Available online at https://journal.rescollacomm.com/index.php/ijqrm/index

International Journal of Quantitative Research and Modeling

e-ISSN 2721-477X
p-ISSN 2722-5046

Vol. 6, No. 3, pp. 364-376, 2025

Digital Image Security with AES and Blowfish Double Encryption

Iqbal Dwi Nulhakim[1*], Asep Id Hadiana[2], Melina[3]

[1,2,3] *Department of Informatics, Faculty of Science and Informatics, Universitas Jenderal Achmad Yani, Jl. Terusan Jend. Sudirman, Cimahi, West Java, 40525, Indonesia*

**Corresponding author email: iqbaldwin21@if.unjani.ac.id*

Abstract

Protection of digital images is becoming increasingly important with the growing use of images as a medium of information in various fields, particularly in the healthcare sector. Medical images such as Magnetic Resonance Imaging (MRI) contain sensitive information that requires extra security against unauthorized access and data manipulation. This study aims to design and build a digital image security system using a dual encryption approach and authenticity verification based on watermarking. The security process is carried out in two main stages. First, images with text-based watermarks are encrypted using the Advanced Encryption Standard (AES) algorithm to protect their visual content. Second, the AES key is re-encrypted using the Blowfish algorithm to prevent the key from being stored in plaintext, thereby creating an additional layer of protection. The watermark is embedded into the image using the Singular Value Decomposition (SVD) method and is first converted into a hash value using the SHA-256 algorithm, which serves to verify the integrity of the image after decryption. The testing was conducted using the public dataset "Brain Tumor Image Dataset (Semantic Segmentation)" from Kaggle, which consists of brain MRI images in .jpg and .png formats. The system evaluation encompassed functionality, data security, and process efficiency through system function testing, measurement of encrypted data randomness (entropy test), file penetration using OpenSSL, and performance analysis in terms of processing time and file size. The research results show that the system successfully implemented double encryption with a high entropy level (approaching 8.00) and resistance to penetration attacks. In terms of efficiency, the system achieved an average encryption time of 81.35 ms and decryption time of 13.68 ms with minimal file size increase. Integrity testing confirmed that the SVD-SHA256-based watermark remained intact after the encryption-decryption process, enabling verification of image authenticity. The developed system efficiently maintains the confidentiality and authenticity of digital images and can be applied in electronic medical record systems or sensitive digital archives.

Keywords: Digital images, double encryption, AES, Blowfish, SVD

1. Introduction

In the digital era, the protection of sensitive data has become a crucial issue, particularly for digital images that are widely used as a medium of information in fields such as healthcare, military, and finance. In the medical sector, images such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT-scan) play a vital role in electronic medical record (EMR) systems, supporting diagnostic processes and clinical decision-making. Consequently, ensuring the security of medical images is essential to prevent unauthorized access that could threaten both the privacy and integrity of the data (Nagamunthala & Manjula, 2023).

In addition to healthcare, the creative industry—including printing and image editing—also faces challenges in protecting visual data. Manipulation or leakage of digital images may harm intellectual property rights and damage the reputation of individuals or organizations ("Enhancing Video Encryption: AES and Blowfish Algorithms with Random Password Generation," 2023). Previous studies have shown that companies without adequate digital protection systems remain vulnerable to manipulation and image theft, highlighting the urgency of robust image security solutions (Malvi, n.d.).

Cryptography has been recognized as one of the most effective solutions for preventing unauthorized access to digital data (Dzikri Azhari Ali & Id Hadiana, 2024). Algorithms such as Triple DES (TDES) and Blowfish have been widely applied to strengthen system security, while advanced methods like the Advanced Encryption Standard (AES) are preferred for their high performance and reliability (Nagamunthala & Manjula, 2023). Nevertheless, most existing approaches focus solely on confidentiality, often neglecting mechanisms that ensure data authenticity after decryption. In this context, digital watermarking emerges as a complementary technique, enabling authentication and integrity verification without compromising the quality of diagnostic data (Haddad, n.d.; Zermi et al., 2021).

Several previous studies have attempted to combine encryption with steganography or watermarking to enhance image protection. For instance, research integrating AES with steganography demonstrated resilience against third-party applications, yet lacked mechanisms to validate image integrity after extraction (Fajriati Romli et al., n.d.). Other works applied combinations of AES with hashing methods to secure text documents but failed to address the unique challenges of visual data protection (Rahayu et al., 2024). Meanwhile, watermarking methods such as Singular Value Decomposition (SVD) have proven effective for embedding ownership information, but they often lack integrated encryption or key protection mechanisms (Solikhudin et al., n.d.).

