

A combined framework for data classification on data confidentiality levels for hybrid cloud storage using hybrid chaotic maps

Bharath Kumar Rama^{1*}, Dr. S. Thaiyalnayaki²

¹Research Scholar, Dept. of CSE, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

²Associate Professor, CSE, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India

Abstract

Cloud storage plays a pivotal role in modern data management yet requires robust security solutions to defend against potential vulnerabilities. Researchers behind this study focus on solving encryption challenges in hybrid cloud storage through adaptable encryption techniques due to the need for customized protection of various data confidentiality levels. The research introduces an integrated framework that uses hybrid chaotic maps to encrypt data according to a classification system designed for hybrid cloud storage. The system establishes different confidentiality ratings of Low, medium, and high by analyzing entropy as well as access frequency data. Security and efficiency are achieved through the implementation of three chaotic maps, including Logistic and Tent and Hybrid Map, which produce encryption keys for different levels. The proposed framework demonstrates superior performance according to an analysis that compares it with Blowfish and ECC algorithms for encryption/decryption times as well as entropy and energy use and throughput. The Python-based framework delivers efficient and secure encryption services while maintaining strong randomness capabilities, which make it appropriate for hybrid cloud scalability needs.

Keywords: Chaotic maps, cloud computing, data classification, Logistic map, Tent map

1. Introduction

As cloud computing concerns scenarios, it stands out. It offers a wide range of services—unlimited database storage, networks and communications, and more—that are available from anywhere, regardless of the location or medium. Because of its convenience, people are using the cloud more and more, which is producing a lot of data and raising privacy and security concerns. Users' deliberate or unintentional actions in the cloud have the potential to cause serious problems with cloud services, most notably with data security and data breaches. Unauthorized and unverified sources should therefore be subject to data accessibility limitations. Devices could potentially lead to data breaches because they allow users to reuse APIs and data. Hybrid cloud storage solutions are gaining popularity in modern digital times because of growing data volumes since they integrate public and private cloud infrastructure systems. Increased cyber threats underline the critical need to protect data security despite the widespread adoption of hybrid cloud solutions. Dealing with data of varying sensitivity levels often compromises the security, computational efficiency, and flexibility of traditional encryption approaches. Cryptographic methods primarily utilize various keys to encrypt and decrypt data on the cloud [1].

The foundation of the Advanced Encryption Standard (AES) relies on the symmetric key encryption technique, which replaced the Data Encryption Standard (DES) [2]. AES's 128-bit key size makes it much faster than DES, which has a 64-bit key size. So, asymmetric key encryption is the basis of Elliptic Curve Cryptography (ECC) [3]. ECC is better than other cryptographic methods because it provides better security with a smaller key size than other systems in use today. A hybrid cloud system stores public data with private company records alongside other secured information that requires different levels of confidentiality. Risk arises from the present encryption algorithm implementation because it protects all data at one level which causes resource inefficiencies and slows system performance. Furthermore, the dynamic nature of hybrid cloud storage makes it challenging to deploy effective encryption techniques tailored to data sensitivity. The pseudo-random features of

existing chaotic map-based encryption solutions make them attractive, but they aren't always scalable or adaptable enough for use in hybrid cloud settings.

Chaos-based systems have done better than traditional encryption methods [4–7] because their use of changeable keys has been shown to improve security and privacy. Using a variety of chaotic components to enhance a scheme's confusion and dispersion abilities is the foundation [8]. For secure data exchange, especially in lightweight and resource-constrained situations, recent research has shown that chaotic map-based encryption is effective [9]. Nevertheless, very little is known about how to use them for hybrid cloud storage. Based on previous work on chaotic systems [10], this paradigm takes into account the specific difficulties of working in a hybrid cloud setting. In response to these issues, a new paradigm for hybrid cloud security makes use of encryption with hybrid chaos maps and classifies data according to confidentiality level [11].

This article discusses about a unified framework for classifying and encrypting data in hybrid cloud storage. The article also tackles some of the previously listed issues and limitations.

(i) Construct a module that uses metadata and content sensitivity analysis to classify data into predetermined degrees of confidentiality (e.g., low, medium, high);

(ii) This technique combines logistic maps, tent maps, and hybrid chaotic maps to construct encryption keys;

(iii) reduces the burden of encryption by using lightweight methods for data that are not very sensitive and saving resource-intensive encryption for data that is very sensitive;

(iv) Evaluate the suggested framework using performance measures such as energy use and encryption time along with security measures such as entropy and key sensitivity.

