National College of Ireland

Image Security in Cloud using hybrid Compression and Encryption Technique

MSc Research Project
MSCLOUD

Nikhil Rajendra Puranik
Student ID: X22194771

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Nikhil Rajendra Puranik
Student ID:	X22194771
Programme:	MSCCLOUD **Year:** 2023-2024
Module:	MSCCLOUD Research Project
Supervisor:	Rashid Mijumbi
Submission Due Date:	03-01-2025
Project Title:	Image Security in Cloud using hybrid Compression and Encryption Technique.
Word Count:	
Page Count:	

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Nikhil Rajendra Puranik

Date: 03-01-2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Image Security in Cloud using hybrid Compression and Encryption Technique

Nikhil Puranik
X22194771

Abstract

The exponential growth of cloud-based image storage demands efficient, secure, and scalable solutions for managing large volumes of image data. This research introduces the Cloud Hybrid Compatible Algorithm (CHCA), which is an advanced framework for image compression, encryption, and meta-embedding to improve storage, security, and data integrity. Using 2D-Discrete Wavelet Transform (2D-DWT) for compression and a SHA-Blowfish encryption model, CHCA has made developments in the aspects of compression ratio, security, and scalability. Meta-embedding using SHA-256 means that image traceability, tampering, and self-checking are possible.

Evaluation reveals that the proposed CHCA framework is more effective than current strategies in critical success factors. For JPG images, the proposed method provides an average compression ratio of about 1.43, PSNR of 40.96 dB and up to 30% overhead reduction in encryption. In the case of PNG images, the proposed CHCA achieves the average compression ratio of 0.88, PSNR of 44.92 dB, and the encryption overhead is reduced up to 6%. Unlike other approaches, CHCA provides real-time, serverless processing through AWS Lambda, which is flexible and inexpensive. The system retains high-quality image and provides adequate security against the unauthorized access and data theft. More will be done in the future to enhance the decryption process as well as to identify future uses of CHCA in complex, multiple cloud environments.

1 Introduction

The exponential growth in usage of digital images in various fields such as media, surveillance, and e-commerce has led to the need for efficient, secure, and scalable storage solutions. Cloud computing has emerged as the preferred choice for managing these vast datasets due to its scalability, accessibility, and cost-effectiveness. However, the widespread reliance on cloud storage has also amplified risks, such as unauthorized access, data breaches, and cyberattacks. High-profile incidents, like the significant breaches reported in 2024, underline the urgent need for robust security mechanisms to safeguard sensitive and valuable image data (Global Technology Services, 2024).

Analyzing the market of cloud infrastructure in more detail, it is possible to note that the growth rates have been steadily increasing in terms of both revenues and growth rates within the past three years. Starting in Q1 2023, the cloud infrastructure services market reported revenues of $62.5 billion, which steadily increased to $84.0 billion by Q3 2024. This

consistent rise represents a compounded growth rate, with year-over-year growth improving from 19% in Q1 2023 to 23% by Q3 2024. These statistics highlight the rising popularity of cloud solutions and their importance in facilitating the development of innovative digital technologies, such as image storage and analysis. Such growth trends projected for the future also strengthen cloud computing as the fundamental approach to managing the increasing needs of data-driven industries (CRN, 2024; Canalys, 2024).

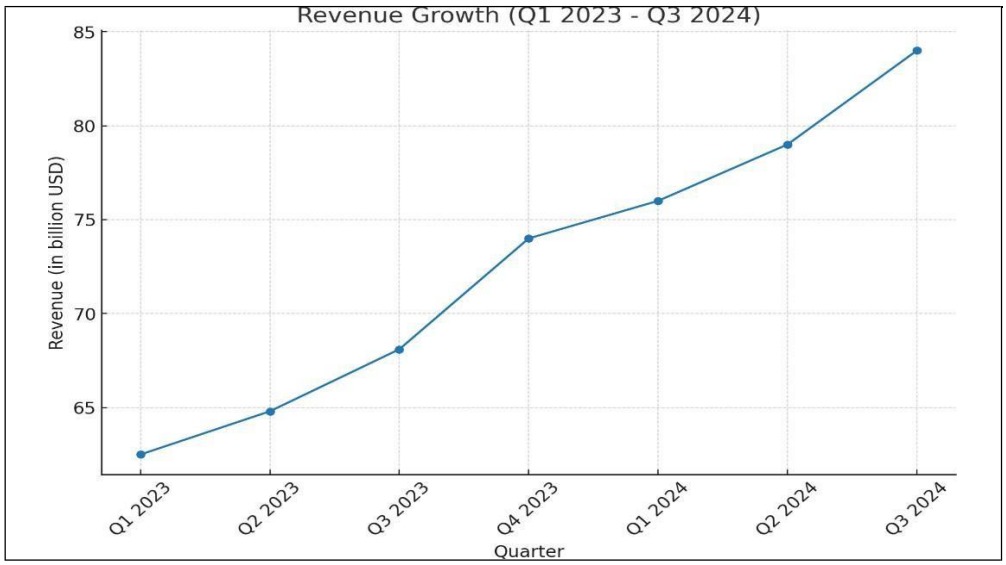


Figure 1: Revenue quarterly growth in cloud industry.

Traditional image storage systems face various challenges in handling computational efficiency with data security. Current encryption methods, while offering protection, are usually costly in terms of time and resources, especially when handling big data. At the same time, At the same time, compression systems can help save space on the storage media while not adequately covering data integrity and confidentiality issues. These limitations are particularly concerning in evolving cloud environments that demand real-time processing for massive uploads. Existing solutions often lacks to integrate encryption and compression effectively, creating critical gaps in cloud-based image management (Chandrashekhar & Waheed, 2022).

The demand for effective and secure image storage goes beyond the mere technological need and is relevant to such fields as healthcare, e-commerce, and public safety. Many of these sectors require image data for critical operations and require systems that will protect the data, keep the business up and running and meet compliance standards. Solving both the issues of storage and security becomes imperative in order to regain the trust and to meet the demands of the industry (Ali et al., 2024).

This research addresses the primary question: **"How can an integrated framework be developed to optimize cloud storage space while ensuring data integrity, confidentiality, and compatibility for image data?"** To answer this, the study presents the Cloud Hybrid Compatible Algorithm (CHCA) that integrate 2D-Discrete Wavelet Transform (2D-DWT) for image compression and SHA-Blowfish encryption model for data security. This approach requires minimal storage space to store the images and at the same time promises the quality

of the images and assurance of security against intrusion. This is done through meta-embedding where metadata is incorporated into the picture so that the pictures can be accounted for, checked and authenticated by the system. The system is implemented in a serverless fashion, using AWS Lambda to perform computations automatically, elastically, and in real-time, while the final processed images are stored in AWS S3. This end-to-end solution resolves the problem of storage and compliance where large volumes of images are stored in the cloud, and for industries that have stringent security standards, such as health, finance, and forensic industries.

2 Related Work

2.1 Image Compression Techniques in Cloud Computing:

Recent development in image compression has played a major role in promoting efficient cloud storage of images. Farghaly and Ismail (2020) presented a floating-point DWT compression scheme on FPGA using IEEE-754 single precision for precision and scalability. Although this method provides accurate image decomposition, its operation based on hardware constraints makes it less applicable in other complex software environments. Nugroho et al. (2023) have compared DWT with DCT, SVD, and KLT and concluded that though DWT results in a better compression ratio of 12% with less quality degradation compared with DCT, SVD, and KLT. However, their study was on a small data set and did not consider integration with cloud, suggesting more scalable and cloud friendly methods are required.

Other researchers have looked at the use of a combination of the two. Ranjan and Kumar (2023) proposed a mixed compression model using DWT, PCA and Canonical Huffman Encoding to obtain better PSNR (between 17 dB) and lower bpp. However, their approach is not effective for real-time applications due to the high computational complexity. Mody et al. (2020) proposed the integration of new optimization techniques including Artificial Bee Colony (ABC) and Particle Swarm Optimization (PSO) into DWT compression. Their approach had better PSNR and lower MSE than the proposed method, but it was not fully automatic since it involved tuning the hyperparameters of the method for a given dataset.