Building on these findings, this study proposes the development of a dual-encryption system that applies AES to secure image content and Blowfish to protect the AES key, while incorporating watermarking based on SVD that is reinforced with SHA-256 hashing. This approach is designed not only to preserve the confidentiality of digital images but also to guarantee their authenticity and efficiency in terms of processing time and storage requirements. By addressing the limitations of earlier approaches, the research aims to provide a more comprehensive solution for securing digital images, with potential applications in medical imaging systems and other sensitive digital archives.

2. Literature Review

2.1. Cryptography

Cryptography is a fundamental technique for securing digital data by transforming readable information into an encrypted form that can only be accessed using a valid decryption key. The primary goals of cryptography are confidentiality, integrity, authentication, and non-repudiation, which are critical for protecting sensitive information both during storage and transmission (Melina et al., 2024). Modern cryptography is generally divided into two categories: symmetric encryption and asymmetric encryption. Symmetric algorithms such as AES, DES, and Blowfish rely on a single key for both encryption and decryption, while asymmetric algorithms like RSA use a public-private key pair. Furthermore, cryptographic hash functions such as SHA-256 provide integrity checks and digital signatures that are irreversible and collision-resistant (Nagamunthala & Manjula, 2023).

In the context of digital images, cryptography ensures that medical images, confidential documents, or multimedia data remain secure even if intercepted by unauthorized parties. However, encryption alone does not guarantee proof of authenticity. For this reason, cryptography is often combined with watermarking techniques to strengthen authenticity and ownership verification (Wayan Angga Wijaya Kusuma et al., 2021). This integration is particularly important in the medical field, where patient confidentiality and trustworthiness of diagnostic images must be preserved (Jamaluddin et al., 2021).

2.2. Watermarking

Watermarking refers to embedding hidden information into digital content, particularly images, with the goal of authentication, copyright protection, or tamper detection. A watermark can be either visible or invisible, depending on the application, and should remain robust against common image processing operations such as compression, resizing, filtering, or even malicious tampering. In medical imaging, watermarking plays a vital role in ensuring that data remains authentic even after being encrypted, transmitted, and decrypted (Zermi et al., 2021).

Several approaches to watermarking exist, including spatial-domain techniques (direct pixel manipulation) and transform-domain techniques, such as Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Singular Value Decomposition (SVD). Among these, SVD-based watermarking has demonstrated superior robustness against attacks because it modifies singular values of the image matrix, which represent intrinsic properties less affected by distortion. Recent advancements also explore deep learning–based watermarking methods, but SVD remains one of the most efficient techniques when the priority is maintaining high visual quality and resilience (Taj et al., 2024).

2.3. Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is one of the most widely used symmetric block ciphers, adopted as a global encryption standard by NIST in 2001 (Monica et al., 2024). AES operates on fixed-size blocks of 128 bits and supports key lengths of 128, 192, or 256 bits. Its encryption process involves multiple rounds of transformations, including SubBytes, ShiftRows, MixColumns, and AddRoundKey, which are designed to provide both confusion and diffusion, thereby protecting against cryptanalysis attacks. Due to its parallel structure and lightweight operations, AES achieves high performance in both hardware and software implementations, making it particularly suitable for real-time applications (Jamaluddin et al., 2021).

In the context of medical image security, AES has been extensively applied because of its balance between computational efficiency and cryptographic strength. The algorithm's deterministic block structure ensures that even a single-bit modification in the input produces a completely different ciphertext output, thereby protecting sensitive data from statistical analysis. Figure 1 illustrates the basic flow of AES encryption and decryption, emphasizing its transformation rounds. AES operates as a block cipher with a fixed block size of 128 bits (16 bytes). In AES-128, the

key length is also 128 bits, and the encryption process consists of 10 rounds of transformations, while AES-192 and AES-256 employ 12 and 14 rounds, respectively. Each round consists of four major transformations (And & Expert, 2022b):

1) SubBytes: a nonlinear byte substitution using an S-Box to provide nonlinearity and diffusion.
2) ShiftRows: cyclic shifting of rows in the state matrix to spread data across columns.
3) MixColumns: a linear mixing operation based on matrix multiplication that combines bytes within each column.
4) AddRoundKey: a bitwise XOR between the state and a round key derived from the main secret key.

Through these iterative rounds, AES ensures strong resistance against linear and differential cryptanalysis. Both encryption and decryption are performed on 128-bit data blocks, with key sizes of 128, 192, or 256 bits determining the number of rounds required. The structured yet efficient process makes AES a robust standard for securing digital images, where confidentiality and rapid processing are equally essential. The detailed encryption flow of AES is presented in Figure 1, showing the transformation steps in each round (And & Expert, 2022b).

