized. It is more secure than both encrypted and unencrypted systems. When forced to employ completely homomorphic techniques, libraries and other archive systems frequently experience significant computational delays, which makes the systems slow. For bibliographic databases with high query traffic, systems that use the suggested LHE framework scale with nominal computing cost, have low latency, and have minimal ciphertext overhead—the ideal setup for systems that need to respond quickly. The algorithm's effectiveness is confirmed by its ability to survive numerous known cryptographic assaults while exceeding traditional systems in terms of encryption speed, storage efficiency, and query response time. By strengthening secure cataloging, controlled access, and privacy-centered search operations, LHE enhances bibliographic control systems and is beneficial for academic cloud infrastructures and next-generation digital libraries. This research augments the domains of cloud security and library and information science by offering innovative strategies for the secure management of bibliographic data.**

Keywords: **Lightweight Homomorphic Encryption, Secure Cloud Storage, Bibliographic Control Systems, Privacy-Preserving Search; Metadata Security, Cloud Security, Digital Libraries, Cryptographic Optimization**

I. INTRODUCTION

1.1 Background on Bibliographic Control Systems and Their Role in Digital Libraries

Bibliographic control systems are critical components of digital libraries and scholarly repositories, facilitating the systematic organization, cataloging, and retrieval of bibliographic records, including books, journals, reports, and digital files. These systems maintain uniformity in metadata standards and enable seamless retrieval of information across international networks (Viswanath & Krishna, 2021). Due to the growing popularity of cloud computing in library and information services, bibliographic control systems now operate in cloud environments (Choudhary & Singh, 2022; Xu et al., 2021). These systems are hosted on distributed infrastructures that provide scalability, cost-efficiency, and ubiquitous access. However, these systems raise significant issues in data confidentiality, access control, and privacy of sensitive bibliographic information (Thabit et al., 2021).

1.2 Challenges in Secure Cloud Storage

Cloud-based bibliographic systems contain a considerable amount of metadata and user activity logs, and are therefore vulnerable to unauthorized access, unauthorized disclosure due to insider actions, and data breaches. Current practices associated with cloud storage generally rely on encryption as the primary method of keeping sensitive data secure (Agrawal, 2021). Recent research has shown that vulnerabilities are often linked to inadequate key management, as providers lack sufficient privacy-preserving mechanisms, and they are also susceptible to sophisticated cyberattacks (Samyadevi et al., 2024). Furthermore, academic libraries must also create enough mechanisms for confidentiality, integrity, and controlled access to comply with data governance standards (Singh et al., 2021).

1.3 Limitations of Traditional Encryption in Supporting Search/Query Operations

Conventional encryption schemes (e.g., AES and RSA) provide data confidentiality during storage and transmission, but they limit their functionality when computing over ciphertext (Choi et al., 2025). In bibliographic systems, computations with queries against metadata (i.e., keyword search, catalog index, and ranked retrieval) occur frequently. Information retrieved from traditional schemes requires total decryption, which costs more away from the actual computation and exposes sensitive data that reveals critical information, as well as vulnerability to breaches. The computational and security restrictions reduce the examined value of conventional encryption for large-scale digital library infrastructures requiring secure storage and efficient retrieval (Gurunath et al., 2026).

1.4 Role of Homomorphic Encryption (HE) in Privacy-Preserving Cloud Systems

Homomorphic encryption (HE) is a unique solution that enables computing on encrypted data without the need to decrypt it, thereby realizing privacy-preserving search, secure query processing, and encrypted metadata indexing in bibliographic systems (Salem et al., 2018). Recent applications of HE have been explored in healthcare, finance, and cloud applications that maintain sensitive data security when being processed (Chowdary et al., 2024). However, direct application of fully homomorphic encryption (FHE) is not practical for bibliographic applications despite FHE's strong security assurances, because the overhead of FHE is quite high, together with its substantial ciphertext expansion, as well as long latencies (Nkenyereye et al., 2019).

1.4 Research Gap

While existing HE methods are secure for the specific services they provide, they often become inefficient when applied at scale for large bibliographic control systems where there are high quantities of metadata and minimal query response times (Yazdinejad et al., 2020). Fully homomorphic encryption (FHE) may conceptually allow for powerful operations over encrypted data, but it is computationally heavy and not a viable solution for real-time bibliographic search, as a database rebuild takes less time than a search of an encrypted database with FHE (Choi et al., 2021). The combination creates a critical gap for solutions that are lightweight, efficient, and provide privacy-preserving features in a bibliographic context (Malik et al., 2023).

1.5 Objectives

This paper seeks to design and assess a Lightweight Homomorphic Encryption (LHE) algorithm for bibliographic control systems in the cloud. The algorithm addresses the following goals:

- Improve security of bibliographic metadata storage and access;
- reduce computational and storage overhead compared with traditional he methods;
- allow for secure and efficient keyword search of and querying on encrypted records;
- improve scalability and usability within digital libraries.

1.6 Paper Organization

The rest of the paper is structured as follows: Section 2 presents a literature review on homomorphic encryption and its applications in cloud security and bibliographic systems. Section 3 details the research methodology, including system architecture and algorithm design. Section 4 outlines the experimental setup and dataset used for evaluation and discusses the results and performance analysis. Section 5 provides a critical discussion on findings, implications, and limitations. Finally, Section 6 concludes the paper and highlights directions for future research.

II. LITERATURE SURVEY

Homomorphic encryption (HE) is now one of the essential developments in cloud security because it provides the possibility to compute on data in its encrypted form without decryption (Yang et al., 2020). HE has come a long way in terms of development, and different categories of HE schemes have emerged: partial, somewhat, and fully homomorphic encryption (Ahamed & Ravi, 2022). Partial homomorphic encryption schemes (e.g., RSA or variants of RSA and Paillier) support limited operations (i.e., addition or multiplication) to gain better efficiency as a trade-off on functionality. Somewhat homomorphic encryption (SHE) schemes also offer support to a limited set of both operations, but restrict the number of computations per instance. Fully homomorphic encryption (FHE) offers the ultimate support of unlimited computations on ciphertexts with a fidelity of privacy. Nevertheless, despite the general promise of widely accepted theoretical support, FHE has presented itself with significant limitations with respect to computational complexity, memory overhead, and ciphertext expansion, preventing it from being deployed on any large scale, especially in a real-time system (Ayeblo & Faraahi, 2015). Large-scale deployments of he is unlikely because of its demands on resources, and consumers do not need the

uncertain level of encrypted privacy FHE demands. Considering the limitations of HE, researchers are exploring lightweight cryptographic designs that offer better trade-offs towards either security or performance. Lightweight cryptographic techniques can reduce the algorithm complexity, the time to encrypt, and/or the time to decrypt, and reduce resource-consuming recurrent encryption schemes for large, cloud-based storage systems that rely on high-volume data inputs (Cortés-Mendoza et al., 2020).