2. Related works

Cloud storage has gained popularity because it permits users instantly exchange resources with each other. The availability of cloud storage at any time makes data owners prefer it over alternative solutions. It is essential to validate data preservation and data integrity since the system needs to remain secure.

In [12], the authors provided a strong defense against cyber-attacks by presenting integrated homomorphic encryption (FHE) and secure hash algorithm 3 (SHA-3). FHE and SHA-3 may resolve this issue. Factors such as computational overheads, complexity requirements, key management challenges, and limited support from vendors or providers offering specialized hardware and software to support complex mathematical functions were considered in their method before it was combined into a hybrid form. This form is suitable for scaling horizontally across different nodes within clouds, which is used by end users globally for mission-critical workloads running inside virtualized machines on public and private clouds. The findings demonstrate that when key infrastructure components are operated in cloud settings, an organization's cybersecurity posture may be greatly enhanced by combining FHE with SHA-3. It outperforms other approaches that are presently used for complicated encryption and decoding, including AES and DSA, homomorphic encryption and RSA, and AES and RSA, according to the findings. Our suggested hybrid technique uses completely homomorphic & SHA3 with 2048 bits, takes 0.789 seconds to encrypt and 0.001 seconds to decode, and takes 4.25 seconds to execute.

In [13], the authors proposed a chaotic theory-based encryption and compression method to ensure the privacy and integrity of data while it is sent or stored on the cloud in bulk. All of the data is compressed before transmission by using an adaptive Huffman encoding technique and a chaotic substitution box. As a result, the original data is replaced with an efficient, unreadable version of the input data, making it harder for an immoral person or organization to understand its actual meaning. Compared to earlier methods, the suggested chaotic approach offers better privacy, as shown by its highest entropy value of 7.99 in the assessment findings. With the best bits per code of 4.41, a throughput of 2.89 MB/s, and a low information loss

percentage of 0.0011%, it performs better in terms of space and time savings while also enhancing data integrity.

In [14], the authors developed a hybrid encryption algorithm that combines the advantages of symmetric and asymmetric encryption to improve data security in cloud computing settings. Asymmetric encryption is used for safe key exchange in the proposed hybrid technique, while symmetric encryption is used for efficient and quick data encryption. This two-layer encryption method may greatly enhance the security and privacy of transmitted and stored data. The method employs strong key management procedures to further safeguard encryption keys from unauthorized access. They proved that the hybrid encryption technique is appropriate for real-time applications thanks to its excellent security and acceptable performance levels with a battery of tests measuring both. According to the findings, this method successfully reduces the chances of data breaches and illegal access, making it a viable candidate for improving cloud computing data security.

In [15], the authors presented a hybrid diffusion-based encryption system to ensure the safety of stored visual images in the cloud. To create the permutation picture, the original image is first split into many equal parts, and then each part's column and row are sorted and moved accordingly. Next, they decompose the permutation image of each sub-block to create the high- and low-bit planes. Weighted diffusion and sum-preserving diffusion are applied to both bit planes for diffusion before creating an encrypted visual image. The method provides exceptional correlation protection for encrypted visuals because it operates effectively. The visual appeal of encrypted images together with sub-bit plane diffusion in each sub-block and minimal picture expansion rate makes this possible. The visual image encryption system is an excellent option for securing image data in cloud storage environments, as shown by the experimental assessment of its better performance.

In [16], the authors presented a hybrid encryption approach that uses both AES and ECC. Unfortunately, the high computational cost of AES, a symmetric encryption technology with enormous key sizes, offsets its strong data security. However, ECC protects sensitive data with smaller keys via an asymmetric encryption mechanism. The suggested model achieves optimal computing efficiency and excellent security by combining these two approaches. Since the model uses ECC to produce AES encryption keys, it reduces key size and improves encryption performance. By integrating the best features of both methods, this hybrid approach guarantees data integrity and prevents unauthorized access. Traditional algorithms like DES and RSA aren't as good as the hybrid AES-ECC model when it comes to encryption speed, memory optimization, and the avalanche effect, which shows how resistant the system is to changes in input.