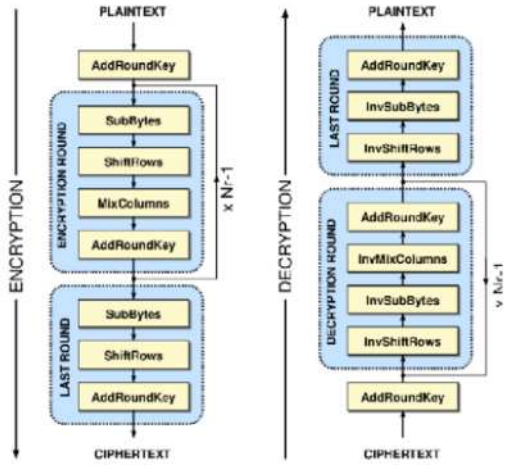

Figure 1: Tahapan Proses Enkripsi dan Dekripsi Algoritma AES

2.4. Blowfish

Blowfish is a symmetric block cipher algorithm with a fixed block size of 64 bits (8 bytes). It supports a variable key length ranging from 32 bits (4 bytes) to 448 bits (56 bytes), with a default size of 128 bits (16 bytes). Due to its flexibility in key size and relatively lightweight computation, Blowfish is widely adopted in applications that require secure but efficient encryption (And & Expert, 2022a).

The algorithm consists of two main components: key expansion and the encryption–decryption process. Key expansion must be performed first to generate the required subkeys, while the encryption and decryption operations are carried out using a Feistel network with 16 rounds. The encryption procedure of Blowfish can be described as follows (Fahriani & Rosyid, 2020):

1) Initialization of P-array with 18 subkeys (P1, P2, …, P18), each containing a 32-bit value.
2) Initialization of S-boxes, which are four substitution boxes (S1, S2, S3, S4), each consisting of 256 entries with 32-bit values.
3) A 64-bit plaintext block is taken as input. If the input does not meet the required size, padding is applied to fit the block length.
4) The 64-bit plaintext block is divided into two 32-bit halves, denoted as XL (left part) and XR (right part).
5) For each encryption round, the following operations are performed:
   - XL = XL XOR Pi
   - XR = XR XOR F(XL) where F is a nonlinear function that uses the initialized S-boxes.
   - Swap the values of XL and XR.
   - Repeat the process for a total of 16 rounds.
6) After completing the 16 rounds, a final step is performed:
   - XR = XR XOR P17
   - XL = XL XOR P18
7) The two halves (XL and XR) are concatenated to form the 64-bit ciphertext.

For the decryption process, the same sequence of steps is applied, but the P-array is used in reverse order. This property is an advantage of the Feistel structure, where the encryption and decryption operations are essentially identical, differing only in the key scheduling order. Figure 2 illustrates the overall process of Blowfish encryption and decryption, showing the repeated Feistel rounds and the way subkeys are applied.

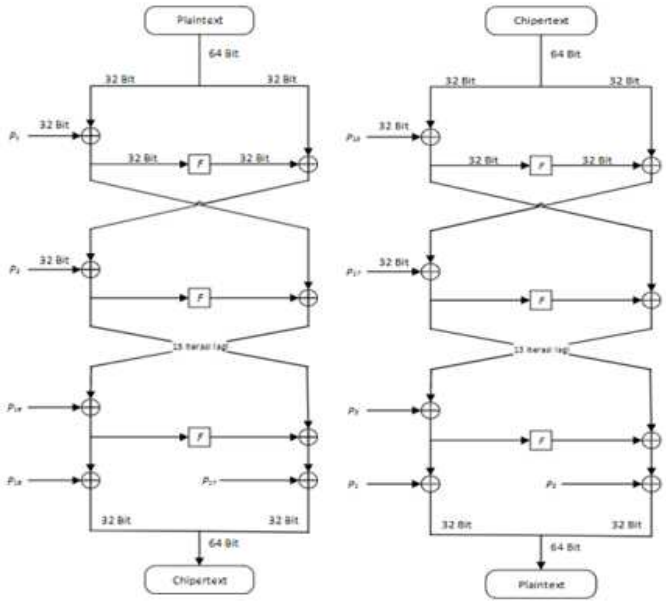

Figure 2: Tahapan Proses Enkripsi dan Dekripsi Algoritma Blowfish

2.5. Combination of AES and Blowfish

The integration of AES and Blowfish offers a layered security model that leverages the strengths of both algorithms. AES provides strong encryption for the actual image data, while Blowfish secures the AES key with its flexible key structure and Feistel-based transformations. This combination reduces the risk of single-point failure because even if the AES ciphertext is exposed, it cannot be decrypted without also recovering the Blowfish-protected key (Fahriani & Rosyid, 2019; Jamaluddin et al., 2021).

Previous studies have demonstrated that hybrid encryption approaches increase resistance against brute-force and differential attacks, while maintaining processing efficiency. Thus, combining AES and Blowfish provides an optimal balance of performance, scalability, and robustness for multimedia security, particularly in contexts where both data confidentiality and key protection are paramount (Jamaluddin et al., 2021).