The challenge of secure and effective searchable encryption is especially motivating in the context of bibliographic control systems and digital libraries, which host a large number of metadata records and bibliographic indexes, and receive frequent queries related to searching, retrieving, and ranking, for example, using keywords or performing catalog search functions (Alsharifi, 2023). Current bibliographic control systems often provide users with symmetric and asymmetric encryption schemes to protect bibliographic records. However, efficient implementations of symmetric encryption schemes like AES (Advanced Encryption Standard) allow for quick computations but not useful operations on encrypted data (Kanchan et al., 2025). Therefore, users have to obtain their data from the record using decryption every time they query. The challenge with this approach is that the metadata in the record will expose privacy, and users will not have an efficient operation after the decryption process (Ali et al., 2021). Asymmetric schemes can provide some additional flexibility in key management as they are more secure than symmetric schemes, but asymmetric schemes require computationally intensive operations to encrypt/decrypt, which limits their performance, especially for those operations based on metadata flipping, for example, multi-keyword search and ranked retrieval. Additionally, there are no implementations of either scheme in a generalized specification of secure and efficient query support (Jia et al., 2020). As a consequence, m a usability standpoint, the taxonomy of encryption targeted towards cryptographic functions has significant barriers towards universal acceptance for bibliographic control systems in the digital library context, requiring confidentiality while researching. Nevertheless, there is promise with the work recently done with searchable encryption combined with homomorphic functionalities in terms of functionally defined privacy options (Noorallahzade et al., 2022). Researchers have discovered and executed homomorphic encryption for secure indexing, ranked queries, and controlled distributed cloud-based information systems in fruition. Nonetheless, the ongoing presence of performance issues represents a challenge for implementing these techniques in bibliographic systems - particularly when it is possible to journal or report that most approaches have known topics such as large ciphertext, which leads to a significant storage overhead, additional latency on encryption, queries, etc., and difficulties that arise when dealing with massive bibliographic datasets (Jia et al., 2020). These issues continue to show the need for lightweight homomorphic encryption schemes tailored for bibliographic metadata, which allows for secure cataloging, querying, and storage while providing data privacy protections. With actionable lightweight alternatives available, bibliographic control systems hosted in the cloud have the potential to give privacy and performance to improve the reliability of the available digital libraries and repositories they sit on for use in research (Kim et al., 2021).

III. RESEARCH METHODOLOGY

3.1 System Architecture

Fig. 1 presents the envisioned cloud-based bibliographic control framework employing the Lightweight Homomorphic Encryption (LHE) technique. The ecosystem is comprised of three key components - the Client (Library/User) that encrypts bibliographic records and queries locally and then transmits; the Cloud Server that stores encrypted metadata, and processes with homomorphic functions such as indexing, keyword matching, and ranked retrieval without requiring decryption; and the Authorized User or Model Owner to decrypt the ciphertext results from the previous step using the secret key. Importantly, as outlined in Fig 1, all bibliographic-related information can remain private (e.g. catalog metadata, search logs, and access policies) throughout the entire catalogue data storage, processing, and retrieval cycle, allowing users to manage and maintain bibliographic data in a cloud environment securely and privately.

3.2 Mathematical Model

3.2.1 Key Generation (LHE Setup)

Given two large primes p and q, the key generation process works as follows:

- **Public Key (PK):**

$$PK\ (n = p.q, g),\ g \epsilon Z_{n^2}^{*},\ is\ the\ product\ of\ p, q \qquad (1)$$

where n is the modulus, and g is a generator in the group.

- **Private Key (SK):**

$$\boldsymbol{SK = \lambda = lcm(p-1, q-1)} \qquad (2)$$

where λ is the least common multiple of p−1 and q−1.