3. Materials and Methods

The representation of hybrid cloud storage presents three primary components in Fig.1 as follows: (i) The data classification module uses Random Forest and features including entropy and access frequency alongside file type to categorize information into low, medium, and high confidentiality levels. The encryption key generator depends on confidentiality levels through its (ii) Chaotic key generation module, which uses Logistic maps for low-key security, Tent maps for medium-key security, and hybrid maps for high-key requirements. Security maps provide systems with three vital features, including random numbers, initial state dependency, and maximum entropy, which create excellent security results. Data encryption occurs through a module with operations designed to match confidentiality requirements. The data encryption scheme employs lightweight XOR encryption at low-level data and substitution-based encryption at a medium level while using hybrid block encryption at high-level data to achieve efficiency and scalability in hybrid cloud systems.

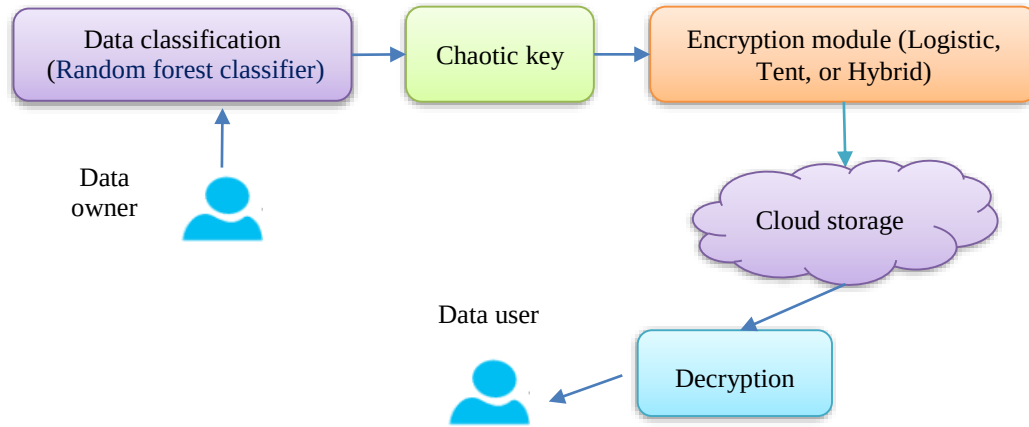


Fig. 1. The proposed framework for hybrid cloud storage

3.1 Data classification:

The RFC collects data classifications through ensemble decision trees, which produce aggregated outputs to determine data confidentiality status (Low, Medium, High). The model splits data through tree assignments that use the entropy (E) variable along with the file type (F_t) and access frequency (A_f) criteria for detecting sensitive data.

$$D = \{(X_1, y_1), (X_2, y_2), \dots, (X_n, y_n)\} \quad (1)$$

$X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ are feature vectors of m attributes for the i -th data point.

$y_i \in \{\text{Low, Medium, High}\}$ is the confidentiality label for X_i .

The process of splitting relies on achieving optimum information gain, which computes the entropy decrease when separating nodes. The entropy quantification method (H) identifies the most informative split points because it calculates data randomness levels. Each node of the split decision trees selects a random set of features. This process keeps going until it reaches a certain point, like maximum depth or when all the leaf nodes are empty. The selected feature x_j at nodes comes from maximizing the information gain, which measures $H(P)$, the entropy of the parent node P .

$$IG = H(P) - \sum_{k=1}^K \frac{|P_k|}{|P|} H(P_k) \quad (2)$$

$$H(P) = - \sum_{c \in C} p(c) \log_2 p(c)$$

where P_k indicates child nodes after the split, $p(c)$ represents proportion of data in class c at node P , and C indicates set of confidentiality levels {Low, Medium, High}. The RFC aggregates predictions from N decision trees:

$$y_{final} = mode(y_1, y_2, \dots, y_N) \quad (3)$$

where y_i is the predicted confidentiality level from the i -th tree. The RFC employs bootstrap sampling, where each tree is trained on a random subset of the training data. Predictions are aggregated by majority voting or probability averaging to classify data.

