Dynamic Encryption-Based Cloud Security Model using Facial Image and Password-based Key Generation for Multimedia Data

Naima Sultana Ayesha[1], Mehrin Anannya[1], Md Biplob Hosen[1,2], Rashed Mazumder[1]
[1]*Institute of Information Technology*, *Jahangirnagar University*, Savar, Dhaka, Bangladesh
[2]*Department of Information Systems*, *University of Maryland Baltimore County*, Maryland, United States

Abstract—**In this cloud-dependent era, various security techniques, such as encryption, steganography, and hybrid approaches, have been utilized in cloud computing to enhance security, maintain enormous storage capacity, and provide ease of access. However, the absence of data-type-specific encryption and decryption procedures renders multimedia data vulnerable. To address this issue, this study presents a dynamic encryption-based security architecture that adapts encryption methods to any file type, using keys generated from facial images and passwords. Four diverse datasets are created, each with a consistent size of 2GB, containing varying combinations of image, audio (MP3 and MPEG), video, text, CSV, PPT, and PDF files, to implement the proposed methodology. AES is used to encrypt image data, AES-CTR is employed for audio or video data to meet real-time streaming needs, and Blowfish is used for other types of data. Performance analysis on all four datasets is conducted using AWS servers, where DATASET-1 demonstrates the best performance compared to the others.**

Index Terms—**Cloud Data Security, Key Generation, Dynamic Encryption.**

I. Introduction

There are numerous methods to ensure the highest level of data security in the cloud, including encryption, steganography, authentication, the use of firewalls and antimalware software, data backup, and monitoring of data access, among others. The categorization of biometric authentication is discussed in [1]. By utilizing biometric data, which are inherently difficult to replicate or compromise, this method not only enhances data security but also simplifies the key management process. Data integrity is maintained through cloud encryption techniques such as RSA, AES, and steganography [2], [3]. Digital images are the most commonly used cover media for steganography, with various methods depending on the embedding domain, size, and type of retrieval [4], [5]. The effectiveness of hybrid techniques that combine steganography and encryption is demonstrated in [6]–[8]. Text-based approaches are inadequate for large files, which presents challenges for multimedia encryption [9]. A study [10] examines the drawbacks and offers recommendations for future work, such as enhanced key generation, blockchain utilization, personalized encryption, and multispectral data encryption.

In cloud security, the use of biometric features for key generation and effective key management significantly enhances system security. Utilizing multimedia data necessitates research to improve security by leveraging facial images and adaptive encryption methods. Addressing existing encryption issues is essential to provide robust security for multimedia content using facial traits and flexible encryption techniques.

The limitations and challenges identified in current research give rise to several issues in the fields of cloud security and multimedia encryption. Relying on a single biometric feature for key generation can undermine cloud security if additional factors are not incorporated. Furthermore, the absence of data-type-specific encryption techniques creates vulnerabilities for data stored in the cloud.

Hybrid approaches introduce complexity, which can negatively impact key management, system performance, and data capacity. The proposed work aims to achieve robust key generation by combining facial images and passwords, while developing flexible data-type-specific encryption and decryption methods to ensure maximum security for multimedia data. Additionally, diverse datasets containing various data types will be created to validate the proposed methodology. This paper introduces a novel method to establish a robust security framework for cloud data by incorporating key generation using facial images and passwords, along with flexible encryption techniques.

The rest of the paper is structured as follows: Section II presents a literature review on related work. Section III introduces the proposed system model, including the system architecture and a description of each stage. Section IV analyzes and documents the results of the proposed system. Finally, Section V provides the conclusion of the proposed model.

II. Literature Review

In the field of cloud computing, data security has been a prominent area of research, with various approaches, such as steganography, encryption, authentication, and hybrid methods, being developed. A summary of different studies on data security reveals a range of methodologies and their limitations. Markandey et al. [11] introduced an approach focused on information ownership, which reduces input/output costs and allows clients to verify data integrity on untrusted servers. However, their study does not address future research on data classification or secure, trust-based cloud solutions. Shabbir et al. [12] proposed a health information security system using the Modular Encryption Standard (MES), based on layered security measures. However, their method does not accommodate image-oriented datasets and is limited to textual data encryption. Farid et al. [13] suggested a new identity management paradigm for IoT and cloud-based healthcare, utilizing multimodal encrypted biometric traits. Despite this, their approach does not fully address classical identity-based security issues, such as replay and man-in-the-middle attacks. Future research is needed to test the model's authentication in various scenarios.