2.6. Algoritma Hash SHA-256

SHA-256 is one of the most prominent members of the SHA-2 (Secure Hash Algorithm 2) family, designed by the National Security Agency (NSA) and standardized by NIST. The algorithm produces a fixed 256-bit hash value regardless of the size of the input, ensuring consistency in output length while providing strong collision resistance. Unlike encryption, hashing is a one-way operation, meaning the original input cannot be reconstructed from the hash value. One of the key features of SHA-256 is its avalanche effect, where even a single-bit change in the input results in a completely different hash output. This sensitivity makes it highly reliable for integrity verification, ensuring that no alteration—whether accidental or malicious—has occurred in the data (Ariska, 2022).

In the context of digital image security, SHA-256 is widely used to generate digital signatures, to verify the integrity of medical images, and to convert watermark text into a secure hash value before embedding it into an image. By embedding the hash rather than the original watermark text, the system strengthens both security and robustness, since the hash is computationally infeasible to reverse-engineer (Ariska, 2022).

2.7. Entropy Test

Entropy is a fundamental statistical measure used to evaluate the randomness and unpredictability of encrypted data. In cryptographic systems, an ideally encrypted image should resemble pure random noise, making it infeasible for attackers to deduce any patterns from the ciphertext. The entropy value is calculated using Shannon's information theory, defined as (Dirjen et al., 2017):

$$H(m) = \sum_{i=0}^{255} P(m_i).log_2 P(m_i). \quad (1)$$

where:

- m_i represents the possible intensity levels of a pixel (0–255 for an 8-bit grayscale image),
- $P(m_i)$ is the probability of occurrence of the pixel value mim_imi.

For an 8-bit image, the maximum possible entropy value is 8, indicating that all pixel values occur with equal probability. A value close to 8 suggests that the encrypted image has high randomness and does not preserve any statistical characteristics of the original image. On the other hand, values significantly lower than 8 indicate potential weaknesses in the encryption process, as residual patterns may remain detectable.

In image encryption research, entropy analysis is considered a benchmark test to validate the robustness of cryptographic algorithms against statistical and differential attacks. In this study, entropy analysis is applied to the encrypted (.bin) files to ensure the ciphertext does not reveal any exploitable structure of the original image.

2.8. Penetration Test

While entropy measures the statistical randomness of ciphertext, penetration testing evaluates the practical resilience of an encryption system against unauthorized access. It simulates real-world attack scenarios where an adversary attempts to decrypt encrypted data without the correct key. One of the commonly used tools for such testing is OpenSSL, which provides cryptographic command-line utilities to attempt decryption with incorrect keys or brute-force attacks. A secure system must return failure messages ("bad decrypt") when an invalid key is provided, ensuring that no partial or corrupted data can be reconstructed (Fajriati Romli et al., n.d.).

In digital image security, penetration testing serves as an essential complement to entropy analysis, because high statistical randomness alone does not guarantee resistance against practical attacks. By combining both methods, researchers can confirm that the encryption system is not only theoretically secure but also practically robust against adversaries with computational resources (Fajriati Romli et al., n.d.).

3. Materials and Methods

3.1. Research Approach

This research applied a quantitative experimental approach in developing and evaluating a secure system for digital medical images. The experimental method was chosen to objectively measure the performance and effectiveness of the proposed double encryption and watermarking scheme in realistic scenarios. The system integrates two symmetric cryptographic algorithms: AES-128 is used to encrypt watermarked images, while Blowfish secures the AES key itself. This layered mechanism strengthens confidentiality, ensuring that unauthorized users cannot access image data even if the main encryption key is compromised. Additionally, the system embeds ownership verification through a watermarking process based on Singular Value Decomposition (SVD). The watermark text is first hashed using SHA-256, then embedded into the image matrix, allowing later verification of integrity and authenticity.

3.2. Research Flow

The research process was structured in several stages, starting from theoretical review to experimental evaluation. The first stage involved a literature review, focusing on cryptographic algorithms, watermarking techniques, and methods for evaluating system security such as entropy analysis and penetration testing. The second stage was the requirement analysis and system design, which defined functional needs such as image upload, encryption, decryption, watermark embedding, and verification, as well as non-functional needs such as processing speed and storage efficiency. Based on these requirements, a system architecture was designed to include input, processing, and output layers.

The third stage was dataset collection and preprocessing, which ensured the selected MRI brain images were suitable for the system in terms of size and format. The fourth stage was system implementation, which used Python and Flask to develop a web-based platform that allows users to upload images, insert watermarks, provide encryption keys, and process encryption/decryption automatically. The fifth stage was system testing and evaluation, which included functional testing to confirm correct operations, security testing with entropy and penetration analysis, and performance testing by measuring time efficiency and file size changes. Finally, the sixth stage was the reporting of results, where findings were documented and compared with existing approaches to highlight strengths and limitations.