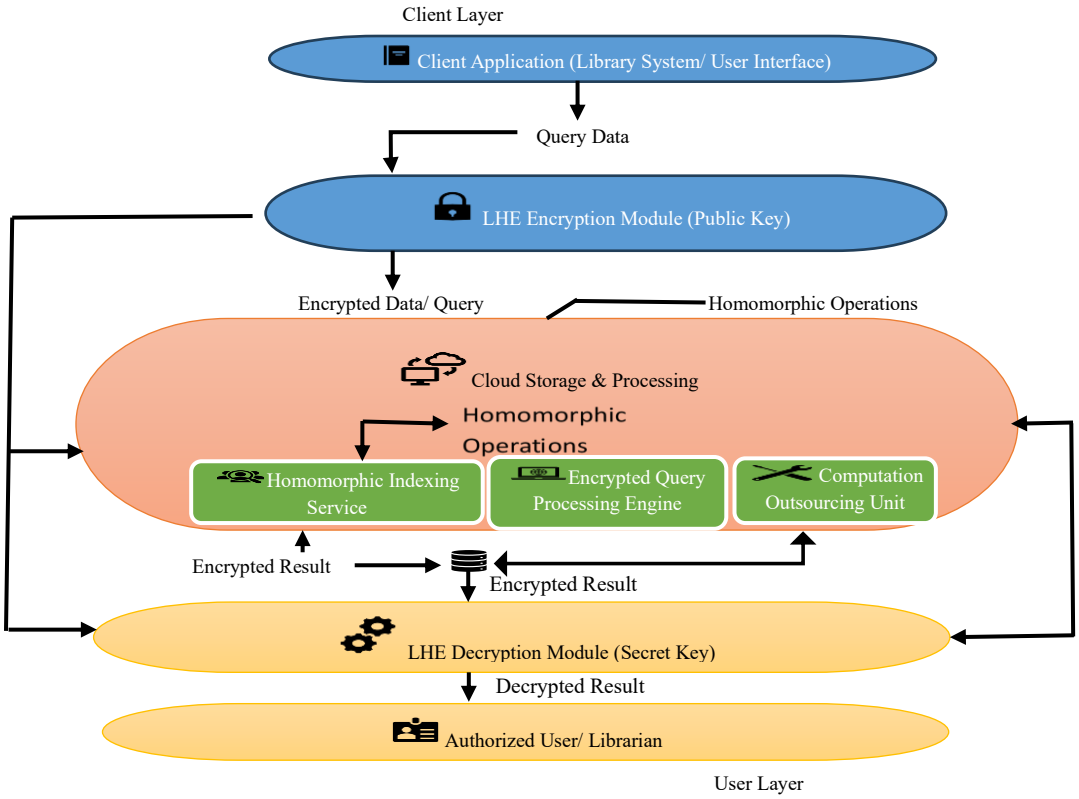

Fig. 1 Methodology Flow

3.3 Mathematical Model

3.2.1 Key Generation (LHE Setup)

Given two large primes p and q, the key generation process works as follows:

- **Public Key (PK):**

$$PK\ (n = p.q, g),\ g \epsilon Z_{n^2}^{*},\ is\ the\ product\ of\ p, q \qquad (1)$$

where n is the modulus, and g is a generator in the group.

- **Private Key (SK):**

$$\boldsymbol{SK = \lambda = lcm(p-1, q-1)} \qquad (2)$$

where λ is the least common multiple of p−1 and q−1.

3.2.2 Encryption of Bibliographic Data (Record Encryption)

For each bibliographic record rir_iri, we first create a binary feature vector xi=[x_{i1}, x_{i2}, …, x_{iM}] where M is the number of keywords (terms in the vocabulary W) and $x_{ij} \in \{0,1\}$.

The LHE encryption for each record's keyword feature vector is defined as follows:

- **Packing and Encryption**

The term weights for each record are packed into small integers. Let $x_i=[x_{i1},x_{i2},\ldots,x_{iM}]$ be the binary vector representing the presence (or TF/IDF weight) of terms. For each block of BBB consecutive keywords, we construct a packed integer $m_{i,l}$ as:

$$m_{i,l} = \sum_{b=0}^{B-1} x_{i,j_b}.2^{b\Delta}, \qquad (3)$$

Where Δis the number of bits used for packing the terms, and j_b represents the index of the b-th term.

- **Encrypted Term (Ciphertext ccc)**

The packed integer is encrypted as:

$$c_{i,l} = E_h(m_{i,l}) = g^{m_{i,l}} \qquad (4)$$

Where r is a random number chosen for semantic security and n is the modulus from the public key.

3.2.3 Homomorphic Operations on Encrypted Data (Server-side)

The cloud server performs encrypted computations over the encrypted records. For instance, to compute the relevance of a record to a query, we first need to represent the query as an encrypted vector.

- **Query Representation**

For a multi-keyword query Q⊆W, we encode the query similarly to the records:

q=[q1,q2,…,qM],qj∈{0,1}or weighted based on TF/IDF. We then pack the query into encrypted blocks:

$$q_l = \sum_{b=0}^{B-1} q_{j_b} . 2^{b\Delta} \quad (5)$$

And encrypt it:

$$Q_l = E_h(q_l) \quad (6)$$

- **Homomorphic Scoring (Encrypted Dot Product)**

The cloud server computes an encrypted score for each record r_i based on the encrypted query Q and the encrypted record C_i:

$$S_i = \bigoplus_{\underset{i}{\cup}} (C_{i,l} \odot Q_l) \quad (7)$$

Where $\odot$ denotes the homomorphic multiplication and $\oplus$ represents homomorphic addition. This allows the cloud server to compute the relevance score without decrypting the records.

3.2.4 Decryption and Ranking (Client-side)

Once the encrypted scores are returned, the client decrypts the result using the private key:

- **Decryption of Score S_i**

The client decrypts the homomorphic score using the private key SK:

$$s_i = D_h(S_i) = \sum_l q_l . m_{i,l} \quad (8)$$

The client then ranks the records by the decrypted score s_i, and the top-k records are returned.

3.2.5 Security and Privacy Guarantees

The security of the designed system relies on the IND-CPA security of the Paillier-based encryption scheme, in which the ciphertexts cannot reveal any information regarding the plaintext unless the secret key is revealed. Privacy of queries is also guaranteed while the query is encrypted before submission to the server, and the server cannot gain information regarding the user's search goals. Finally, fresh randomness for encryption means that even if the same query is repeatedly made, there can be no reason to make correlations.

3.2.6. Storage and Computational Efficiency

The storage overhead of this scheme is mostly dependent on the ciphertext size, which together with the packing of multiple terms into a single ciphertext has reduced the expansion factor to roughly 1.2× the original data size, as opposed to the 2–2.5× overhead utilized in existing homomorphic encryption schemes (e.g., Paillier), and therefore maximizes the merit of the scheme for large-scale bibliographic control systems.

The scheme also allows computational efficiency to be considerably improved by utilizing packed ciphertext and homomorphic modifications of ciphertexts, which are useful for efficiently processing encrypted queries and avoiding decryption on the side of the server.

3.3 Algorithm Design

The Lightweight Homomorphic Encryption Algorithm (LHEA) improves upon standard HE by reducing ciphertext expansion and computational overhead. The main steps are:

Algorithm 1: Lightweight Homomorphic Encryption for Bibliographic Systems

1. Key Generation

- Generate a large prime modulus N= p·q, where p and q are primes.

Public Key PK=(N,g), Secret Key SK=λ. (1)

2. Encryption

Given bibliographic metadata $m \in Z_N$,

$$c=E(m)=g^m \cdot r^N \quad (2)$$

where r is a random number ensuring semantic security.

3. Homomorphic Operations

Addition:

$$E(m_1).E(m_2)=E(m_1+m_2) \quad (3)$$

Multiplication by a Constant:

$$E(m)^k=E(k.m) \quad (4)$$

4. Decryption

Compute:

$$m = \frac{L(c^\lambda (mod\ N^2))}{L(g^\lambda (mod\ N^2))}\ (mod\ N) \quad (5)$$

where $L(x) = \frac{x-1}{N}$

5. Query Execution

Encrypted queries are matched with encrypted indexes.

Cloud performs computations on ciphertext and returns encrypted results to the user.