Sannidhan et al. [14] proposed enhancing online transaction security using a pseudo-random key sequence based on facial features extracted with the SURF technique. The main limitation here is the reliance on a single feature extraction technique, which may limit the randomness of the key. Hossain et al. [15] combined AES encryption with fingerprint biometrics and one-time passwords for cloud data security. However, their method's reliance on fingerprints indicates a need for more diverse biometric solutions. Hosny et al. [10] conducted an in-depth analysis of multimedia encryption, identifying several shortcomings and suggesting improvements, such as better key generation and blockchain utilization. They pointed out issues like the complexity of DNA encryption, limited variation in cellular automata, and a lack of balance between security and complexity. Yasser et al. [16] explored new chaotic-based multimedia encryption techniques using 2D modification models and proposed future research directions, including the incorporation of additional chaotic maps and security layers.

Tajammul et al. [17] presented an algorithm that generates a key from the supplied data and encrypts it accordingly. Their future work aims to integrate data compression for more efficient cloud storage. Corpuz et al. [18] modified the Blowfish encryption algorithm to cater to various data formats. However, the processing time for Excel data remains a limitation. Khwailleh et al. [19] introduced a dynamic encryption technique that adjusts encryption keys based on the data type. Their approach is expected to improve with further research, especially in the areas of image classification and encryption for health platforms. Namasudra et al. [20] proposed a DNA-based encryption strategy that creates a strong 1024-bit key by combining user characteristics and passwords for large multimedia files. However, there is significant room for improvement in authentication procedures within cloud computing. Sajay et al. [21] suggested a hybrid technique combining homomorphic encryption and Blowfish encryption to address cloud security issues, anticipating future advancements in algorithms and secure storage solutions.

Finally, Adee et al. [6] proposed a four-step data security paradigm incorporating Rivest-Shamir-Adleman (RSA), Advanced Encryption Standard (AES), identity-based encryption, and Least Significant Bit (LSB) steganography. They also highlighted the need for additional research to improve the combination and security of multimedia data.

III. System Methodology

In this work, a cloud security model is proposed. Figure 1 illustrates the overall architecture of the system in brief.

In this system, distinctive facial features are extracted from face images, and passwords are provided by the user to create the keys needed for encrypting and decrypting data. Depending on the type of data being encrypted, the system integrates adaptive encryption techniques. AES-ECB is employed for image data, AES-CTR is used for audio and video data to support real-time streaming or random access playback, and flexible Blowfish encryption is applied for other data categories. Based on the required file name, data is retrieved and decrypted according to its type using the appropriate encryption key.

A. Key Generation

This proposed work generates keys by combining a password-based key with a key derived from facial features. This process is illustrated in Figure 2. A pre-trained model, 'shape_predictor_68_face_landmarks.dat', is used to initialize the face detector and landmark predictor. The predictor first detects faces in the image and then identifies 68 distinct facial landmarks. The SHA-256 hash value of the encoded features, along with the given password, are generated. These two hash values are then combined bitwise using XOR, with their corresponding bytes, to produce a final key, which serves as the security key for both encryption and decryption.

B. Encryption and Decryption Based on Data Type

In this stage, different encryption-decryption methods are implemented based on the data type, all using the same generated key. For encrypting pixel block-structured image data, AES in ECB mode is effective, as it provides reliable encryption with easy decoding. During cloud uploads, padding ensures that the plaintext aligns with the AES block size. Similarly, AES in CTR mode efficiently encrypts audio data that is continuously transmitted, enabling fast playback and seeking. PKCS7 padding ensures the plaintext is properly sized for encryption, which can be decrypted with the provided AES key. With the same key and nonce used for encryption, AES-CTR's stream cipher functionality supports efficient streaming for video content. Finally, the Blowfish algorithm strikes a balance between speed and security while offering flexibility in processing plaintexts of various lengths. It uses initialization vectors and padding to ensure the block size is appropriate, and during decryption, any padding applied is removed to yield the final, unpadded output.

C. Dataset Description

Four diverse datasets, each with a size of 1.2GB and containing different combinations of data types, have been created to implement the proposed methodology. These datasets are available in [22]. The system operates based on data type, and thus, various types of data, including image, audio, video, CSV, PPT, PDF, and text, were collected to create the customized datasets. The details of these datasets are provided in Table I.