$$P(y = c|X) = \frac{1}{N} \sum_{i=1}^N P_i(y = c|X) \quad (4)$$

where $P_i(y = c|X)$ is the probability assigned by the i -th tree. Based on probabilities:

$$y_{final} = \arg \max_{c \in \mathcal{C}} P(y = c|X) \quad (5)$$

Chaotic Key Generator: The CKG depends on chaotic map characteristics for producing encryption keys that exhibit both high random patterns and sensitivity levels. Key generation begins with the selection of two crucial parameters known as control variable (r) alongside seed (x_0). Initializations of these variables prove highly vulnerable to any modification. The chaotic maps include: The expression for the Logistic Map appears like this:

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n), \quad (6)$$

The state x_n exists between 0 and 1 at step n under the chaos-controlling parameter r which ranges between 3.57 and 4.0. A seed x_0 is selected from the (0,1) interval together with a control parameter r function as input. At step $n+1$ the computation of the following value starts from the existing value from step n .

High-speed processing, coupled with its easy structure, allows the encryption of data with minimal sensitivity, such as shared logs and public records. A chaotic sequence goes through a transformation process which creates encryption keys.

The Tent Map, given by:

$$x_{n+1} = \begin{cases} \mu \cdot x_n, & \text{if } x_n < 0.5, \\ \mu \cdot (1 - x_n), & \text{if } x_n \geq 0.5, \end{cases} \quad (7)$$

The key sensitivity along with higher randomness emerge from a μ value set to 2 in the comparison to the Logistic Map. For values of x_n less than 0.5 the mapping process expands the value through $x_{n+1} = \mu \cdot x_n$. If $x_n \geq 0.5$, the map flips and compresses the value as $x_{n+1} = \mu \cdot (1 - x_n)$.

$K_n = [10^5 \cdot x_n] \bmod 256$ to produce 8-bit keys.

$$x_{n+1} = \alpha \cdot \text{Logistic}(x_n) + (1 - \alpha) \cdot \text{Tent}(x_n) \quad (8)$$

X_0 should belong to the interval (0,1) when selecting parameters r for the Logistic Map and μ for the Tent Map. The selection of α should be between 0 and 1. The encryption process requires this map to protect vital items such as medical files or financial records by maintaining high entropy against cryptanalysis.

Encryption Module: Data security within the module depends on its classification through encryption keys derived from chaotic maps which include Logistic, Tent, and Hybrid methods. Security measures and processing speed result from confidentiality-level decisions that establish encryption methods.

Low Confidentiality (Logistic Map): Data is encrypted using XOR operations with keys generated by the Logistic Map

$$C_i = P_i \oplus K_i \quad (9)$$

where P_i is the plaintext bit, K_i is the key, and C_i is the ciphertext.

Medium Confidentiality (Tent Map): Combines XOR with substitution. Keys are used to index a substitution box (S):

$$C_i = S[K_i \oplus P_i] \quad (10)$$

High Confidentiality (Hybrid Map): Uses block encryption with hybrid keys. For each block j :

$$C_j = \text{substitution}(P_j, K_j) \oplus K_j \quad (11)$$

Every operation maintains sensitivity towards chaotic encryption keys, which give both high randomness values and powerful cryptographic properties. Data security reaches its maximum strength while maintaining minimal processing requirements for unimportant information within the module framework.

Decryption Module: A receiver generates chaotic keys through the identical application of initial conditions together with parameters. The process for plaintext retrieval requires both XOR and substitution inversion operations. The proposed algorithm performs classification through analysis of entropy and access frequency features, which results in low, medium, or high confidentiality tiers. The key generator employs chaotic mapping to choose encryption keys from Logistic Maps for Low confidentiality, followed by Tent maps for medium-level keys and then selects the High key through Hybrid maps. The encryption methods for Low, medium and high tiers employ special purpose designs where Low uses XOR and medium implements substitution, and high uses hybrid encryption, thus providing scalable security. The key-generation process involves continuous chaotic map applications until 8-bit encryption values are achieved. The process of data encryption leads to secure data transmission and storage. The pseudocode within Algorithm 1 describes how features are extracted and then classified before keys are generated for encryption alongside secure storage processes.

Algorithm 1: Combined framework for Data Classification and Encryption

Input: Data D

Output: Encrypted data C

Input Data: Receive raw data D for classification and encryption.

Data Classification:

Analyze features like entropy (E), access frequency (A), and file type (F).

Classify D into L, M, H (Low, Medium, High confidentiality levels).

Chaotic Key Generation:

For L : Use Logistic Map $x_{n+1} = r \cdot x_n \cdot (1 - x_n)$.