The complete research workflow is summarized in Figure 2, which shows each stage from literature review to final reporting. This diagram emphasizes the step-by-step process, ensuring that the experiment is reproducible and systematically structured.

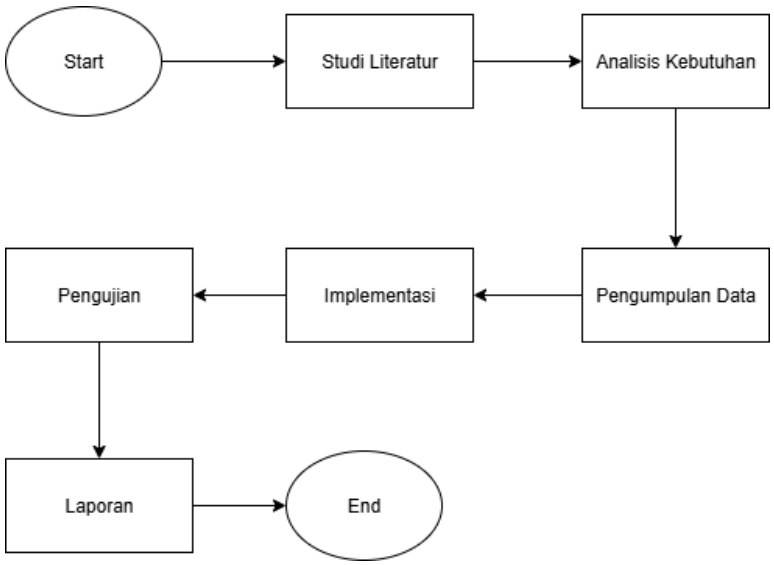

Figure 3: Research Flow

3.3. Dataset

The dataset used in this study was the Brain Tumor Image Dataset (Semantic Segmentation), publicly available on Kaggle. It consists of anonymized brain MRI images in .jpg and .png formats with file sizes ranging from 20 KB to 70 KB. These sizes are representative of typical medical images stored in electronic health records, making the dataset relevant for evaluating both encryption efficiency and security. Six representative images with varying file sizes were selected for testing to cover different storage and processing scenarios. No additional labeling or classification was required, since the focus of this research was not on medical diagnosis but rather on the security and integrity of the images.

Before testing, a preprocessing step was conducted to verify file formats, consistency, and size distribution. This ensured compatibility with the system and allowed encryption and decryption processes to be applied directly without further modifications. Using real MRI images rather than synthetic data strengthens the reliability of the evaluation, as the results reflect potential real-world deployment in healthcare environments.

Figure 3 provides examples of the selected MRI images, highlighting the range of file sizes and formats used for experimental testing.

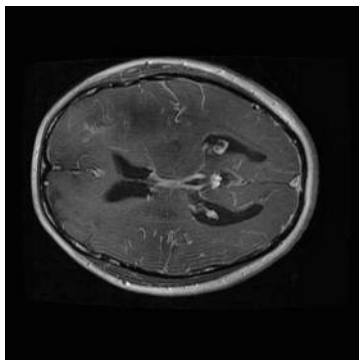

Figure 4: Dataset

3.4. System Design

The system was designed as a web-based application that integrates watermark embedding, dual-layer encryption, and verification features into a unified workflow. The design follows a three-layer structure. In the input layer, users are required to upload an MRI image, enter a textual watermark, and provide encryption keys: a 16-character AES key and a Blowfish key between 4 and 56 characters. In the processing layer, the system first hashes the watermark text using SHA-256, then embeds the hash into the image matrix using Singular Value Decomposition (SVD). Afterward, the watermarked image is encrypted with AES-128, while the AES key is encrypted separately using Blowfish. Both the encrypted image (.bin) and the encrypted key (in Base64 format) are generated. In the output layer, users can download the encrypted data, perform decryption by re-entering the correct keys, and verify the extracted watermark hash against the original to ensure authenticity.

The system was implemented using Python with the Flask framework, enabling interactive web-based access. Cryptographic operations were performed using the PyCryptodome library, while image manipulation used OpenCV.

Hashing was handled by Python's hashlib library. The design also included additional testing tools: entropy analysis to assess randomness of ciphertext, and penetration testing using OpenSSL to simulate brute-force or invalid key attempts. This multi-feature design ensures that the system not only secures images but also validates their authenticity and resists potential security attacks, making it suitable for deployment in environments such as cloud-based medical archives or electronic health record systems.