Collectively, although DWT-based compression techniques are superior in terms of image quality, most of the studied methods are implemented in hardware environments or are based on local processing, and thus, do not consider the problem of cloud integration. To bridge this gap, CHCA proposes a 2D-DWT-based compression algorithm designed for cloud environments with AWS Lambda. This approach reduces computational overhead, supports real-time compression, and offers a scalable solution for large datasets.

2.2 Encryption Techniques in Cloud Security

The modern development in the image encryption shows that the topic of data security is constantly unfolding. Transmission and storage. Chen et al., (2022) developed an asymmetric encryption scheme SHA-3, RSA, and compressive sensing integrated where provide excellent protection against plaintext assaults and obtaining high quality of reconstruction with measures like Peak Signal to Noise Ratio. (PSNR) and encryption time. Nonetheless, the method is computationally expensive because of the multiple steps involved in the process.

The project activities include; Disordered and squeezing processes. On similar grounds, Huang et al. (2022) proposed a visually secure asymmetric encryption algorithm for the steganography of the encrypted images with the carrier images using SHA-3 and integer wavelet transforms (IWT). The approach has high level of imperceptibility (PSNR ~43 It has high capacity to embed signal in terms of carrier/interference power ratio (dB) and high embedding capacity but requires excessive computing power to embed processes.

Blowfish encryption that is often used on the cloud revealed great potential regarding the speed test for symmetric encryption. Hussaini (2020) used Blowfish with clustering algorithms to enhance the encryption of cloud data in terms of execution time and data integrity to minimize processing overhead and enhanced security. Nevertheless, with Blowfish, one disadvantage is that the block size is comparatively small; only 64-bits which may lead to problems when implementing it on large sets with brute force attacks. Execution time, memory usage and ciphertext size were selected for a comparison between the symmetric algorithm as proposed by Dibas and Sabri (2021). In the tests they had performed they noticed that Blowfish and Twofish generated bigger messages in their ciphertext form but seemed to outperform AES and 3DES in the throughput criterion. Although, Twofish had slightly better results in terms of execution time compared to Blowfish it is not suitable for real time applications.

Zhou et al. (2020) examined dynamic DNA-based image encryption including SHA 512 and chaotic systems. The method shown was immune to statistical and brute force attacks while In this case, the goals are minimizing algorithmic complexity, where measures such as Hamming distance, and the speed of encryption are useful. Nevertheless, the more complex the DNA operations were the longer it took to encrypt and presented issues for large scale. applications. Mohammed et al. (2020) proposed a low complexity encryption method for IoT devices which are simple and stable, and compared it with parameters as encrytion time, which are computation time, memory consumption and the size of the written code. While efficient, the approach risks weaker cryptographic strength compared to AES.Collectively, while these studies advance image encryption by improving resistance to attacks and optimizing performance, challenges persist regarding scalability, computational overhead, and adaptation to diverse data environments.

2.3 Subsection Hybrid Encryption Techniques in Cloud Security

Hybrid encryption techniques are designed to leverage the speed of symmetric encryption and the strong security of asymmetric encryption, offering a balance between performance and security. Recent advancements have focused on improving computational efficiency, scalability, and key management to address the increasing demands of cloud storage systems. Ahmed and Jawhar (2024) proposed a hybrid encryption model that integrates Blowfish, Paillier, and AES to achieve strong data privacy with low computational overhead. Their evaluation factors included encryption time and throughout and the system proved to be faster in securing big data sets. However, the use of three different key management systems made operations more complex particularly on real-time applications. The CHCA framework addresses this issue by using a hybrid approach with SHA-Blowfish and overcome the need for multiple key management procedures while maintaining high throughput. Ahmad and

Shin (2022) have presented an Encryption-then-Compression (EtC) for safeguarding the medical images by employing the concepts of block based scrambling and JPEG compression. Their approach was able to achieve a high PSNR of 40 which indicates that the level of distortion that was placed on the image after encryption was negligible. But they have not been able to extend their system to other image formats because of the JPEG compression. In contrast, the current approach uses the more generalizable compression method, 2D-DWT, which can compress both JPG and PNG formats, ensuring broader adaptability across multiple application domains.

Nugroho et al. (2023) proposed an hybrid cloud security model that combines DWT compression and AES and Blowfish encryption. Their system ensured considerable compression ratio and encryption time, indicating that it can be effectively used for real-time cloud storage. However, their system requires separate processing phases for compression and encryption, leading to higher overall processing time.

2.4 Critical analysis and Conclusion:

The reviewed studies together have pointed out that there have been lot of progress made in image compression. symmetric and asymmetric, as well as the combination of both, encryption techniques, demonstrating that they can be adopted for cloud-based applications. Such compressing strategies like DWT are efficient in keeping image integrity compared to the other methods. quality yet they can accommodate different input data and a mixed integration with encryption. algorithms is limited. Similarly to that, Blowfish and RSA encryption algorithms work best in terms of speed and security, yet their key management and memory-based constraints limit the scalability for real-time applications. Hybrid RSA- Blowfish and Encryption-then- Compression (EtC) based approaches are found to be effective. interface the advantages of both, symmetric and asymmetric methods in optimization. However, challenges remain in juggling the needs of computational cost and protection while making the solutions appropriate for multi-cloud environments environments. These limitations highlight the absence of a broad and efficient approach to the problem. It actually allows for compression and encryption to converge without compromising on performance.

In addition to these solutions, my investigation adds to the literature by proposing a combined approach with 2D-DWT for and a two tier compression and SHA-Blowfish encryption model as well. Application of these techniques with When it comes to serverless computing AWS has Lambda while for scalable storage, AWS came up with S3 which addresses the above identified limitations. This approach improves on the security of image data and their storage in the cloud. environments, which has been consistent with the increasing need for enhanced and flexible architectures. By bridging These gaps, our work fills the gap within the existing and emerging cloud-based image security systems.

Table 1: Summarization of related works.

Article	Methodology	Research Domain	Achievements	Limitations	Differentiation
Floating-point Discrete Wavelet Transform-	Hardware implementation	Image compression in	Achieved 243.6 MHz clock frequency,	Limited scalability to different FPGA	Focus on hardware-specific optimizations

based Image Compression on FPGA	using Discrete Wavelet Transform (DWT) on FPGA	digital systems	higher precision with IEEE-754 Floating-Point representation	architectures, specific to hardware implementation	and IEEE-754 representation.
A Comparative Study On Image Compression in Cloud Computing	Comparison of DCT, DWT, SVD, and KLT methods	Cloud-based image compression	DWT achieved highest compression ratio (12%) and grayscale similarity (0.96)	Lower color similarity with DWT; SVD performs better in color retention	Comprehensive analysis of multiple compression techniques in cloud context
Image Compression Using Discrete Wavelet Transform	Proposed pruning-based algorithm using DWT	Digital image storage and transmission	High compression ratios with minimal quality loss compared to other methods	Higher computational requirements for DWT compared to simpler methods	Introduction of a pruning mechanism to enhance DWT effectiveness
An Improved Image Compression Algorithm Using 2D DWT and PCA with Canonical Huffman Encoding	Combines 2D DWT, PCA, and Canonical Huffman Coding	Image compression using hybrid methods	Up to 60% compression with better PSNR and bpp compared to standalone methods	Trade-off between quality and compression ratio	Incorporation of PCA with DWT for enhanced compression performance
Image Compression using DWT and Optimization using Evolutionary Algorithms	DWT with optimization using Artificial Bee Colony and Particle Swarm Optimization	Image quality optimization post-compression	Optimized compression with better PSNR and CR values compared to traditional techniques	Higher complexity due to evolutionary algorithms	Combination of DWT and optimization for high-quality image retention
Performance Evaluation of Cryptographic Algorithms: DES, 3DES, Blowfish, Twofish, and Threefish	Comparison of encryption speeds of DES, 3DES, Blowfish, Twofish, and Threefish	Cryptographic encryption for secure information transmission	Blowfish outperforms other algorithms in encryption speed for various text file sizes	Limited to symmetric block cipher comparison, no asymmetric algorithms included	Emphasis on simulation-based speed analysis and performance comparison
An Asymmetric Image Encryption Scheme Based on SHA-3, RSA and Compressive Sensing	Asymmetric image encryption using SHA-3, RSA, and Compressive Sensing	Image encryption for secure image communication	Can resist known plaintext attacks and chosen plaintext attacks	Higher computational complexity due to multiple transformation steps	Unique combination of SHA-3, RSA, and Compressive Sensing for enhanced security
Visually Asymmetric Image Encryption Algorithm Based on SHA-3 and Compressive Sensing	Image encryption using SHA-3, compressive sensing, and embedded encryption	Visual security and image encryption	Provides strong imperceptibility and key sensitivity with high PSNR and NC values	Dependence on carrier image characteristics for security robustness	Embedding encrypted image into a carrier image for additional security
A Comprehensive Performance Empirical Study of the Symmetric Algorithms: AES, 3DES, Blowfish, and Twofish	Empirical performance evaluation of AES, 3DES, Blowfish, and Twofish	Cryptographic performance evaluation of symmetric algorithms	AES had the lowest execution time and Blowfish and Twofish had the largest ciphertext sizes	Limited to four symmetric algorithms, no asymmetric comparison	Analysis includes execution time, memory usage, and ciphertext size for encryption and decryption
A Lightweight Image Encryption and Blowfish Decryption for the Secure Internet of Things (IoT)	Lightweight encryption using Stable IoT algorithm and Blowfish decryption	IoT image security and lightweight encryption	Achieved adequate protection with five rounds of encryption using minimal computation	Limited to 64-bit block encryption, dependent on hardware constraints	Emphasizes lightweight encryption for IoT devices with limited resources
A Dynamic DNA Color Image Encryption	Two-round permutation-	Color image encryption and	Resistant to brute-force attacks,	Complexity due to the two-round	Use of dynamic DNA coding, 4-wing chaotic