For M : Use Tent Map $x_{n+1} = \mu \cdot x_n \cdot (1 - x_n)$ if $x_n < 0.5$, else $\mu \cdot (1 - x_n)$.

if $x_n < 0.5$, the map expands the value as $x_{n+1} = \mu \cdot x_n$. If $x_n \geq 0.5$

For H : Combine Logistic and Tent Maps $x_{n+1} = \alpha \cdot \text{Logistic}(x_n) + (1 - \alpha) \cdot \text{Tent}(x_n)$

Extract keys from x_n : $K_n = \lfloor 10^5 \cdot x_n \rfloor \bmod 256$

Encryption:

For L : Use XOR Encryption: $C_i = P_i \oplus K_i$.

For M : Use Substitution Encryption: $C_i = S[K_i \oplus P_i]$.

For H : Use Block-Based Hybrid Encryption: $C_j = \text{substitution}(P_j, K_j) \oplus K_j$.

Secure Transmission/Storage: Store or transmit C securely.

4. Results and Discussion

The results show that the proposed chaotic map-based system delivers superior performance compared to the Blowfish algorithm and ECC protocol through detailed analysis of encryption and decryption speed, entropy strength system, energy utilization, and data processing speed. The proposed framework provides superior performance characteristics, including rapid execution speed alongside high entropy values and reduced energy requirements, together with high throughput, which makes it a suitable choice for hybrid cloud storage. The experiment uses simulated confidentiality levels ranging from low to medium and high, during which standard hybrid cloud environment datasets undergo metric assessment. The application utilized the Python programming language that integrated NumPy together with the pandas and Matplotlib libraries to handle data processing and draw chaotic maps along with generating visual representations. This framework demonstrates flexibility for different confidentiality levels while maintaining a lightweight design structure, which contributes to enhanced real-time cloud security together with future application scalability.

A necessary set of security protocols gets deployed by the confidentiality-based data classification model after it conducts data sensitivity level categorization. The success rate of protecting against unauthorized access determines the effectiveness of implemented protective measures. Table 1 displays an analysis of various encryption techniques where it shows how the proposed chaotic maps-based framework measures up to Blowfish and ECC for Low-Medium-High confidentiality levels according to performance results. The table compares encryption time, decryption time, and entropy. The proposed framework achieves enhanced efficiency by delivering swift encryption and decryption performance at every security level, particularly during high confidentiality tests, which result in increased times for both Blowfish and ECC because of their increased computational demands. The proposed framework delivers superior security through elevated entropy metrics, which reflect increased randomization capabilities. Blowfish operates at a speed surpassing ECC, although it creates weaker entropy, thereby becoming inadequate for safeguarding highly confidential data. The proposed chaotic map approach establishes itself as the most suitable solution for securing hybrid cloud storage security.

Table 1 Comparison of different encryption techniques for the performance metrics

Confidentiality Level	Algorithm	Encryption Time (s)	Decryption Time (s)	Entropy
Low	Proposed (Chaotic Maps)	0.2	0.18	0.95
Medium		0.5	0.48	0.97
High		0.9	0.85	0.98
Low	Blowfish	0.4	0.38	0.85
Medium		0.7	0.68	0.87
High		1.1	1.05	0.90
Low	ECC	0.6	0.58	0.90
Medium		0.9	0.88	0.92
High		1.5	1.4	0.93

Three encryption algorithms are compared for their time requirements, including a confidentiality level analysis in Fig.2, which shows how the proposed chaotic maps methodology outperforms both Blowfish and ECC methods. All steps of encryption run faster through the proposed framework due to the simple characteristics of chaotic maps, particularly at low confidentiality levels when using the logistic map. The encryption processing of Blowfish takes an average amount of time but becomes slower when applying high confidentiality levels, attributed to its block cipher encryption method. ECC requires the most time to complete its process at every security level, especially during high confidentiality operations, as elliptic curve operations demand extended processing time. The proposed framework exhibits exceptional speed, enabling it to support real-time scalability for encryption requirements.