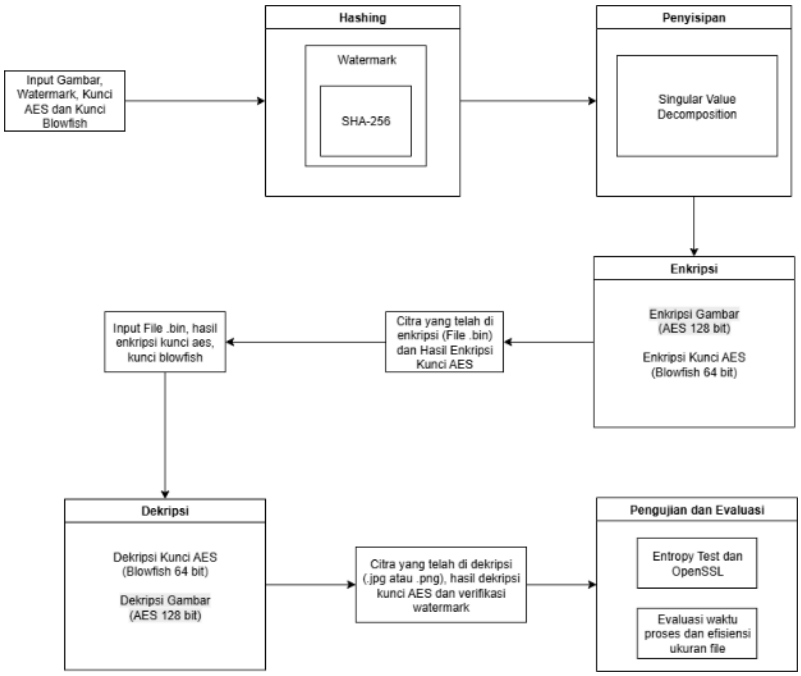

Figure 5: System Design

4. Results and Discussion

The developed system was implemented successfully as a web-based application on Windows 11 using Python and Flask, with AES for image encryption, Blowfish for key encryption, SHA-256 for watermark verification, and SVD for watermark embedding. The following subsections present the functional testing, security evaluation, and performance analysis of the system.

4.1. Functional Testing

Functional testing was carried out to verify that each feature operated as expected. Encryption consistently produced ciphertext files, decryption with valid keys restored the original images, and invalid keys failed to decrypt. Watermark embedding and verification also functioned correctly, with the extracted watermark matching the original hash value.

Table 1: Functional Testing Results

No	Expected Response	Actual Result	Status
1	System successfully performs encryption by uploading an image and receiving watermark, AES key, and Blowfish key inputs.	Match	Pass
2	System successfully generates .bin file and AES key encrypted with Blowfish.	Match	Pass
3	System successfully performs decryption using .bin file, encrypted AES key, and Blowfish key.	Match	Pass
4	System successfully restores the original AES key and accurately verifies the hash.	Match	Pass
5	System displays the entropy value of the encrypted file.	Match	Pass

As shown in Figure 1, the encryption page provides the interface where users can upload MRI images, insert watermark text, and input AES and Blowfish keys. The encryption result page (Figure 2) displays the generated

encrypted image file in *.bin* format along with the encrypted AES key in Base64, both of which can be downloaded by the user. Meanwhile, the decryption page (Figure 3) allows users to upload encrypted files and provide the required keys, and the decryption result page (Figure 4) presents the recovered image together with the extracted watermark and SHA-256 hash verification. These interfaces demonstrate that the system has successfully integrated encryption, decryption, and watermark verification in a user-friendly manner. In addition, the entropy test page (Figure 5) enables users to evaluate the randomness of encrypted files. The calculated entropy values approached 8.00, indicating that the encrypted images were highly randomized and resistant to statistical attacks.