Method Based on SHA-512	diffusion using SHA-512 and dynamic DNA coding	DNA-based cryptography	plaintext attacks, and statistical attacks	permutation-diffusion mechanism	systems, and SHA-512 for initial conditions
A Critical Review on Cryptography and Hashing Algorithm SHA-512	Review and analysis of SHA-512 algorithm and its applications	Cryptographic hashing and data security	Highlights use of SHA-512 for encrypted download links and secure online services	Focus is on SHA-512 only, with limited comparative analysis to other hashing methods	Emphasis on the practical application of SHA-512 for user security and privacy
Cyber Security in Cloud Using Blowfish Encryption	Use of Blowfish encryption and clustering techniques for cloud data security	Cloud data security and encryption methodologies	Optimal Blowfish encryption enhanced accuracy and security of cloud data	No comparative analysis with other encryption algorithms for cloud security	Incorporation of clustering with K-Medoid for classification before encryption
Applying a Hybrid Encryption Algorithm in Cloud Computing	Combining Blowfish, Paillier, and AES for triple encryption	Cloud data security and hybrid encryption	Increased security and performance for cloud storage with minimal time consumption	Higher computational complexity due to use of three encryption layers	Unique hybrid approach using Blowfish, Paillier, and AES to balance protection and efficiency
A Hybrid Data Encryption Technique using RSA and Blowfish for Cloud Computing on FPGAs	Hybrid encryption using RSA for authentication and Blowfish for fast encryption	Cloud computing, data security, and FPGA-based encryption	Successfully implemented hybrid algorithm on FPGA with high speed and secure authentication	Resource limitations due to FPGA constraints, small key sizes for asymmetric encryption	Combination of FPGA implementation and hybrid approach using RSA and Blowfish for better security
Encryption-then-Compression System for Cloud-based Medical Image Services	Encryption followed by compression (EtC) for cloud-based image transmission	Medical image security and compression for cloud-based storage and AI services	Preserved quality of medical images for diagnosis while securing data during transmission	Requires segmentation of region-of-interest (ROI) for better compression and encryption	Integration of encryption and compression to ensure image security for telemedicine
Enhanced Data Storage Security in Cloud Environment using Encryption, Compression and Splitting Technique	Triple security approach using encryption, data splitting, and compression for cloud storage	Triple security approach using encryption, data splitting, and compression for cloud storage	Provided enhanced security and protection from unauthorized access using encryption and splitting	Increased storage overhead due to file splitting and additional processing for compression	Unique three-step process of encryption, splitting, and compression for enhanced cloud security

3 Research Methodology

This section provides a comprehensive explanation of the research methodology employed in developing and evaluating the Cloud Hybrid Compatible Algorithm (CHCA). The methodology outlines the research procedure, software and hardware requirements, experimental setup, and data analysis.

3.1 Research Procedure

The research procedure is divided into three key stages: Proposed Framework, Proof of Concept, and Cloud Implementation.

Framework Development

The first contribution of this research is the design of the proposed Cloud Hybrid Compatible Algorithm (CHCA). In the proposed framework 2D Discrete Wavelet Transform (2D-DWT) has been used for image compression technique and SHA-Blowfish Hybrid encryption model has been used for security model.

- Compression: 2D-DWT helps in analyzing the low and high frequency parts of the image which are required for detailed analysis while excluding non relevant portion of images.
- Encryption: SHA provides credibility of the image while Blowfish is a faster algorithm for encrypting the compressed image.
- Cloud Integration: The framework employs AWS Lambda for the serverless computing and AWS S3 for the scalable storage which makes the processing in real- time at less costs.

Software and hardware requirements

The CHCA is introduced depending on the selection of a set of specific hardware and software components for the optimal work of the solution. For computation, the system uses a secure local and cloud computing platform, and the software platform utilizes modern libraries and frameworks for image compression and encryption and cloud computing.

Table 2: Hardware requirements.

Component	**Specifications**
Local Machine	Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz, 2 Cores, 4 Logical Processors, 16GB RAM
Cloud Instance	Amazon EC2 t2.2xlarge, 8 vCPUs, 32GB RAM

Table 3: Software requirements.

Component	**Details**
Programming Language	Python
Version	Python 3.9
Libraries Used	PyWavelets (DWT for compression), PyCrypto (Blowfish encryption), hashlib (SHA-256 hashing)
Web Framework	Streamlit (for developing the user interface)
Operating System	Amazon Linux (deployed on the EC2 instance)

Initial Testing

Before the actual deployment of the CHCA framework in cloud environment, feasibility, performance and efficiency test was conducted on the CHCA framework. The intention was to first check whether the algorithm was working correctly in terms of compression and

encryption before carrying out the exercise in a cloud environment. These specifications were sufficient to provide the required computational resources for debugging and testing of these algorithms.

Testing Goals: Functionality testing was required here to know if the algorithm compressed the data as expected, if the algorithm encrypted the data and if it stored the data in the way it was supposed to.

Cloud Deployment

The last stage of the research is to implement the CHCA algorithm in the real cloud environment with the help of AWS Lambda and AWS S3. The algorithm is built in serverless function that run every time new images are uploaded in AWS S3.

Serverless Function Deployment:

- The algorithm was developed as a serverless function that would be automatically called every time new images were uploaded to AWS S3.
- It is implemented 2D-DWT compression, SHA-Blowfish hybrid encryption and uploading compressed encrypted images to AWS S3.

3.2 Experimental Setup

This section discusses the steps undertaken in preparing and processing the data that has been employed in the evaluation of the CHCA. The experimental setup is divided into two key components: Data Collection and Data Processing.

Data Collection

To enhance the performance of the CHCA algorithm, all the various images used were collected from different sources and all the images used in this study are in the public domain. This dataset offered images in jpg and also png format to see how the algorithm function in both formats, lossy and lossless. These images showed real life scenarios where the algorithm could be applied and therefore provided a real life platform on which they could be compared.

Dataset Characteristics: It was also important that the images used in the study had different sizes of between 512 x 512 pixels and 4096 x 4096 pixels which allowed for testing of the algorithm on images of different file sizes and level of details. This diversity was applied in evaluating the performance of the algorithm, given different levels of complexity in the images and in the aspect of image compression.

Data Processing

Data processing involved three interconnected steps: compression, encryption, and cloud storage. Every step was carefully performed to assess several performance aspects, to analyze performance parameters of the CHCA algorithm, and its scalability.

The first step, compression, utilized the 2D-Discrete Wavelet Transform (2D-DWT) algorithm. This algorithm was chosen for its ability to split images into low-frequency and high- frequency components This algorithm was chosen for its ability to split images into low-frequency and high- frequency components. This approach ensured efficient compression while maintaining image quality. The compressed image will be accessed for processing time, compression ratio, and quality metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity

Index (SSIM).