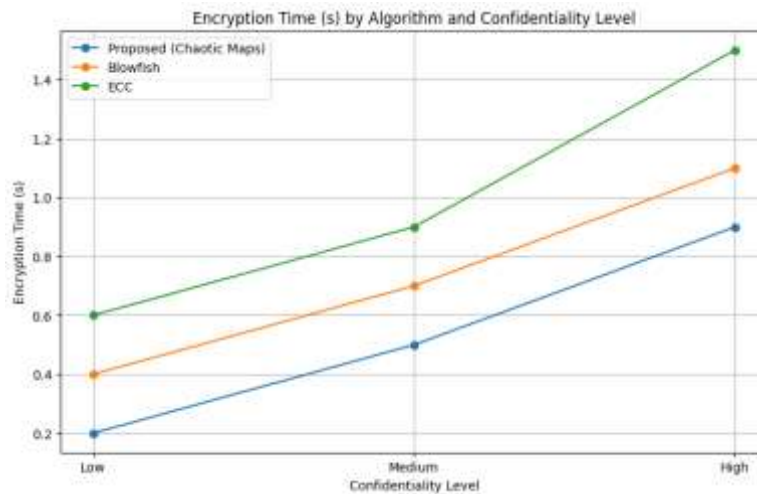


Fig. 2 Encryption time comparison of three algorithms

The decryption time analysis presented in Fig.3 evaluates the performance of the proposed chaotic maps framework against Blowfish algorithms and ECC at various confidentiality levels. The proposed decryption framework achieves the fastest decryption times at all operational levels due to its use of simple chaotic map-based key regeneration and lightweight cryptographic operations. While Blowfish offers slower encryption speeds because of its structured block design, the elliptic curve algorithm exhibits the longest execution times, particularly at high confidentiality levels, owing to its complex elliptic curve procedures and key management requirements.

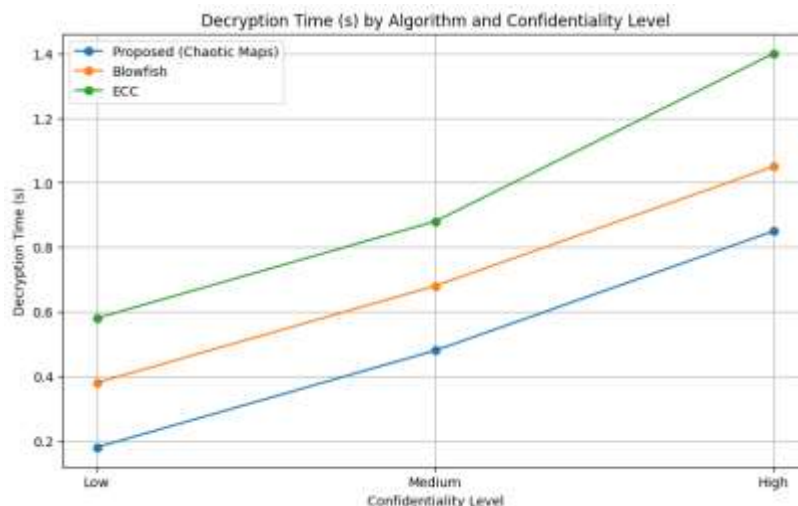


Fig. 3 Decryption time comparison of three algorithms

The entropy data in Fig.4 makes it clear how random and safe the key generation is with the proposed chaotic maps, Blowfish, and ECC algorithms. The suggested framework has the highest entropy scores across all levels of privacy, showing the highest levels of randomness along with excellent security. At high confidentiality, the system reaches its peak unpredictability through its implementation of dynamic hybrid chaotic maps between logistic and tent maps. ECC's strong cryptographic system prevents easy changes, resulting in an about-average entropy level. The entropy level of Blowfish reaches its minimum because its structural blocks remain rigid.