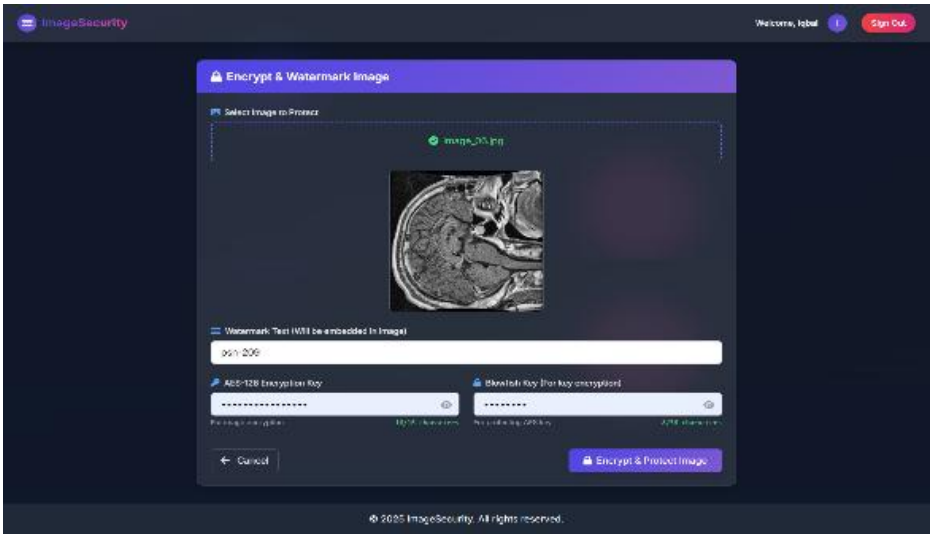

Figure 6: System interface for image encryption and watermark embedding

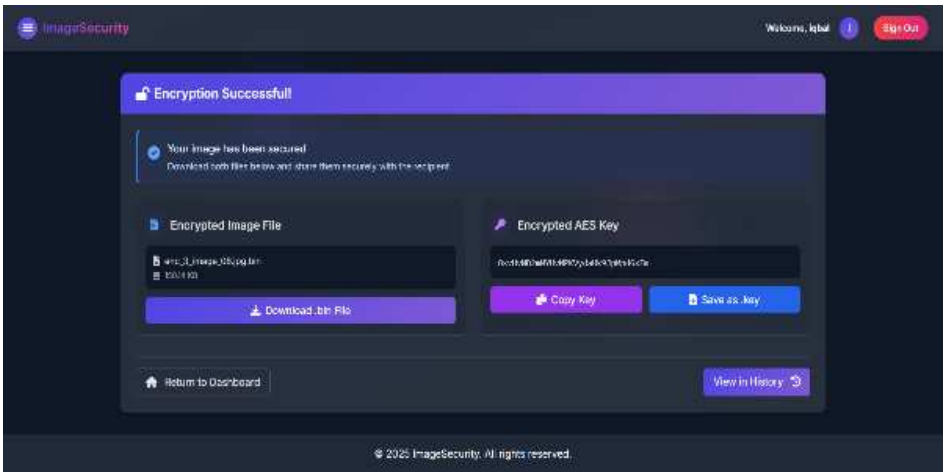

Figure 7: Interface of encryption result page

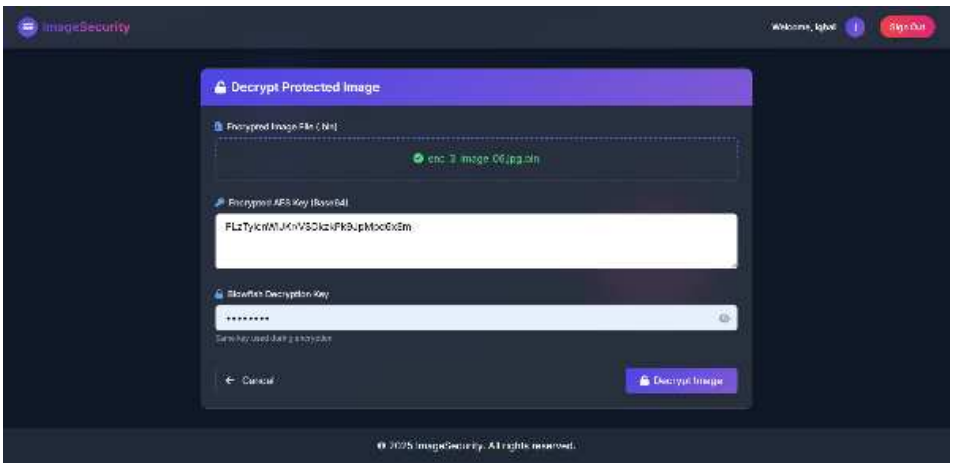

Figure 8: Interface of decryption page

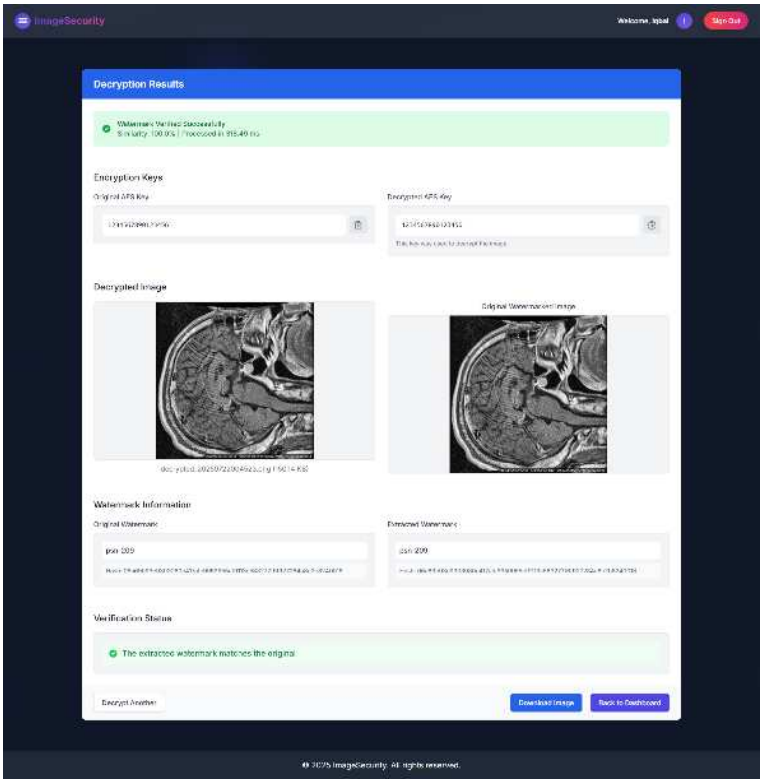

Figure 9: Interface of decryption result page

4.2. Security Testing

Security evaluation was carried out through entropy analysis and penetration testing to validate the robustness of the proposed system against unauthorized access and statistical attacks.

4.2.1 Entropy Analysis

Entropy was used to measure the randomness of the encrypted MRI images. Table 2 summarizes the results of entropy measurement for six encrypted files. The entropy values ranged from 7.91 to 8.00, very close to the theoretical maximum of 8.00, which indicates that the encrypted data had high randomness and showed no exploitable patterns. This demonstrates that the combined use of AES and Blowfish encryption successfully concealed the original image structure and produced ciphertext resistant to frequency-based cryptanalysis.

Table 2: Entropy Analysis

Image File	Watermark	File Size (KB)	Encrypted Size (KB)	Entropy Value	Interpretation
image_01.jpg	psn-201	28.87	56.92	7.91	High randomness
image_02.jpg	psn-202	38.96	76.41	8.00	Near maximum randomness
image_03.jpg	psn-203	48.96	94.73	8.00	Near maximum randomness
image_04.jpg	psn-204	58.62	112.83	8.00	Near maximum randomness
image_05.jpg	psn-205	68.49	132.36	8.00	Near maximum randomness
image_06.jpg	psn-206	78.55	150.16	8.00	Near maximum randomness