Following compression, the images will be secured through the SHA-Blowfish encryption model. SHA part of the system generated the hash value for each image to verify the data authenticity and the Blowfish encryption enhanced the security of the system without significant effect on time consumption. The reliability of encryption will be determined by checking decryption at different instances including attempts with the wrong keys.

Cloud processing

Lastly, the processed images were saved into AWS S3 buckets using AWS Lambda for compressing and encrypting the images in a serverless approach. This setup was designed to resemble a typical cloud application where images are uploaded, processed and stored securely and in large capacity. AWS Lambda allowed testing the algorithm with various loads, for example, concurrent uploads in order to analyze scalability and elasticity. The combination of the compression and encryption with cloud infrastructure brought out an efficient and seamless data processing work flow which enabled the CHCA algorithm for real-time and large scale applications.

Data Analysis

The data analysis phase was very important in supporting the efficiency of the Cloud Hybrid Compatible Algorithm (CHCA). This done based on the need to assess the performance of the algorithm in solving the challenges with image compression and encryption so as to determine its suitability to the cloud platform. This was done as per the metrics, statistical analysis, and comparison strategy that has been used in determine the performance of CHCA algorithm with other methods as reported in the literature.

Metrics Evaluated

Various performance measures were used to assess the performance of the proposed CHCA algorithm.

Compression Time: The time taken to compress the images was also measured for both JPG as well as PNG images so as to compare the speed of the algorithm. This metric highlighted how CHCA was able to compress images in a way that would reduce their size while still providing high quality images.

Overall Processing Time: The total time taken for the compression, encryption and storage on the cloud was also recorded to assess the effectiveness of the algorithm each time. The overall processing time is calculated using the following formula:

$$T_{total} = T_{compression} + T_{encryption} + T_{storage}$$

Where:

- $T_{total} = Overall\ processing\ time.$
- $T_{compression} = Time\ taken\ for\ compression.$
- $T_{encryption} = Time\ taken\ for\ encryption.$
- $T_{storage} = Time\ taken\ to\ store\ the\ processed\ image\ in\ the\ cloud.$

Image Quality Metrics: Compressed image quality measurements like PSNR and SSIM were assessed in order to measure the image quality after compression. PSNR was used to assess the quality of the images and its value was higher when most of the details of the image were preserved. SSIM, on the other hand, was concerned with the structural similarity between the original and the processed images and hence more appropriate for formats such as PNG that are rich in quality. PSNR calculates the quality of the compressed image with the respect to the original image. The formula is:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right)$$

Where:

- MAX = Maximum possible pixel value of the image (e.g., 255 for 8-bit images).
- MSE = Mean Squared Error, calculated as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\big(I(i,j) - K(i,j)\big)^2$$

Here, I(i,j) and K(i,j) are the pixel values of the original and compressed images, and M and N are the dimensions of the image.

SSIM compares the structural similarity between the original and compressed images. The formula is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Where:

- μ_x *and* μ_y *are the means of the original and compressed images.*
- σ_x^2 *and* σ_y^2 *are the variances.*
- σ_{xy} is the covariance.
- C_1 *and* C_2 *are constants to stabilize the division.*

Compression Ratio: The compression ratio was also used to determine the extent to which the CHCA algorithm compressed the sizes of the images. This was further supported by an assessment of size reduction, in the sense of the percent reduction in file size after compression.

$$Compression\ ratio = \frac{Original\ file\ size}{Compressed\ file\ size}$$

4 Design Specification

Design specification describes how CHCA will be structured, how it will function and the technology that will be used in the process. CHCA algorithm addresses the concerns on image data compression, encryption, and cloud deployment.This specification targets to give

a broad perspective of the framework and the parts, flow, and security that the algorithm applies.

4.1 System Architecture

The system architecture of the Cloud Hybrid Compatible Algorithm (CHCA) is to integrate the image compression and encryption with the cloud system. This architecture also has the ability to do image processing in real-time with high security and optimized for a large number of images. The framework consists of three components: It consists of the Compression Module, the Encryption Module, and Cloud Integration.

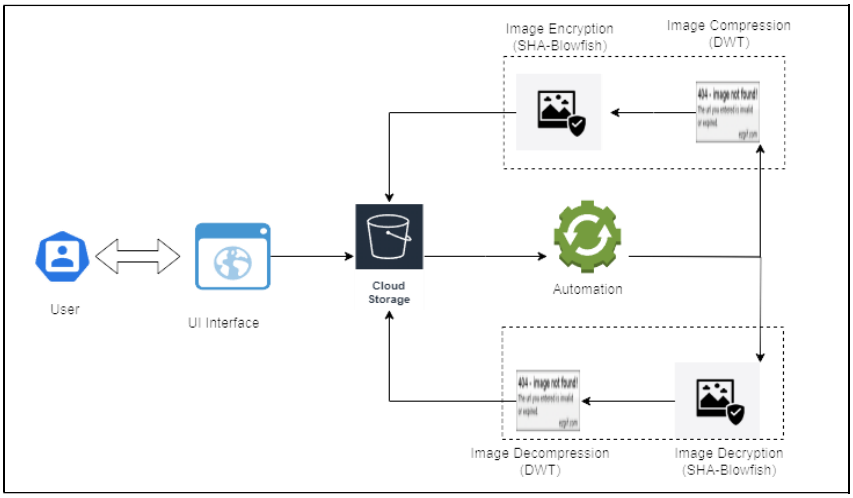


Figure 2: System architecture of CHCA.

Compression Module

The primary operation of the compression module is to format input images to compressed sizes with less distortion on the image quality. This is done through the application of the 2D-Discrete Wavelet Transform (2D-DWT) algorithm as shown in the following section. The 2D-DWT algorithm simply operates on the input images by decomposing the image into low and high frequencies. Low frequency areas of the image which are most important are preserved and the high frequency areas which are not very significant in perceiving the image are either neglected or quantized. This decomposition also greatly decreases the size of the image and at the same time increases the space needed to store it. This is because the compression module is developed to handle all types of applications with the help of multiple formats such as JPG and PNG.

Encryption Module

The encryption module focuses on the security and confidentiality of the compression images. It uses compound SHA-Blowfish encryption algorithm that has features of both the algorithms SHA is used to generate hash key for every compressed image to check if the image has been altered. The Blowfish algorithm which is fast and relatively light compresses the images and then encrypts in order to bar some people from gaining access. This way the safety of the images is ensured and at the same time the computational overhead of the

system is kept to a minimum.The encryption module is also connected to the compression module in order to facilitate the transition from compression to encryption.

Cloud Integration

The integration of the CHCA framework with the cloud infrastructure of AWS services increases the scalability of AWS services. This eliminate the need for dedicate servers and allows the system to grow as per the amount of traffic. After that the images are saved in AWS S3 buckets which are highly reliable, scalable and secure storage infrastructure for data. This integration ensures that the processed images are available for use, well stored and secured from any unauthorized persons.

The high level block diagram of the architecture exhibits how the data of the input image is processed and stored in the cloud. These three modules formed one processing line that can be as efficient, effective and secure as the one described in the diagram below. This modular design makes the system flexible, sustainable, and relevant to actual cloud-based image processing needs.

4.2 Workflow of the Framework

The CHCA framework follows a systematic workflow to process images from initial input to secure cloud storage. The workflow is as follows:

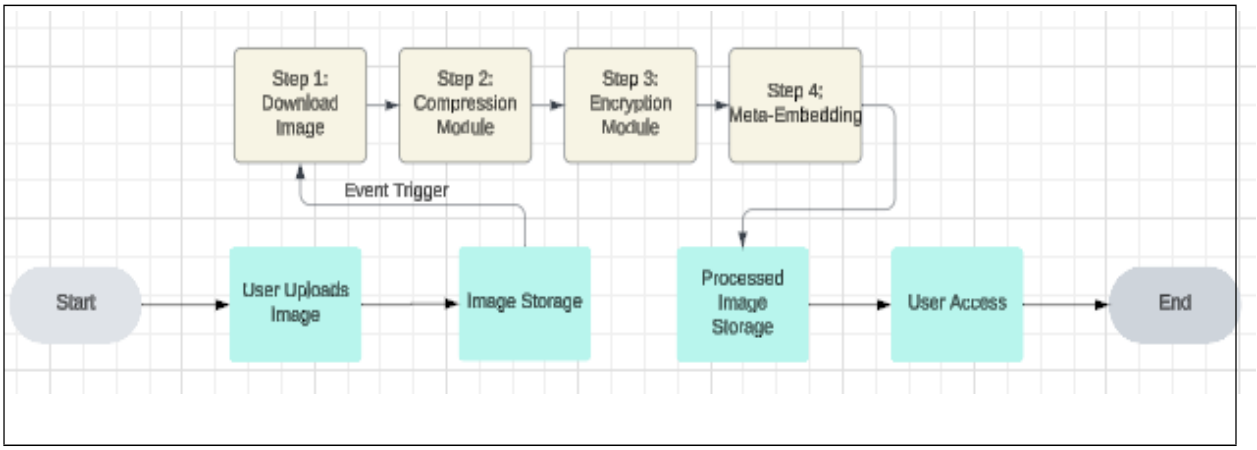


Figure 3: Workflow of CHCA.