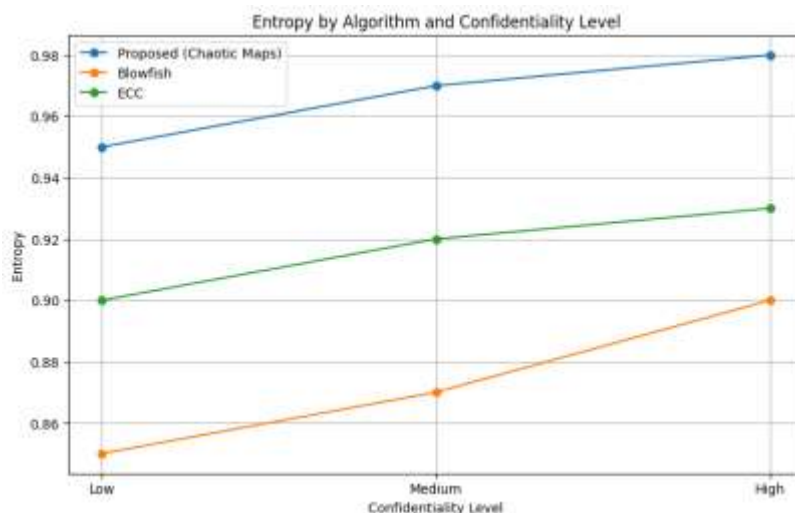


Fig.4 Entropy comparison of three algorithms

Fig.5 shows how much energy the proposed chaotic maps framework, the Blowfish, and the ECC algorithms use compared to the level of privacy they provide. At every level of privacy, the proposed framework uses the least amount of energy because chaotic maps keep their structure light and their key generation methods simple. The energy intake of Blowfish remains average yet grows more substantial at medium and high confidentiality because it applies block cipher algorithms. It takes a lot of computing power to do elliptic curve cryptographic operations, so ECC is the algorithm that uses the most power. This is especially true at the high confidentiality level. Due to its energy-efficient features, the proposed framework proves suitable for implementation in resource-limited hybrid cloud systems.

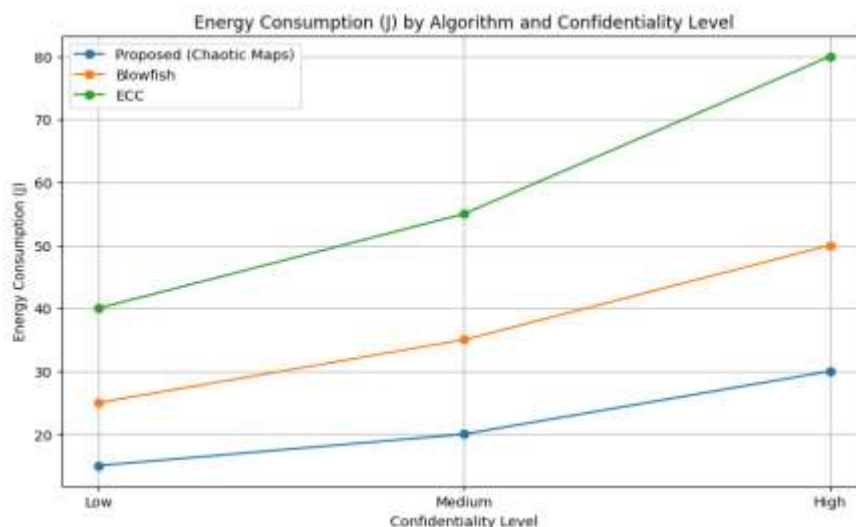


Fig.5 Energy consumption comparison of three algorithms

The data processing efficiency evaluation of the proposed chaotic maps framework as well as Blowfish encryption and ECC algorithms is displayed in Fig.6 regarding throughput by algorithm and confidentiality level. The proposed framework demonstrates the best data processing speed at every security level because it implements simple encryption technologies and fast chaotic key generation mechanisms. Because of its block-based operations, Blowfish exhibits moderate throughput, but the performance declines as confidentiality levels increase. The implementation of ECC produces the lowest data transfer speed, especially when security requirements are at the highest level because elliptic curve cryptography requires significant processing power.

(1)

5. Conclusions

The proposed framework based on chaotic maps offers an efficient protection system for hybrid cloud data storage applications. The classification approach, along with algorithm-specific encryption, enables the system to deliver encryption and decryption processing at a heightened rate with elevated randomness, minimal energy waste, and superior throughput compared to standard Blowfish and ECC implementations. The implementation of Logistic, Tent and Hybrid Maps ensures flexible security measures that enhance both performance levels and security features. The Python-based framework shows both scalability and practical usage abilities as it serves resource-low environments. The tested framework proves effective as a simple yet secure and flexible solution to deal with current cloud storage problems. Future work will expand the framework to provide real-time encryption and dynamic classification services for streaming data across hybrid cloud deployments. The security of systems can be boosted through additional research integrating quantum-inspired encryption techniques into advanced chaotic systems.