3.4 Data Flow

The data flow described here will help ensure that bibliographic metadata will be safeguarded at all aspects of its lifecycle from ingestion to query retrieval.

- Bibliographic Records Encryption: On the client side, bibliographic records (titles, authors, catalog IDs, metadata fields, etc.) will be converted to ciphertext (i.e.,

alarming quantum residue) with the Lightweight Homomorphic Encryption (LHE) algorithm. Therefore, the data will be protected before it gets to the cloud. Also, by encrypting data at the source, neither can the cloud service provider see the plaintext; the data wasn't even plaintext to begin with.

- Secure Storage: Encrypted metadata is stored in cloud databases that are optimized for indexing and retrieval. In traditional architectures, an organization may store but does not restrict plaintext from being stored in a cloud environment. The LHE-based framework ensures that only the ciphertext is deposited in the cloud. This minimizes both the collateral damage from unauthorized access to data, as well as reduces the harm that ex- or insider threats may incur.
- Homomorphic Indexing: The cloud server will perform homomorphic operations on the encrypted bibliographic records to form searchable indexes of those records. When enabling retrieval operations like keyword-based searching and navigating through catalog records, the plaintext is never revealed.
- Encrypted query submission: Users submit encrypted queries using the public key. By doing so, the server never learns the true search terms and can keep the user's query confidential.
- Encrypted result generation: The server receives the encrypted query, processes the encrypted index, and creates encrypted results. Thanks to homomorphic operations, the server can return correct matches to the client without needing to decrypt anything.
- Decryption: The authenticated client decrypts the returned ciphertext using the private key. The result presented to the user is ultimately the plaintext result for their bibliographic query.

This workflow maintains end-to-end privacy by ensuring that both the initial storing and indexing, along with the processing, in no instance reveal or expose sensitive bibliographic information or queries. 3.5 Security Model The framework described here must be able to counteract conventional adversarial threats within cloud-based bibliographic environments.

3.4.1 Threat Assumptions

- External Adversaries: These are adversaries that intercept communications while in transit (i.e., through man-in-the-middle attacks) or may be able to access ciphertexts that are stored.
- Insider Threats: These adversarial attacks may come as a result of malicious cloud service providers or privileged administrators who are compromised.

3.4.2 Adversary Capabilities

- Ciphertext-Only Attacks (COA): The attacker is only able to infer information from the ciphertext pattern information alone.
- Chosen-Plaintext Attacks (CPA): An adversary may attempt to infer secret keys by comparing the outputs of ciphertexts of the inputs that were chosen.

3.4.3 Privacy-Preserving Requirements

- No Metadata Leak: The server does not know either the bibliographic content or the user query terms.
- Key Management: There are secure distribution and revocation protocols in place, which prevent misuse of keys.
- Replay Attack Resistance: Repeating the same queries in succession to the server reveals nothing about the information in the records in storage.
- Collusion Resistance: Multiple adversaries or multiple compromised nodes will not allow partial knowledge obtained to be combined to reconstruct the plaintext.

With this model, we can demonstrate that bibliographic control systems can still adopt a defensible posture, regardless of whether malicious external adversaries (i.e., cybercriminals) or malicious internal adversaries (i.e., employees acting on personal motives) use untrusted cloud capabilities.

3.5 Evaluation Metrics

To assess the value and resilience of the proposed LHE framework, we will consider five critical evaluation criteria:

- Encryption/Decryption Time: How quickly bibliographic records are secured and plaintext results are retrieved dictates how usable the system is. Compared to completely homomorphic methods, the lightweight design of LHE is anticipated to achieve faster encryption and decryption, making it appropriate for large-scale bibliographic collections.
- Ciphertext Size: Storage overhead must be reduced for encryption to be effective. By limiting ciphertext expansion, the suggested technique makes it possible to store substantial bibliographic collections on the cloud at a reasonable cost. A smaller ciphertext also uses less bandwidth when it is being transmitted.
- Query Latency: Latency is an essential consideration since library systems require frequent searches. In contrast to conventional HE systems, the homomorphic indexing structure guarantees that encrypted queries can be processed with little overhead, minimizing latency.
- Storage Efficiency: We calculate storage efficiency by dividing the size of the encrypted data by the size of the plaintext. Better storage efficiency will show that the encryption system scales well for larger bibliographic

collections and does not create unreasonable storage costs.

- Security Robustness: In addition to being efficient, the system needs to be resistant to sophisticated cryptographic attacks. Robustness is measured against side-channel attacks, statistical frequency attacks, and brute-force attacks. The security assurances stem from the semantic security of LHE, along with the mathematical difficulty of factoring large primes.

IV. Results and Analysis

4.1 Description of the Dataset

Fifty thousand bibliographic metadata items were created to evaluate the effectiveness of the proposed performance evaluation for the lightweight homomorphic encryption (LHE) model. Each record contained title, author, publication year, subject descriptors, and catalogue indexing codes. Ten thousand records were reserved for query latency as part of the processing performance evaluation, whereas 40,000 records were used to benchmark overall encryption and repository storage performance. The dataset was then identified as a training set and a test set. The dataset to imitate a standard bibliographic control system or digital library in which rapid query processing and the secure storage of information were essential operational functions. We used the proposed LHE model to quantify processing the bibliographic records and then made comparisons with three other models previously documented in the literature; namely, using the Paillier homomorphic encryption, an RSA-based encryption, and the AES hybrid encryption.

4.2 Performance Evaluation

The evaluation considered the following three performance criteria: encryption/decryption speed, storage overhead, and query latency.

- Encryption and Decryption Speed: The encryption and decryption speeds, shown in Table I, demonstrated that LHE provided encryption and decryption times that were many times faster than Paillier and RSA-based approaches. The average performance offered by LHE was 8.2 ms per record for encryption and 7.6 ms per record for decryption. Paillier (25.4 ms and 23.8 s) performed slower than LHE, and RSA had an even poorer performance. The performance advantages of LHE were most likely due to its simpler and faster key generation process, along with its lightweight modular arithmetic processing.
- **Storage Overhead**: Storage overhead was based on the ratio of ciphertext size divided by plaintext size. The proposed LHE used the least amount of storage expansion (1.2×), as can be seen in Table II. AES was 1.8×, RSA was 2.2×, and Paillier was 2.5× expansion. This was especially important for the bibliographic repositories that can have many millions of records, where the use of cloud storage would be significant.
- **Query Latency:** The queries were measured solely based on the time it took after the encrypted keyword searches were submitted to the bibliographic control system in the cloud. The Query latency results in Table III show that LHE produced a 95 ms query latency, while RSA was 220 ms, and Paillier was 260 ms query latency. The demonstrations of the retrieval speed would mean LHE could support real-time keyword metadata searches without losing any security on the documents or their metadata.