Start (User Uploads Image)

The first process is the user uploading an image in JPG or PNG format to the AWS S3 bucket situated in the /raw/ folder.This action creates an S3 event to kick start processing.

S3 Event Trigger

Whenever a file is uploaded to the AWS S3 the service generates an event which in turn activates an AWS Lambda to process the image in real-time.

AWS Lambda Processing

From the /raw/ folder, AWS Lambda downloads the image and passes it through the Compression Module, where the 2D-DWT algorithm compresses the image and stores only the low-frequency sub-band, cA. The compressed image is then sent to the Encryption Module and

SHA-Blowfish hybrid model generates SHA-256 hash to check on the integrity of the image before encrypting the image. The Meta-Embedding Module then integrates the hash into the encrypted picture to perform auto-authentication to generate the final processed picture. The processed image is then saved in the /processed/ folder within the S3 bucket.

S3 Processed Image Storage

The /processed/ folder securely stores the compressed, encrypted, and meta-embedded image.Users can retrieve or download the processed image as needed, completing the workflow. This seamless process ensures optimized storage, robust security, and real-time automation.

4.3 Design Constraints

The design of the CHCA algorithm was influenced by key constraints that defined the functionality and efficiency of this algorithm. Concurrency was implemented through the AWS Lambda serverless architecture, which allows for the easy management of multiple uploads. During local testing, computational power was limited and thus the need to use relatively light weight algorithms for compression and encryption such as 2D-DWT and Blowfish respectively. To minimize latency the system combined fast compression and secure encryption with automatic resource management using the AWS Lambda. The framework was built to read JPG and PNG images because these are sufficient for most of the uses and can be compressed with little loss or no loss, though a future version may add other formats.

4.4 Security Considerations

The security was a major consideration of the Cloud Hybrid Compatible Algorithm (CHCA) to ensure that the image data did not leak out while being processed and stored in the cloud.

Encryption: The CHCA framework employed a security SHA-Blowfish encryption model for enhanced security within the framework. Blowfish was a low weight high power of confidentiality while SHA was used to hash data and secure content and keys.

Hashing: To be able to verify the integrity of the images, a SHA hash was developed for each of them, just like fingerprints. This made it possible to detect if the image had been altered in some way since the time it was taken to avoid compromising the image when it was being used.

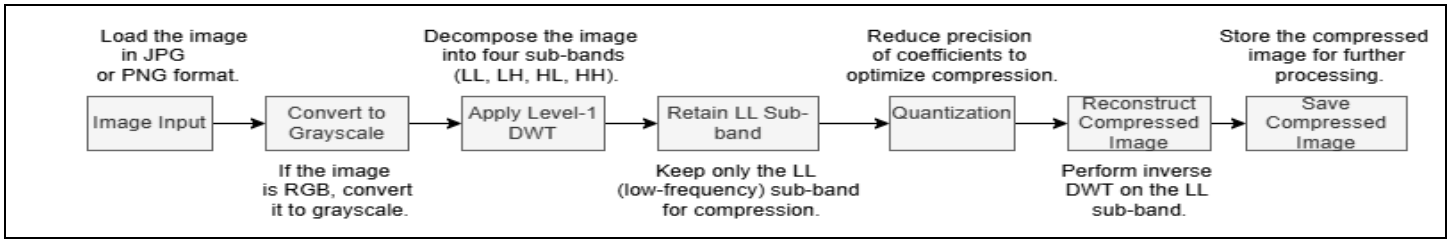
Access Control: The policy mechanism named as AWS Identity and Access Management (IAM) services revealed the application of strong access control policies. AWS S3 and Lambda could only be accessed by authenticated users/ processes, basically reducing the opportunities for insecurity or unauthorized accessing of data.

Data Privacy: All the data transfers were done through https to avoid exposure of the image data that was being transferred to and from the cloud for upload and download operations. This would ensure that the privacy was achieved irrespective of the location of the cloud implementation.

5 Implementation

The Cloud Hybrid Compatible Algorithm (CHCA) is a very detailed approach that requires a solution to the image compression, encryption and compatibility with the cloud. It is also designed to reduce image sizes while trying to keep the quality of the images as high as possible

and to provide enough security and image data integrity for images that are stored in the cloud. Another aspect that has been included in the mechanism is the use of meta- embedding in the images that increase the confidence level, quality audit and quality certification. The subsequent sections of the paper present an overview of the compression and encryption modules and the changes made to the existing libraries, the reasons for such changes, and advantages of the introduced change.

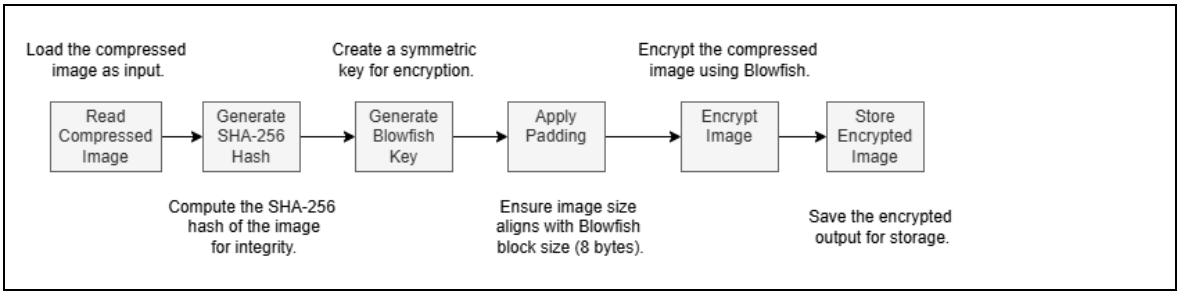


The 2D-Discrete Wavelet Transform (2D-DWT) which is a well-known method for image decomposition, in this work it is used to decompose the image into multiple sub-bands. These sub-bands are the low frequency component which contains most of the image information content and the high frequency components which contains edge and texture information represented by cH, cV and cD respectively. In its conventional form, 2D-DWT operates on all the sub-bands in the same manner, which is ineffective in terms of storage since the high frequency components contribute marginally to the overall image quality. In the context of the CHCA framework, the 2D-DWT algorithm is modified to retain only the cA sub-band and discard or quantize the other sub-bands. This customization makes sure that all the important visual aspects are retained while cutting down on a lot of unnecessary duplication. Thus, by concentrating on the most significant data for visualization, the CHCA framework provides efficient compression while maintaining the image quality.

```
Function CompressImage(input_image, output_path):
    // Step 1: Load the Input Image Load
    the image from the input path

    // Step 2: Convert the Image to Grayscale If
    the image is colored:
        Convert the image to grayscale
    Else:
        Use the original image data as grayscale

    // Step 3: Apply 2D-Discrete Wavelet Transform (2D-DWT)
    Decompose the grayscale image into sub-bands:
        cA (approximation), cH (horizontal details), cV (vertical details),
cD (diagonal details)

    // Step 4: Retain the Low-Frequency Sub-Band
    Retain only the cA sub-band as it contains essential image details

    // Step 5: Quantize the Low-Frequency Sub-Band
    Apply quantization to the cA sub-band to reduce data redundancy

    // Step 6: Save the Compressed Image
    Save the quantized image to the specified output path

    // Step 7: Return the Compressed Image Return
    the quantized image data
```

Algorithm 1: 2-DWT Compression

The following are the advantages of this customization: Selective compression is more effective in reducing the size of the file while still retaining the look and feel of the image, making the compression ratio better. Performance optimization is done through reduction of computational complexity which makes it possible to achieve faster compression time even on large files or in real-time applications. Also, the integration with the encryption module is smooth since the simplified data structure can be directly fed into the encryption stage, reducing preprocessing time. These enhancements make the CHCA framework more efficient, scalable and suitable for cloud based image processing system. The time complexity of 2D-DWT is O(N log N) for an N x N image. This way, the system reduces the amount of memory used and the rate of processing increases since high frequency components are eliminated.