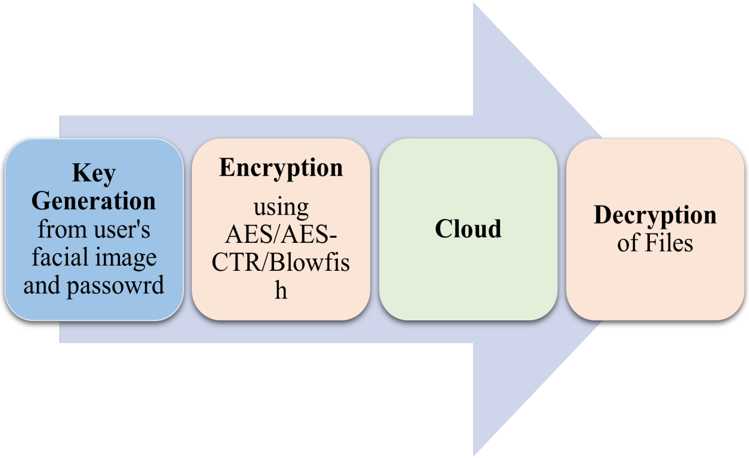
Key Generation from user's facial image and passowrd
Encryption using AES/AES-CTR/Blowfis h
Cloud
Decryption of Files

Fig. 1. Overview of system architecture.

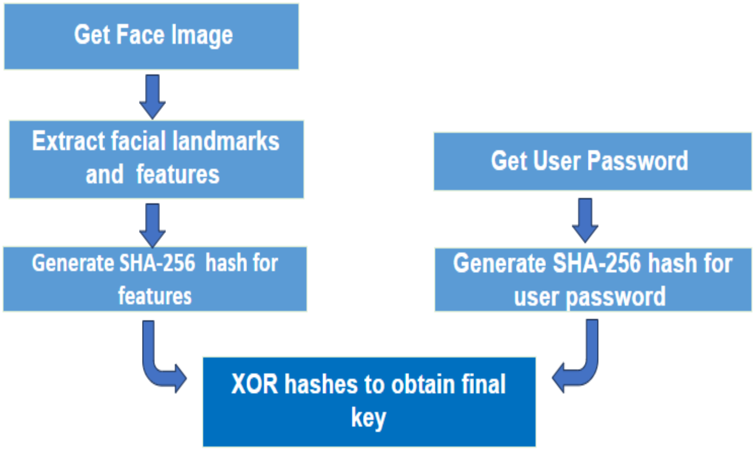
Get Face Image
Extract facial landmarks and features
Generate SHA-256 hash for features
Get User Password
Generate SHA-256 hash for user password
XOR hashes to obtain final key

Fig. 2. Key Generation Process.

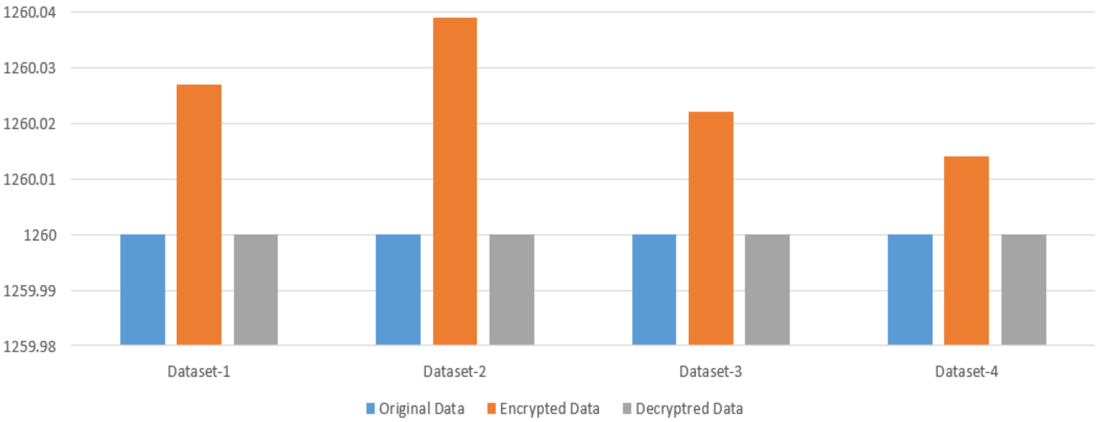

Fig. 3. Size variation among original, encrypted, and decrypted data.