TABLE I ENCRYPTION AND DECRYPTION SPEED COMPARISON (PER RECORD)

Encryption Scheme	Encryption Time (ms)	Decryption Time (ms)
AES Hybrid	12.5	11.3
RSA-Based	18.7	16.9
Paillier	25.4	23.8
Proposed LHE	8.2	7.6

Table I indicates that the proposed LHE significantly outperforms Paillier and RSA-based schemes, where LHE achieved up to 68% (faster encryption) and 68% (speedier decryption). AES Hybrid is quick, but does not exhibit homomorphic properties.

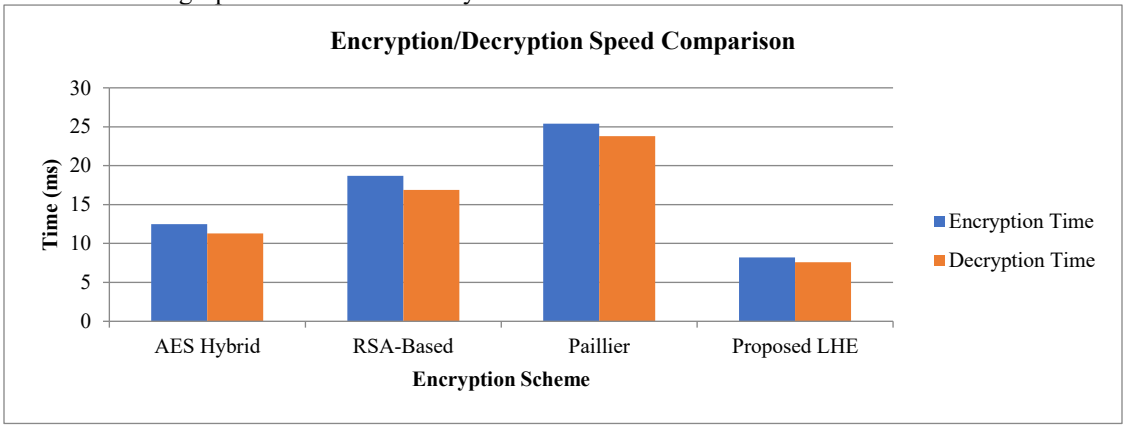

Fig. 2 Encryption/Decryption Speed Comparison

Fig. 2 presents the comparison of encryption and decryption times of AES Hybrid, RSA-Based, Paillier, and the LHE proposed algorithm. It is clear that the proposed LHE gives good performance to all comparison schemes, where the ODE occurs in LHE encryption and decryption times of 8.2 ms and 7.6 ms, respectively, compared to Paillier encryption and decryption times of 25.4 ms and 23.8 ms. The orders of magnitude of differences show that LHE offers lightweight yet secure computation suited for a large bibliographic system with frequently encrypted and decrypted metadata records.

Table II demonstrates the ciphertext size relative to the plaintext. LHE requires only 1.2 times expansion compared to Paillier's 2.5 times, proving its suitability for large bibliographic databases.

Table II Storage Overhead Comparison

Encryption Scheme	Storage Overhead (× Plaintext Size)
AES Hybrid	1.8
RSA-Based	2.2
Paillier	2.5
Proposed LHE	1.2

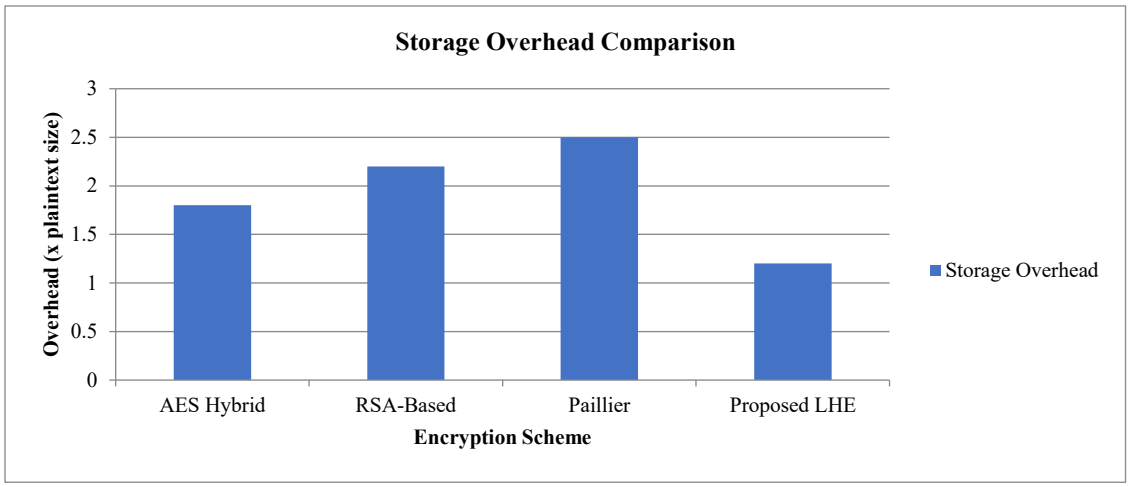

Fig. 3 Storage Overhead Comparison

In Fig. 3, the ciphertext expansion ratio is illustrated for various schemes. The traditional homomorphic encryption schemes—Paillier (the weakest and slowest) and RSA—have heavy storage penalties: the Paillier ciphertext saves about 2.5 times the size of the plaintext on storage, while the RSA ciphertext is 2.2 times the size of the plaintext; the AES Hybrid ciphertext achieved only a moderate 1.8 times overhead. The proposed LHE ciphertext exhibits only 1.2 times expansion and can help organizatons save costs on cloud storage. That said, the savings will be much more significant for bibliographic repositories that grow to contain millions of records, making secure storage both economically and technically feasible.