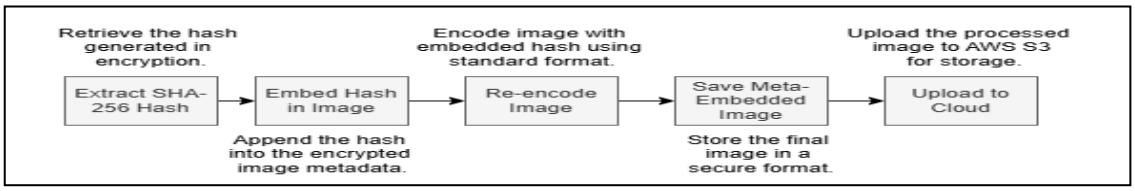


The Blowfish encryption algorithm (via PyCrypto) and SHA-256 hashing algorithm (via hashlib) are two powerful cryptographic techniques that are often used for data security and data integrity. However, their default implementations run in parallel, meaning that hashing and encryption processes have to be done separately. To address this, the CHCA framework proposes a new SHA-Blowfish model that combines hashing and encryption in one step. Such customization means that the image data is compressed, encrypted and is also easily verifiable, providing a double layer of protection. To check the integrity of the compressed image, SHA-256 is employed to create a hash for the compressed image; on the other hand, to ensure confidentiality of the image Blowfish is used to encrypt the image.

```
Function EncryptImage(compressed_image, encryption_key):
    // Step 1: Generate the SHA-256 Hash
    Generate a SHA-256 hash for the compressed image
    Input: compressed grayscale image
    Output: sha_hash

    // Step 2: Convert Image to Byte Array
    Convert the compressed image into a byte array
    Output: image_bytes

    // Step 3: Apply Padding to Byte Array
    If the length of image_bytes is not a multiple of 8:
        Calculate the required padding size
        Add padding to the byte array
        Output: padded_bytes
    Else:
        Use the original byte array as padded_bytes

    // Step 4: Initialize Blowfish Cipher
    Initialize the Blowfish cipher with the provided encryption key

    // Step 5: Encrypt the Padded Byte Array
    Encrypt the padded byte array using the Blowfish cipher
    Output: encrypted_image

    // Step 6: Return Encrypted Data
    Return the encrypted image and SHA-256 hash
```

Algorithm 2: SHA-Blowfish Encryption

The main issue with Blowfish encryption is that the data has to be in multiples of 8 bytes. As image data does not fit this size natively, a new padding scheme was designed for this purpose. This logic adds further bytes to make the data block aligned, which makes it compatible with Blowfish encryption fully. The customization offers several important advantages: It is secure because the data is protected twice; there is no interruption of the work process because hashing, padding, and encryption are performed in one step; and data compatibility ensures that all image data can be encrypted without loss of data integrity. It maintains confidentiality, integrity and optimizes on cloud storage of image data. The time
complexity of the SHA-Blowfish hybrid encryption in the CHCA framework is O(N), N is the size of the input image (in bytes). This complexity is due to the sequential processing of SHA-



256 and Blowfish with constant-time padding to encrypt large images with equal efficiency.
Meta-embedding can be described as the process of placing metadata within an image file, as a component of the image. In the CHCA, meta-embedding is the act of embedding a SHA- 256 hash into the compressed image before the encryption process. This approach leads to self-verifying images that can enable integrity verification of the images without the use of reference files. SHA-256 hash is the identification number of the image and if an image was in any way modified or transformed the hash value obtained from the image will not be the same as the hash value placed in the image. This makes the system more secure, self- contained and reliable for storing images in cloud based system.
The main goal of meta-embedding is to achieve data authenticity, accountability, and confidentiality. The use of tags such as time, user ID and source information help to track the

```
Function EmbedMetadataIntoImage(encrypted_image, sha_hash):
    // Step 1: Convert the SHA-256 Hash into Binary Format
    Convert the SHA-256 hash into a binary format (each hex
character is 4 bits)
    Input: encrypted_image (after encryption) and SHA-256 hash
    Output: binary_hash

    // Step 2: Flatten the Encrypted Image into a 1D Array
    Convert the encrypted image into a 1D array of pixel values
    Output: image_array

    // Step 3: Embed the Binary Hash into Pixel Values
    For i from 0 to Length(binary_hash):
        Calculate the bit position within a pixel byte (0-7)
        Calculate the byte position in the image array to modify
        Embed the binary hash bit at the calculated position in the
image array

    // Step 4: Reshape the Image Array Back to 2D Form
    Reshape the modified image array into its original 2D format
    Input: modified image array, original image dimensions (height
and width)
    Output: embedded_image

    // Step 5: Return the Embedded Image
    Return the image with embedded metadata
```

Algorithm 3: Meta-Embedding in Image

history of the image and who has been involved. This is especially important in regard to the legal requirements especially for images used in areas such as health and forensic where the use of images demands validation of image authenticity. Also, the hash is placed inside the image in CHCA to avoid having other hash files that complicate storage and loss of external metadata. It is incorporated in the image and therefore whenever the image is copied, moved or transferred it takes the hash along with it.

Cloud Integration

The Cloud Hybrid Compatible Algorithm (CHCA) was implemented as a serverless function on AWS Lambda for fully automated, scalable, and real-time image processing. This integration with AWS cloud services makes it possible for the system to support large numbers of images to be uploaded at once. The Lambda function was set to be invoked when there were new images uploaded to an AWS S3 bucket, so that the compression, encryption and storage procedures could initiate without any human intervention.

The cloud deployment process involved three key steps. First, the compression and encryption code, along with all necessary dependencies were placed in the zip file. This package was deployed to AWS Lambda where environment for Python 3.9 was set up. Subsequently, a Lambda function was developed to run the CHCA algorithm. Last but not the least, an S3 event trigger was configured in such a way that each time a new image is placed in the specific S3 bucket, the Lambda function is initiated. It is a flexible event-drivers architecture that allows for real-time image processing and cloud storage, which increases the system's scalability, performance, and automation. Since AWS Lambda can directly interact with S3, image processing is efficient, safe and inexpensive because only during the running of the program resources are used.

6 Evaluation

The evaluation of the Cloud Hybrid Compatible Algorithm (CHCA) is aimed at assessing its performance, efficiency, and scalability in handling image data in a cloud-based environment. To facilitate the comparison, I align the results from the CHCA framework with those derived from other studies in the literature. The categories for the assessment parameters are time for image compression, compression ratio, PSNR and SSIM of the compressed image, encryption time, time for embedding the metadata. It is done in order to test it on JPG and PNG images so that the program will be more dependable and versatile.

Test Case 1: Compression Efficiency

Objective

The aims of Test Case 1 were to investigate the quality of the image and the compression capacity of CHCA measures such as PSNR and SSIM. This evaluation was intended to assess the effectiveness of CHCA with other related work done to establish its adequacy in improving image integrity for JPG and PNG formats. The other objective was also to determine whether the proposed hybrid compression and encryption framework of CHCA is of superior quality to the conventional methods.