TABLE I
DETAILS OF DATASETS

Dataset	Image (mb)	Audio-Video (mb)	Others(mb)
1	400	400	400
2	600	300	300
3	300	600	300
4	300	300	600

IV. Result and Discussion

The proposed architecture was developed in an x64-based PC environment featuring an Intel(R) Core(TM) i5-6200U processor, 8GB of RAM, and running the Windows 11 operating system. The system uses Python 3.10.9 and Conda 23.3.1 for development. Using the generated key, four datasets were successfully encrypted and uploaded to a specific S3 bucket. Data retrieval and decryption are performed using corresponding keys and data-type-specific decryption methods.

As shown in Figure 3, the original and decrypted data sizes are identical, indicating no data loss during the encryption and decryption process. After encryption, the size of the data increased for all datasets, with Dataset-2 resulting in the largest increase in size.

Several evaluation measures, listed below, have been considered to provide a comprehensive performance study of the proposed model:

A. Execution Time for Key Generation

This measure evaluates the time (in seconds) taken by the proposed framework's key generation operations, based on varying image sizes (5KB, 55KB, 150KB, 400KB, 850KB, 2MB, 5MB) and different password lengths (3, 6, 8, 10, 12). The time required to generate the key is presented in Table II.

It can be observed that for the same image size, the validation time for key generation remains minimal despite varying password lengths. However, as the image size increases, the key generation time also increases for the same password, and this variation becomes noticeable. Overall, the key generation time is minimal for smaller image sizes.

B. Entropy Analysis

The entropy is calculated based on the probabilities of '1' and '0' occurring in the binary data. The mathematical equations for this calculation are provided below:

$$\text{probability\_ones} = \frac{\text{ones\_count}}{\text{total\_bits}} \tag{1}$$

$$\text{probability\_zeroes} = \frac{\text{zeroes\_count}}{\text{total\_bits}} \tag{2}$$

$$\begin{aligned}\text{entropy} = &-(\text{probability\_ones} \cdot \log_2(\text{probability\_ones}) \\ &+\text{probability\_zeroes} \cdot \log_2(\text{probability\_zeroes}))\end{aligned} \tag{3}$$

$$\text{max\_entropy} = -(0.5 \cdot \log_2(0.5) + 0.5 \cdot \log_2(0.5)) \tag{4}$$

$$\text{normalized\_entropy} = \frac{\text{entropy}}{\text{max\_entropy}} \tag{5}$$

Equations (1) and (2) calculate the probabilities of obtaining "one" and "zero" in a sequence of bits. Equation (3) calculates the entropy, while equation (4) computes the maximum entropy for the binary data of the key. Equation (5) measures the normalized entropy by dividing the calculated entropy by the maximum entropy. The normalized entropy for the binary data of the key, as calculated in the assessment of the proposed framework, is shown in Table III, indicating the transition from unpredictability (1) to predictability (0). Since the entropy for key generation across different image and password size combinations is close to 1, it can be concluded that the key is highly unpredictable.

C. Brute Force Attack Analysis

A brute-force attack and simulation is a methodical, time-consuming process that tests every potential key in the key space to find the correct one. This approach is used to assess the proposed key-generating technique for different image sizes and password lengths. The time required for this simulation is shown in Table IV. The simulation takes varying amounts of time for different image sizes and password lengths, but it was unable to determine the password. This indicates that the key generation process is secure and that the generated key is strong enough to withstand such attacks.

D. Encryption / Decryption Performance Analysis

The performance analysis of the proposed method includes measurements of encryption and decryption time, aggregate throughput, size overhead, and other factors. The relevant mathematical equations are provided below:

$$\text{aggregate\_throughput} = \frac{\text{data\_size}}{\text{total\_time}} \tag{6}$$

Equation (6) calculates the aggregate throughput by dividing the total data size of the dataset by the total taken time.

$$\text{size\_overhead} = \frac{\text{length(crypted\_data)}}{\text{data\_size}} \tag{7}$$

Equation (7) measures the size overhead by subtracting the original data size from the encrypted data size. The total size overhead for any dataset is calculated by summing the size overheads of each encrypted data item within the dataset. The performance study of the proposed method, applied to datasets with various data combinations, is presented in Table V. The proposed method performs best on Dataset-3, which is enriched with audio and video data, followed by Dataset-2, which contains the most image data. Dataset-2 exhibits the highest size overhead, while Dataset-3 achieves the maximum aggregate throughput with the shortest encryption and decryption times. Dataset-1, with more balanced data sizes, provides better results overall. Dataset-4, which includes a broader variety of data types, shows longer processing times but lower size overhead and aggregate throughput.