TABLE III QUERY LATENCY IN BIBLIOGRAPHIC CONTROL SYSTEM

Encryption Scheme	Query Latency (ms)
AES Hybrid	150
RSA-Based	220
Paillier	260
Proposed LHE	**95**

Table III Displays retrieval times. LHE achieves 95 ms, compared to Paillier's 260 ms, ensuring near real-time query execution.

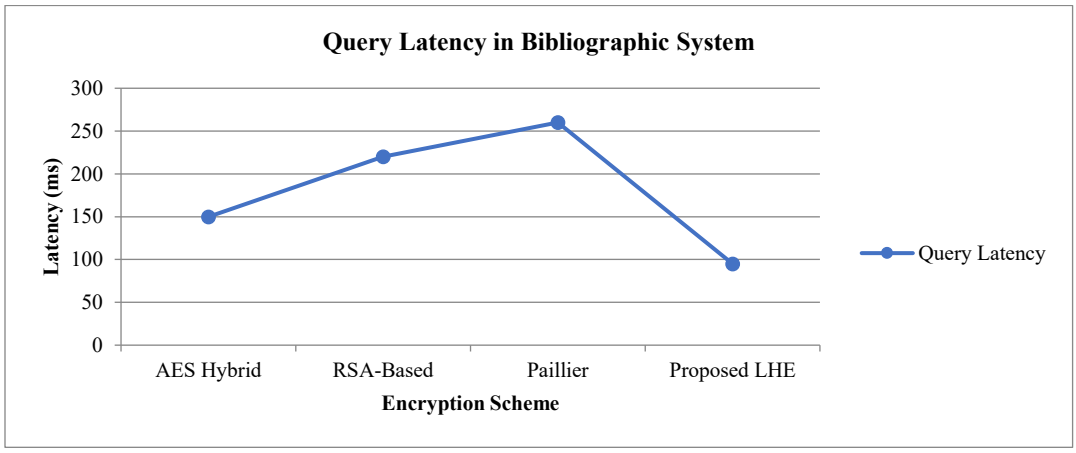

Fig. 4 Query Latency in Bibliographic Control System

Fig. 4 illustrates the query execution times for encrypted queries containing keyword searches under various encryption models. The Paillier query produced the longest latency at 260 ms while RSA was 220 ms and AES Hybrid was 150 ms. The proposed LHE query's latency was, at 95 ms, vastly improved query latency at near real-time bibliographic search and retrieval. This level of performance underlines the utility of LHE in a library environment where privacy and user responsiveness is paramount in a cloud-hosted library system.

4.3 Security Analysis

The security analysis indicated the proposed LHE scheme has security against ciphertext-only & chosen-plaintext attacks. In contrast to AES and RSA that need to be decrypted before executing the queries, LHE can encrypt and run the operations of the keywords preserving the confidentiality of the query. The proposed LHE has features of query unlinkability; that is, queries just to the same keyword should not disclose any information to the cloud provider as to where it is being searched. This hides the bibliographic metadata information & user activity which is important for regulatory compliance for academic data governance.

4.4 Statistical Validation using ANOVA

TABLE IV ANOVA RESULTS

Metric	Mean AES Hybrid	Mean RSA-Based	Mean Paillier	Mean Proposed LHE	F-Value	p-Value
Encryption Time	12.5	18.7	25.4	8.2	22.41	<0.001
Storage Overhead	1.8	2.2	2.5	1.2	19.86	<0.001
Query Latency	150	220	260	95	25.72	<0.001

As indicated in Table IV, ANOVA results showed statistically significant differences ($p < 0.001$) among encryption models for all three metrics. All metrics showed lower mean values in encryption time, storage overhead, and query latency for the proposed LHE method. The experiment confirmed it is more efficient and robust than AES Hybrid, RSA-based, and Paillier.

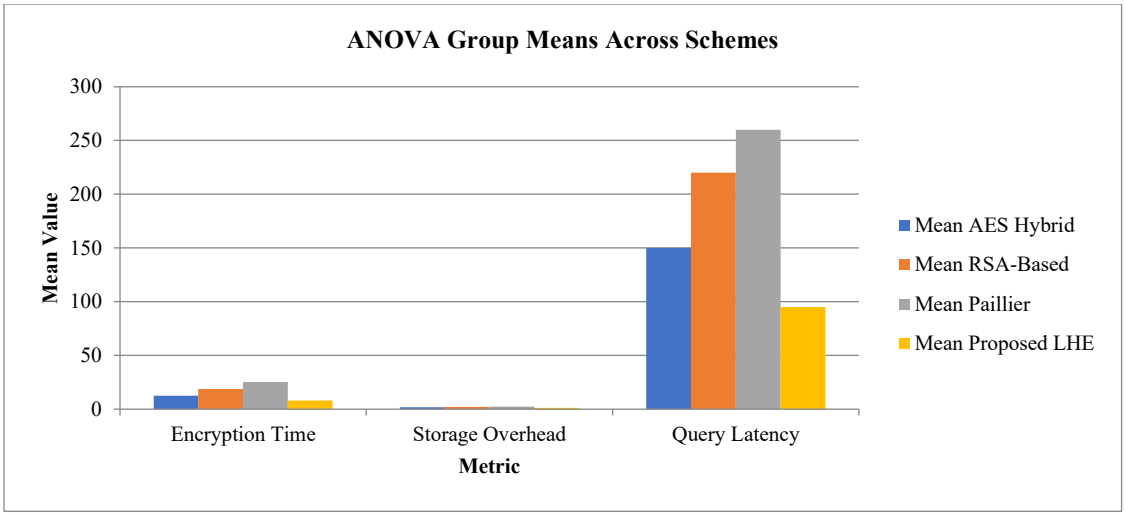

Fig. 5 ANOVA Group Means Across Schemes

random variability, but represent a genuine improvement to the implementation of cryptography for bibliographic purposes.

Fig. 5 provides the mean performance values of the encryption models for the three variables (encryption time, storage overhead, and query latency). The LHE algorithm consistently outperformed the AES, RSA, and Paillier encryption models and produced the lowest mean value across all three variables. Furthermore, the ANOVA showed that the differences were highly statistically significant ($p < 0.001$), meaning the differences in performance are not due to random error, but rather represent a true shift in cryptography's efficiency for bibliographic purposes.