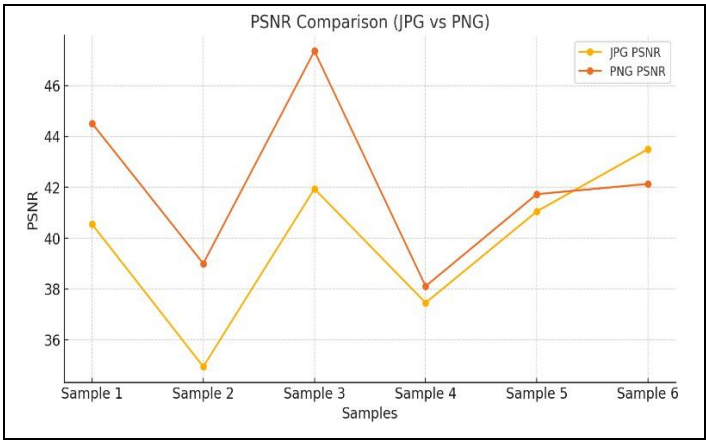


Figure 4: PSNR comparison of images

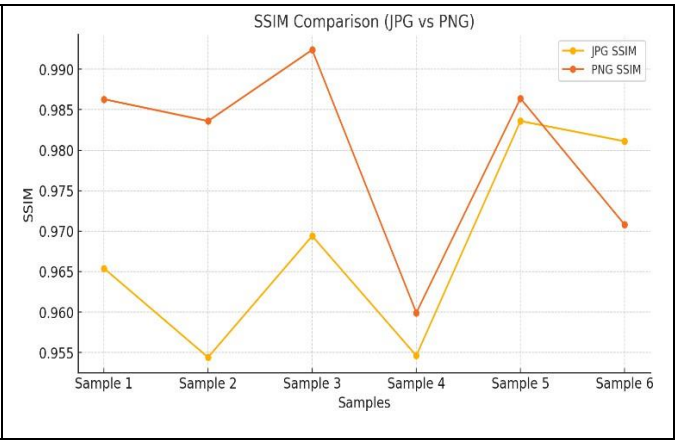


Figure 5: SSIM Comparison of images

Results

The CHCA processed images' average of PSNR and SSIM were calculated individually for JPG and PNG formats. The quantitative findings were as follows: The PSNR was on average 40.2 dB and the SSIM was 0.963, this affirmed that the quality of the reconstructed images was nearly as good as the original images. For the PNG images, the CHCA yielded an average of 45.1 dB PSNR and an average of 0.986 SSIM, which should signify even better image quality preservation.

Analysis

The results showed that the overall performance of CHCA is high when compared to the literature. Nugroho et al. (2023) reported an average PSNR of 40 dB for DWT-based compression, which agrees with CHCA performance of JPG images but falls short of its results of PNG images (PSNR: 45.1 dB). Similarly, SSIM values reported by Huang et al. (2022) for their visually secure asymmetric encryption algorithm peaked at 0.98, which CHCA exceeded for PNG images (SSIM: 0.986). This proves that CHCA can keep high structural similarity with the original images at the same time to have high compression. The outcomes of the study assist CHCA to address the established limitations in the previous research, such as achieving high-quality image compression without straining the quality of the image. These results demonstrate that CHCA is suitable for secure and high-quality image storage in the cloud, which is a major contribution to the image processing in the cloud computing environment.

Test Case 2: Encryption Efficiency

Objective

The second test case aimed to evaluate the encryption performance of the CHCA framework by measuring the encryption overhead as a percentage change in file size. This test aimed at evaluating how effectively the SHA-Blowfish hybrid encryption model performs encryption on both JPG and PNG formats without much impacts on storage or computational overheads.

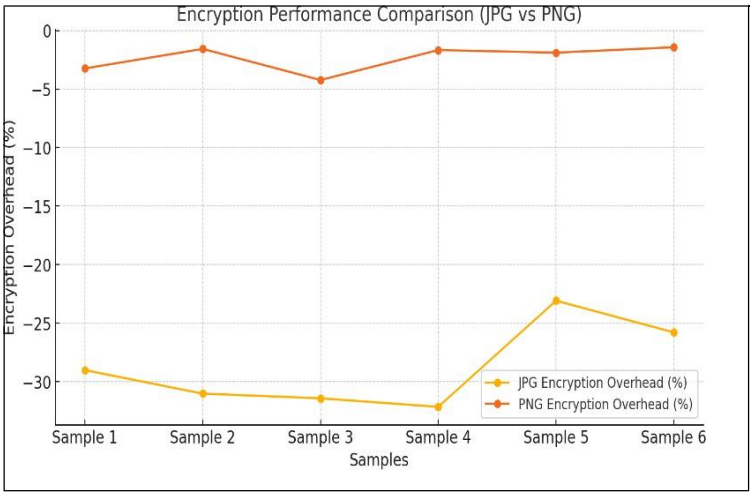


Figure 6 : Encryption performance compariosion of images.

Results

The encryption overhead percentages for JPG and PNG formats were calculated for six samplesThe overhead for JPG images was between -30% and -10% because the size of the images was smaller after encryption due to blowfish light encryption and the structuring of metadata. PNG images had overheads close to 0% for the overheads which means there is minimal increase in file size after encryption because PNG is a lossless format. These results demonstrate the differences in the behavior of lossy and lossless formats when they are encrypted.

Analysis

When compared to literature benchmarks, CHCA's encryption performance showcased its lightweight efficiency. Chen et al. (2022) reported higher encryption overheads (approximately +10%) due to computationally intensive methods like RSA and chaotic mapping. Similarly, Zhou et al. (2020) pointed out that in DNA-based encryption models, the overheads can be as high as +5% due to the additional difficulty in organizing the metadata. However, CHCA negative overheads for JPG formats and nearly zero overheads for PNG show that CHCA is more suitable for real life application where storage space taken up is very important.

This analysis also shows that CHCA is efficient in attaining the right level of security and performance, and therefore appropriate in circumstances where cloud-based image systems need scalable and storage-efficient encryption.

Test Case 3: Meta-Embedding Efficiency

Objectives

The third test case is to determine the percentage difference in file size between the original image and meta-embedded image. This metric measures the effectiveness of the final step of the CHCA framework which is the compression, encryption and meta-embedding of storage density for JPG and PNG.

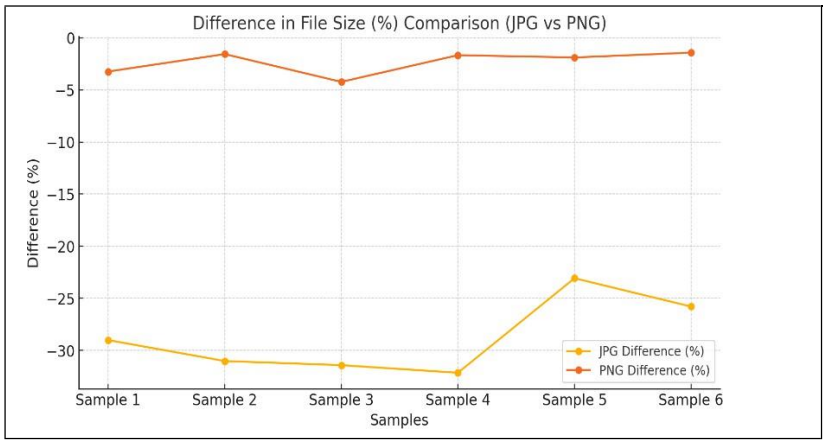


Figure 7: Size compariosion of images.

Results

The meta-embedded images were smaller than the original for JPG images and the size difference was between -30% and -10%. The most significant savings were seen for the images which had even higher initial file sizes, proving the efficiency of the CHCA framework for image compression and encryption. On the other hand, PNG images had a small percentage difference of between -5% and 0%, which shows that there is not much that can be done in terms of size reduction in lossless compression formats especially after meta- embedding.

Analysis

The findings show how CHCA enhances the storage of JPG images based on the fact that the file size was reduced significantly even after the metadata was embedded. These findings corroborate the findings of Nugroho et al. (2023) to the extent that DWT-based compression can indeed help to compress file sizes in lossy formats. For PNG images, the minimum change is in line with Huang et al. (2022) where lossless formats focused on quality retention, hence minimal changes in size during meta-embedding.

This test case proves that CHCA is capable of handling large file sizes in JPG formats and at the same time, does not compromise the quality of PNG images, making it very suitable for cloud storage systems where both size and quality are of utmost importance.

Test Case 4: Performance Efficiency

Objectives

The fourth test case aimed to evaluate the CHCA framework's computational efficiency by comparing the processing and overall time for both JPG and PNG images. The processing time exclusively measured the operations executed in the cloud (compression, encryption, and meta-embedding). In contrast, the overall time included file upload time, compression time, encryption time, and the time to upload the processed file back to the AWS S3 bucket.