TABLE II
TIME MEASUREMENT FOR KEY GENERATION

Image Size / Password Length	3	6	8	10	12
5KB	0.01568	0.01598	0.01667	0.01693	0.01779
55KB	0.18755	0.21577	0.21776	0.23578	0.25242
150KB	0.34375	0.38859	0.61444	0.63400	0.65987
400KB	0.53120	0.67098	0.67676	0.701	0.70280
850KB	0.61328	0.75904	0.77574	0.84460	0.93464
2.5MB	2.19598	2.25511	2.24992	2.30883	2.31677
5MB	3.33198	3.45086	3.54378	3.68518	3.76542

TABLE III
ENTROPY ANALYSIS FOR KEY GENERATION

Image Size / Password Length	3	6	8	10	12
5KB	0.997	0.998	0.998	0.999	0.997
55KB	0.997	0.996	0.992	0.999	0.999
150KB	0.999	0.999	0.999	0.999	0.994
400KB	0.998	0.999	0.999	0.998	0.999
850KB	0.998	0.994	0.976	0.995	0.999
2.5MB	0.997	0.997	0.994	1.00	1.000
5MB	0.999	0.999	0.999	0.999	0.999

TABLE IV
TIME TAKEN FOR BRUTE FORCE SIMULATION

Image Size / Password Length	3	6	8	10	12
5KB	55.49	68.79	66.12	64.78	65.57
55KB	57.74	68.75	62.18	65.98	67.83
150KB	60.28	89.40	90.95	89.33	91.61
400KB	60.97	86.80	91.77	92.30	98.25
850KB	66.36	89.57	90.30	90.49	92.76
2.5MB	63.44	66.17	64.88	66..99	67.89
5MB	65.46	68.49	66.26	64.22	65.42

TABLE V
PERFORMANCE ANALYSIS ON DATASETS

Dataset	Total Datasize (mb)	Encryption time (sec)	Decryption time (sec)	Aggregate Throughput (mb/sec)	Size-overhead (mb)
1	1260	10.94	10.02	115.19	0.027
2	1260	8.89	8.46	141.70	0.039
3	1260	8.45	7.53	148.97	0.022
4	1260	12.15	12.25	103.63	0.014

V. CONCLUSION

This study proposes a dynamic encryption-based security architecture for cloud data, utilizing facial images and passwords for key generation. The system employs AES for image data, AES-CTR for audio/video data, and Blowfish for other data types, providing flexible encryption tailored to different content types. Performance analysis across four diverse datasets demonstrates that the proposed method enhances security for multimedia data, offering efficient encryption and decryption with minimal data loss. Despite the success in securing data, the method does face challenges such as potential accuracy issues from variations in image quality and environmental factors, privacy concerns regarding biometric data storage, and increased computational overhead from adaptive encryption.

To address these limitations, future research will explore biometric liveness detection beyond facial features and investigate privacy-preserving solutions for biometric data storage. Further work will also focus on optimizing encryption algorithms to reduce computational overhead without compromising security. Additionally, entropy analysis, brute-force simulations, and comparative studies using diverse datasets will be conducted to further validate and enhance the robustness of the proposed security model.

REFERENCES

[1] P. Padma and S. Srinivasan, "A survey on biometric based authentication in cloud computing," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 1, pp. 1–5, 2016.

[2] F. F. Moghaddam, O. Karimi, and M. T. Alrashdan, "A comparative study of applying real-time encryption in cloud computing environments," in *2013 IEEE 2nd International Conference on Cloud Networking (CloudNet)*, pp. 185–189, IEEE, 2013.

[3] S. K. Singh, P. Manjhi, and R. Tiwari, "Cloud computing security using steganography," *Journal of Emerging Technologies and Innovative Research,(JETIR)*, vol. 6, pp. 923–927, 2019.