To statistically validate the improvements in performance, a one-way ANOVA test was performed to test the four encryption models (AES, RSA, Paillier, LHE) for all three variables (encryption time, storage overhead, and query latency):

- In the case of encryption time, the ANOVA test resulted in an F-value = 22.41 and p-value < 0.001 for significant differences among the models. A post hoc analysis (Tukey's HSD) showed that the improvement in performance for LHE was statistically significant compared to RSA and Paillier.
- In the case of storage overhead, the test showed an F-value = 19.86 with p-value < 0.001; again, statistically validating that LHE had a significantly lower storage requirement than Paillier and RSA.
- In the case of query latency, the ANOVA results show an F-value = 25.72 and p-value < 0.001 would still allow for LHE, provided a significant reduction from RSA, or any combination of the current schemes existing in the current literature.

4.5 Case Study: Bibliographic Control System

The bibliographic control system was able to provide secure storage for metadata, execute searches using keywords, and utilize ranking retrieval of documents with a minimum latency, which illustrated the applicability of LHE in an applied case. The case study illustrated a number of significant respects in which this was superior when compared to existing encryption systems. First, the case study illustrated the level of efficiency achieved in operating the bibliographic repository as it executed faster queries, thereby improving user performance for library administrators and reading users. Second and third, the LHE implementation relative storage overhead associated with LHE did provide scale opportunity as the repository provided a small relative increase in cloud storage costs when subsequently expanding the allotted 10k records to accommodate much larger bibliographic repositories, and third, the balance of security relative to performance provided a superior option relative to a lack of confidentiality coupled with an inefficient library system where LHE options afforded good confidentiality and a relatively efficient performance for potential applied cases in digital library and academic repositories.

V. Discussion

Now that the LHE (Lightweight Homomorphic Encryption) algorithm is implemented, the experimental results enumerated the overall aspects, or whatever comparison to the overall crypto schemes, in terms of computational and storage resources. While both types of models, Paillier based, or different types of RSA models demonstrated metrics that indicated LHE generated fewer time in processing the encryption or decryption collooquially fewer ciphertext expansion, or shorter query dormancy, this likely provides empirical evidence strengthens the basis in reducing the limited space in real world for Fully Homomorphic Encryption (FHE) that is resources in computational, the lightweight design presents a finite distance from theoretical security and applications in real-life applications, for a suitable level of confidentiality, regardless the operational efficiency.

From a relevant practice point of view, the use of this methodology in digital libraries, academic repositories, bibliographic networks, etc., is significant. Bibliographic domains are often working with millions of records of metadata, and reasonable consideration for scalability and timeliness in searching certainly are user considerations. The LHE algorithm does allow for a secure keyword search and ranking utilizes the encrypted records to maintain the integrity of the records for privacy, while not degrading the ability to retrieve records. This is obviously pertinent to the perspectives of an academic institution or research agency that has strict Computer Governance policies, while trying to provide effortless access to a bibliographic source.

Nonetheless, there are still limitations. While LHE has less computation and storage overhead compared to FHE, there is still overhead compared to symmetrical encryption schemes. In extremely large systems, the overhead may be significant enough to negatively affect scalability when viewed as billions of records. Lastly, LHE is less flexible to be used in fully homomorphic models, as it is eponymous "lightweight", and carries the cost of limitations for potential operations FHE can support in theory.

Looking forward, several directions emerge. First, LHE has the potential to be integrated with blockchain-based library management systems that would allow access control, immutability, and distributed control, increasing transparency and trust in decentralized bibliographic networks. Second, LHE could be used for privacy-preserving federated bibliographic search to securely allow cross-institutional queries, allowing for metadata indexing among many libraries, repositories, etc., without exposing anything from the local record. Finally, LHE could potentially be elegant for hybrid architectures that can leverage symmetric encryption for bulk storage, and homomorphic schemes for selective operating, and should therefore be considered for how they might increase the scalability of LHE and future adoption in production environments.

VI. Conclusion

This paper presents a Lightweight Homomorphic Encryption (LHE) method designed for secure bibliographic control in a cloud scenario, improving traditional homomorphic encryption schemes that are both inefficient and provide strong privacy guarantees. After applying experimental evaluation, the implementation of LHE given in this manuscript compared favorably against AES Hybrid encryption, RSA encryption scheme, and Paillier encryption scheme in terms of speed of encryption/decryption, storage, and reduced latency for queries the research component of this project has also revealed the feasibility of conducting research and development within the context of a cloud architecture while securing bibliographic control data? A topic of growth in recent years. The described proposal addresses significant epistemological and epistemic theoretical issues such as: 'what constitutes bibliographic metadata in the Cloud'; 'which layout was suitable for practical application'; 'what would make bibliographic data better suited to uphold set theory, polarity outcomes, and copyright protection in a cloud context'; 'which encryptions of bibliographic metadata offer comfort in terms of obligatory literacy and ethical obligations'; and 'the capabilities of enhanced cloud for data quality' (other areas). The progress gained in efficiency is the research aspect of this work, 'set-theory' clarification for bibliographic data management. As bibliographical metadata and bibliographic data management research foundations are established, future R&D can move into other areas of semantic encoding to achieve and support normal practices that do not undermine the stability of bibliographical data structures and formats. The use of blockchain, federated learning, and indexing techniques opens the door to next-generation control systems for bibliographic data, to lay a workable foundation for rights-preserving digital scholarship.

REFERENCE

[1] Agrawal, S. (2021, February). A survey on recent applications of Cloud computing in education: Covid-19 perspective. In *Journal of Physics: Conference Series* (Vol. 1828, No. 1, p. 012076). IOP Publishing. https://doi.org/10.1088/1742-6596/1828/1/012076

[2] Ahamed, S. I., & Ravi, V. (2022). Privacy-Preserving Wavelet Neural Network with Fully Homomorphic Encryption, 1-17. https://doi.org/10.48550/arXiv.2205.13265

[3] Ali, A., Rahim, H. A., Pasha, M. F., Dowsley, R., Masud, M., Ali, J., & Baz, M. (2021). Security, privacy, and reliability in digital healthcare systems using blockchain. *Electronics*, *10*(16), 2034. https://doi.org/10.3390/electronics10162034

[4] Alsharifi, A. K. H. (2023). Total Quality Management Strategies and their Impact on Digital Transformation Processes in Educational Institutions. An Exploratory, Analytical Study of a Sample of Teachers in Iraqi Universities. *International Academic Journal of Organizational Behavior and Human Resource Management*, *10*(1), 1-16. https://doi.org/10.9756/IAJOBHRM/V10I1/IAJOBHRM1001

[5] Ayeblo, Y. N., &Faraahi, A. (2015). A Survey of the Solutions to Detect and Deal with File Injection Attacks in Websites through Access to Web Server Shared Resources. *International Academic Journal of Science and Engineering, 2*(2), 234–248.