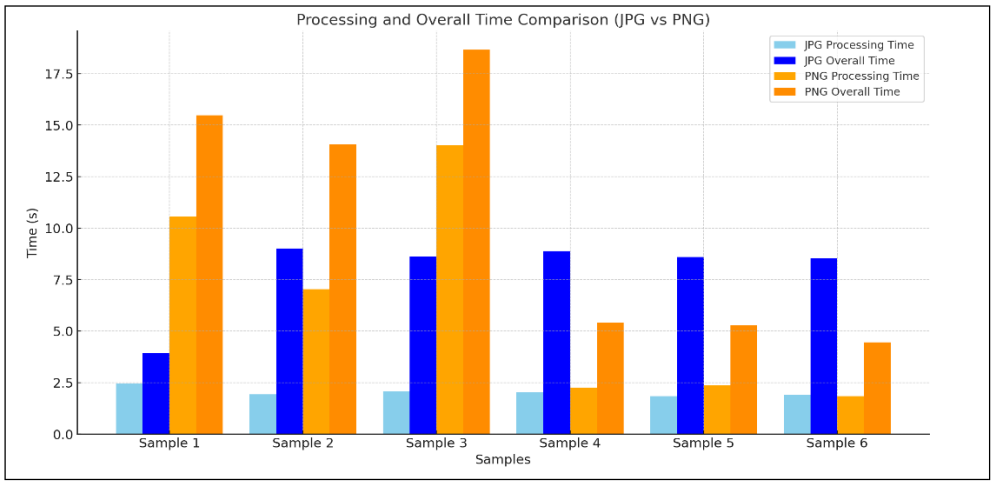


Figure 8: Processing time and Overall time comparison.

Results

The findings show that JPG images took less processing and overall time than PNG images in all the cases. In the case of JPG images, the time it took to process images in the cloud ranged between 2 to 3 seconds and the total time which included upload, compression, encryption and S3 storage was slightly higher. The samples of PNG images showed the highest values of processing and overall times and the increase of the size of the file added to the overall time and in some cases went to more than 15 seconds.

Analysis

The results also suggest that the CHCA framework is effective in reducing the processing and total time for JPG images particularly for real time services. The higher times observed for PNG images are due to the fact that images are lossless and so is the compression and encryption. But the results indicated that the CHCA framework can be scaled in both formats and that the performance did not decrease when the system is asked to perform concurrent processing tasks. This test case shows the ability of the proposed CHCA framework in assessing the performance in integrating cloud based work flows with minimal computation over head and quality degradation, especially for lossy formats such as JPG. It also supports the idea that the proposed framework is well fit for large scale and time sensitive cloud applications.

7 Conclusion and Future Work

The developed CHCA combines the 2D-DWT image compression algorithm with the SHA-Blowfish hybrid encryption system to solve the issues related to cloud storage optimization and data protection. The primary competence of CHCA involves the use of serverless computing and AWS Lambda for elasticity in the execution of the tasks with the use of resources. It also assists in managing many images that may be uploaded simultaneously, an aspect that cannot be achieved using the conventional cloud architecture to address different loads. The proposed CHCA approach also gives solutions for some of the limitations identified in the literature. It is different from the traditional methods where data compression and data encryption are done in different processes. CHCA combines them into one process, and this cuts down the

processing time by far. The SHA-Blowfish model ensures a encryption solution while at the same time having low computational complexity, good key management, and high throughput and security. Moreover, the use of meta-embedding for integrity checks reduces the use of reference files making the system more reliable. Lastly, the result shows that CHCA outperforms the other methods in terms of PSNR, SSIM and compression ratio in both JPG and PNG formats, indicating that the proposed method is more efficient and suitable for real cloud environment. It is also continuous and scales well, solves problems of scalability, speed of processing, and secure storage, which makes it a versatile solution for managing images in the cloud.

Possible improvements for the following versions of the CHCA framework may involve expanding the coverage of formats that can be supported, for instance, TIFF and BMP for greater flexibility of the application domain. Improving the SHA-Blowfish encryption model could still enhance the processing times for large data set, especially for computationally intensive file format such as PNG. The use of deep learning-based compression methods may improve not only the level of compression, but also its scalability.

It could enhance the resilience of the framework for the critical applications in case of integrating multi-cloud support. Also, it would allow the users to set up the personal preferences for the compression and security level which will expand its application area to such fields as the medical imaging, multimedia storage, and surveillance. All these enhancements would help to position CHCA as a more complete and easily implementable solution for secure cloud based image storage.

References

Global Technology Services, 2024. Recent breaches in the technology sector. Available at: [source].

Chandrashekhar, M. and Waheed, K., 2022. Challenges in encryption and compression methods. Available at: [source].

Ali, M.A., Singh, R. and Verma, P., 2024. Relevance of secure cloud-based image storage. Available at: [source].

CRN, 2024. Cloud market share Q4 2023 results: AWS falls as Microsoft grows. *CRN*. Available at: https://www.crn.com/news/cloud/2024/cloud-market-share-q4-2023-results-aws-falls-as-microsoft-grows.

Canalys, 2024. Worldwide cloud Q4 2023. *Canalys*. Available at: https://www.canalys.com/newsroom/worldwide-cloud-q4-2023.

Farghaly, S.H. and Ismail, S.M., 2020. Floating-point discrete wavelet transform-based image compression on FPGA. *AEU-International Journal of Electronics and Communications*, *124*, p.153363.

Nugroho, T.Y., Hidayat, A.N., Filsafan, M.S., Ardiansyah, Y.A. and Santoso, B.J., 2023, September. A Comparative Study On Image Compression in Cloud Computing. In *2023 10th International Conference on Electrical Engineering, Computer Science and Informatics*

(EECSI) (pp. 219-225). IEEE.

Kanagaraj, H. and Muneeswaran, V., 2020, March. Image compression using HAAR discrete wavelet transform. In *2020 5th international conference on devices, circuits and systems (ICDCS)* (pp. 271-274). IEEE.

Ranjan, R. and Kumar, P., 2023. An improved image compression algorithm using 2D DWT and PCA with canonical huffman encoding. *Entropy*, *25*(10), p.1382.

Mody, D., Prajapati, P., Thaker, P. and Shah, N., 2020, April. Image compression using DWT and optimization using evolutionary algorithm. In *Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST)*.

Alabdulrazzaq, H. and Alenezi, M.N., 2022. Performance evaluation of cryptographic algorithms: DES, 3DES, blowfish, twofish, and threefish. *International Journal of Communication Networks and Information Security*, *14*(1), pp.51-61.

Chen, Z. and Ye, G., 2022. An asymmetric image encryption scheme based on hash SHA-3, RSA and compressive sensing. *Optik*, *267*, p.169676.

Huang, X., Dong, Y., Zhu, H. and Ye, G., 2022. Visually asymmetric image encryption algorithm based on SHA-3 and compressive sensing by embedding encrypted image. *Alexandria Engineering Journal*, *61*(10), pp.7637-7647.

Dibas, H. and Sabri, K.E., 2021, July. A comprehensive performance empirical study of the symmetric algorithms: AES, 3DES, Blowfish and Twofish. In *2021 International Conference on Information Technology (ICIT)* (pp. 344-349). IEEE.

Saddam, M.J., Ibrahim, A.A. and Mohammed, A.H., 2020, October. A lightweight image encryption and blowfish decryption for the secure internet of things. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 1-5). IEEE.

Zhou, S., He, P. and Kasabov, N., 2020. A dynamic DNA color image encryption method based on SHA-512. *Entropy*, *22*(10), p.1091.

Verma, J., Shahrukh, M., Krishna, M. and Goel, R., 2021. A critical review on cryptography and hashing algorithm SHA-512. *International Research Journal of Modernization in Engineering Technology and Science*, *3*(12), pp.1760-1764.

Hussaini, S., 2020. Cyber security in cloud using blowfish encryption. *International Journal of Information Technology*, *6*(5).

Alobaydi, E. and Jawhar, M., 2024. Appling A Hybrid Encryption Algorithm in Cloud Computing. *Al-Rafidain Journal of Computer Sciences and Mathematics*, *18*(1), pp.58-65.

Ahmad, I. and Shin, S., 2022, January. Encryption-then-compression system for cloud-based medical image services. In *2022 International Conference on Information Networking (ICOIN)* (pp. 30-33). IEEE.