[4] A. Y. AlKhamese, W. R. Shabana, and I. M. Hanafy, "Data security in cloud computing using steganography: a review," in *2019 International conference on innovative trends in computer engineering (ITCE)*, pp. 549–558, IEEE, 2019.

[5] I. J. Kadhim, P. Premaratne, P. J. Vial, and B. Halloran, "Comprehensive survey of image steganography: Techniques, evaluations, and trends in future research," *Neurocomputing*, vol. 335, pp. 299–326, 2019.

[6] R. Adee and H. Mouratidis, "A dynamic four-step data security model for data in cloud computing based on cryptography and steganography," *Sensors*, vol. 22, no. 3, p. 1109, 2022.

[7] M. S. Abbas, S. S. Mahdi, and S. A. Hussien, "Security improvement of cloud data using hybrid cryptography and steganography," in *2020 international conference on computer science and software engineering (CSASE)*, pp. 123–127, IEEE, 2020.

[8] Z. N. Al-Kateeb, M. J. Al-Shamdeen, and F. S. Al-Mukhtar, "Encryption and steganography a secret data using circle shapes in colored images," in *Journal of Physics: Conference Series*, vol. 1591, p. 012019, IOP Publishing, 2020.

[9] X. Liu and A. M. Eskicioglu, "Selective encryption of multimedia content in distribution networks: Challenges and new directions," *IASTED Communications, Internet & Information Technology (CIIT), USA*, p. 2003, 2003.

[10] K. M. Hosny, M. A. Zaki, N. A. Lashin, M. M. Fouda, and H. M. Hamza, "Multimedia security using encryption: A survey," *IEEE Access*, vol. 11, pp. 63027–63056, 2023.

[11] A. Markandey, P. Dhamdhere, and Y. Gajmal, "Data access security in cloud computing: A review," in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 633–636, IEEE, 2018.

[12] M. Shabbir, A. Shabbir, C. Iwendi, A. R. Javed, M. Rizwan, N. Herencsar, and J. C.-W. Lin, "Enhancing security of health information using modular encryption standard in mobile cloud computing," *IEEE Access*, vol. 9, pp. 8820–8834, 2021.

[13] F. Farid, M. Elkhodr, F. Sabrina, F. Ahamed, and E. Gide, "A smart biometric identity management framework for personalised iot and cloud computing-based healthcare services," *Sensors*, vol. 21, no. 2, p. 552, 2021.

[14] M. Sannidhan, K. Sudeepa, J. E. Martis, and A. Bhandary, "A novel key generation approach based on facial image features for stream cipher system," in *2020 third international conference on smart systems and inventive technology (ICSSIT)*, pp. 956–962, IEEE, 2020.

[15] M. A. Hossain and M. A. Al Hasan, "Improving cloud data security through hybrid verification technique based on biometrics and encryption system," *International Journal of Computers and Applications*, vol. 44, no. 5, pp. 455–464, 2022.

[16] I. Yasser, M. A. Mohamed, A. S. Samra, and F. Khalifa, "A chaotic-based encryption/decryption framework for secure multimedia communications," *Entropy*, vol. 22, no. 11, p. 1253, 2020.

[17] M. Tajammul and R. Parveen, "Auto encryption algorithm for uploading data on cloud storage," *International Journal of Information Technology*, vol. 12, no. 3, pp. 831–837, 2020.

[18] R. R. Corpuz, B. D. Gerardo, and R. P. Medina, "Using a modified approach of blowfish algorithm for data security in cloud computing," in *Proceedings of the 6th International Conference on Information Technology: IoT and Smart City*, pp. 157–162, 2018.

[19] D. Khwailleh and F. Al-Balas, "A dynamic data encryption method based on addressing the data importance on the internet of things," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, p. 2139, 2022.

[20] S. Namasudra, R. Chakraborty, A. Majumder, and N. R. Moparthi, "Securing multimedia by using dna-based encryption in the cloud computing environment," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 3s, pp. 1–19, 2020.

[21] K. Sajay, S. S. Babu, and Y. Vijayalakshmi, "Enhancing the security of cloud data using hybrid encryption algorithm," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2019.

[22] GitHub, "Dataset repository." https://github.com/naisha275/Dataset, 2024. Accessed: 6 May 2024.