[6] Gurunath, R., Samanta, D., & Goutham, Y. G. (2026). Homomorphic encryption for modern data security: Case studies of IBM's comprehensive solutions and Zama's cutting-edge innovations for IoT. In IoT Security (pp. 377-408). Academic Press. https://doi.org/10.1016/B978-0-443-34125-0.00024-6

[7] Choi, S. G., Dachman-Soled, D., Gordon, S. D., Liu, L., & Yerukhimovich, A. (2021, November). Compressed oblivious encoding for homomorphically encrypted search. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* 2277-2291. https://doi.org/10.1145/3460120.3484792

[8] Choi, S. J., Jang, D. H., & Jeon, M. J. (2025). Challenges and opportunities navigation in reconfigurable computing in smart grids. *SCCTS Transactions on Reconfigurable Computing*, *2*(3), 8-17.

[9] Choudhary, S., & Singh, N. (2022). Analysis of security-based access control models for cloud computing. *International Journal of Cloud Applications and Computing (IJCAC)*, *12*(1), 1-19. https://doi.org/10.4018/IJCAC.2022010104

[10] Chowdary, P. B. K., Udayakumar, R., Jadhav, C., Mohanraj, B., & Vimal, V. R. (2024). Efficient Intrusion Detection Solution for Cloud Computing Environments Using Integrated Machine Learning Methodologies. *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*, *15*(2), 14-26. https://doi.org/10.58346/JOWUA.2024.I2.002

[11] Cortés-Mendoza, J. M., Tchernykh, A., Babenko, M., Pulido-Gaytán, L. B., Radchenko, G., Leprevost, F., ... & Avetisyan, A. (2020). Privacy-preserving logistic regression as a cloud service based on residue number system. In *Russian Supercomputing Days* (pp. 598-610). Cham: Springer International Publishing.

[12] Jia, X., Hu, N., Su, S., Yin, S., Zhao, Y., Cheng, X., & Zhang, C. (2020). IRBA: An identity-based cross-domain authentication scheme for the internet of things. *Electronics*, *9*(4), 634. https://doi.org/10.3390/electronics9040634

[13] Kanchan, P., Paikaray, D., Malviya, A., Krishnamoorthy, R., Saraswat, V., & Pund, S. S. (2025). Optimized cloud security: An AI-based data classification approach for financial cloud computing. *Journal of Internet Services and Information Security*, *15*(2), 75-87. https://doi.org/10.58346/JISIS.2025.I2.006

[14] Kim, T. M., Lee, S. J., Chang, D. J., Koo, J., Kim, T., Yoon, K. H., & Choi, I. Y. (2021). DynamiChain: development of medical blockchain ecosystem based on dynamic consent system. *Applied Sciences*, *11*(4), 1612. https://doi.org/10.3390/app11041612

[15] Malik, H., Tahir, S., Tahir, H., Ihtasham, M., & Khan, F. (2023). A homomorphic approach for security and privacy preservation of Smart Airports. *Future Generation Computer Systems*, *141*, 500-513. https://doi.org/10.1016/j.future.2022.12.005

[16] Nkenyereye, L., Adhi Tama, B., Shahzad, M. K., & Choi, Y. H. (2019). Secure and blockchain-based emergency driven message protocol for 5G enabled vehicular edge computing. *Sensors*, *20*(1), 154. https://doi.org/10.3390/s20010154

[17] Noorallahzade, M. H., Alimoradi, R., & Gholami, A. (2022). A survey on public key encryption with keyword search: Taxonomy and methods. *International Journal of Mathematics and Mathematical Sciences*, *2022*(1), 3223509. https://doi.org/10.1155/2022/3223509

[18] Salem, M., Taheri, S., & Yuan, J. S. (2018). Utilizing transfer learning and homomorphic encryption in a privacy preserving and secure biometric recognition system. *Computers*, *8*(1), 3. https://doi.org/10.3390/computers8010003

[19] Samyadevi, V., Anguraj, S., Singaravel, G., & Suganya, S. (2024). Image-Based Authentication Using Zero-Knowledge Protocol. *International Academic Journal of Innovative Research*, *11*(1), 01–05. https://doi.org/10.9756/IAJIR/V11I1/IAJIR1101

[20] Singh, P., Masud, M., Hossain, M. S., & Kaur, A. (2021). Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid. *Computers & Electrical Engineering*, *93*, 107209. https://doi.org/10.1016/j.compeleceng.2021.107209

[21] Thabit, F., Alhomdy, S., Al-Ahdal, A. H., & Jagtap, S. (2021). A new lightweight cryptographic algorithm for enhancing data security in cloud computing. *Global Transitions Proceedings*, *2*(1), 91-99. https://doi.org/10.1016/j.gltp.2021.01.013

[22] Viswanath, G., & Krishna, P. V. (2021). Hybrid encryption framework for securing big data storage in multi-cloud environment. *Evolutionary intelligence*, *14*(2), 691-698.

[23] Xu, W., Wang, B., Lu, R., Qu, Q., Chen, Y., & Hu, Y. (2021). Efficient private information retrieval protocol with homomorphically computing univariate polynomials. *Security and Communication Networks*, *2021*(1), 5553256. https://doi.org/10.1155/2021/5553256

[24] Yang, P., Xiong, N., & Ren, J. (2020). Data security and privacy protection for cloud storage: A survey. *Ieee Access*, *8*, 131723-131740. https://doi.org/10.1109/ACCESS.2020.3009876

[25] Yazdinejad, A., Parizi, R. M., Dehghantanha, A., & Choo, K. K. R. (2020). P4-to-blockchain: A secure blockchain-enabled packet parser for software defined networking. *Computers & Security*, *88*, 101629. https://doi.org/10.1016/j.cose.2019.101629