References

- [1] Shukla, D.K.; Dwivedi, V.K.; Trivedi, M.C. Encryption algorithm in cloud computing. *Mater. Today Proc.* 2020, 37, 1869–1875.
- [2] Yahia, H.S.; Zeebaree, S.R.M.; Sadeeq, M.A.M.; Salim, N.O.M.; Kak, S.F.; Al-Zebari, A.; Salih, A.A.; Hussein, H.A. Comprehensive survey for cloud computing based nature-inspired algorithms optimization scheduling. *Asian J. Res. Comput. Sci.* 2021, 1–16.
- [3] Qazi, R.; Khan, I.A. Data security in cloud computing using elliptic curve cryptography. *Int. J. Comput. Commun. Netw.* 2019, 1, 46–52.
- [4] Hasheminejad, A.; Rostami, M. A novel bit level multiphase algorithm for image encryption based on PWLCM chaotic map. *Optik* 2019, 184, 205–213.
- [5] Yu, J.; Guo, S.; Song, X.; Xie, Y.; Wang, E. Image parallel encryption technology based on sequence generator and chaotic measurement matrix. *Entropy* 2020, 22, 76.
- [6] Yousif, B.; Khalifa, F.; Makram, A.; Takieldein, A. A novel image encryption/decryption scheme based on integrating multiple chaotic maps. *AIP Adv.* 2020, 10, 075220.
- [7] Murillo-Escobar, M.; Meranza-Castillón, M.; López-Gutiérrez, R.; Cruz-Hernández, C. A Chaotic Encryption Algorithm for Image Privacy Based on Two Pseudorandomly Enhanced Logistic Maps. In *Multimedia Security Using Chaotic Maps: Principles and Methodologies*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 111–136.
- [8] Yasser, I.; Khalifa, F.; Mohamed, M.A.; Samrah, A.S. A New Image Encryption Scheme Based on Hybrid Chaotic Maps. *Complexity* 2020, 2020, 9597619.
- [9] Zhou, X., Wang, Q., & Liu, J. (2023). Lightweight chaotic encryption for IoT data protection. *IEEE Transactions on Cloud Computing*, 10(2), 345-357. <https://doi.org/10.1109/TCC.2023.112345>
- [10] Chen, H., Zhang, L., & Li, Y. (2024). Chaotic map-based encryption techniques for secure cloud storage: A survey. *Journal of Information Security*, 47(1), 112-128. <https://doi.org/10.1016/j.jinfosec.2024.01.005>
- [11] Rehman, A., & Kumar, S. (2024). Data confidentiality in hybrid cloud systems: Challenges and solutions. *Cloud Computing Research*, 12(3), 233-246. <https://doi.org/10.1016/j.cloud.2024.03.011>
- [12] Pothered, S., Peddisetty, N., Yellamma, P., Botta, G., & Gottipati, K. N. (2024). Data Security in Cloud Environment by Using Hybrid Encryption Technique: A Comprehensive Study on Enhancing Confidentiality and Reliability. *International Journal of Intelligent Engineering and Systems*, 17(2), 159–170. <https://doi.org/10.22266/ijies2024.0430.14>
- [13] Bhattacharjee, S., Sharma, H., Choudhury, T. et al. Leveraging chaos for enhancing encryption and compression in large cloud data transfers. *J Supercomput* 80, 11923–11957 (2024). <https://doi.org/10.1007/s11227-024-05906-3>.
- [14] Abdulkareem, Name & Idjighere, Oghenerukevwe. (2024). Improving Data Security in Cloud Computing through the Implementation of a Hybrid Encryption Algorithm. *Mikailalsys Journal of Advanced Engineering International*. 1. 213-228. 10.58578/mjaei.v1i3.4034.
- [15] Zhang, Yan & Tong, Yaonan & Li, Chunlai & Peng, Yuexi & Tan, Fei. (2024). Hybrid diffusion-based visual image encryption for secure cloud storage. *Nonlinear Dynamics*. 1-18. 10.1007/s11071-024-10446-9.
- [16] Rehman, S., Bajwa, N. T., Shah, M. A., Aseeri, A. O., & Anjum, A. (2021). Hybrid AES-ECC Model for the Security of Data over Cloud Storage. *Electronics*, 10(21), 2673. <https://doi.org/10.3390/electronics10212673>