Figure 5 shows the entropy comparison between original and encrypted files, confirming that the encryption process increased entropy significantly.

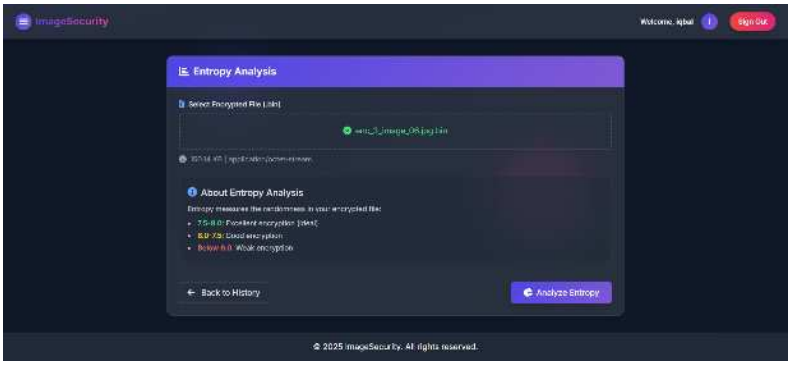

Figure 10: Entropy Analysis (encrypted MRI images)

4.2.2 Penetration Testing

To further evaluate security, penetration testing was performed using OpenSSL by attempting to decrypt encrypted files with both valid and invalid keys. As shown in Table 3, decryption with correct keys consistently restored valid .raw images, while attempts with incorrect keys failed and produced the error message "bad decrypt". This confirms that the system enforces strict key validation and does not leak any information to unauthorized users.

Table 3: Penetration Testing

No	Nama File	Nama Output	Kunci ASCII	Kunci Hexadecimal
1	enc_image_01.bin	dec_image_01.raw	abcdefgh12345678 (kunci benar)	6162636465666768313233343536373 8
2	enc_image_01.bin	dec_image_01.raw	abcd1234abcd1234 (kunci salah)	6162636431323334616263643132333 4
3	enc_image_02.bin	dec_image_02.raw	sandisangatsusah (kunci benar)	73616e646973616e6761747375736168
4	enc_image_02.bin	dec_image_02.raw	susahsandisangat (kunci salah)	7375736168736 16e646973616e676174
5	enc_image_03.bin	dec_image_03.raw	B7e!x2@Y#pLm94Qw (kunci benar)	42376521783240592 3704c6d39345177
6	enc_image_03.bin	dec_image_03.raw	Qw94mLp#Y@2x!e7B (kunci salah)	517739346d4c70235940327821653742
7	enc_image_04.bin	dec_image_04.raw	zT*6R$1mVb8q@XoL (kunci benar)	7a542a365224316d5662387140586f4c
8	enc_image_04.bin	dec_image_04.raw	XoL@q8bVm1$R6*Tz (kunci salah)	586f4c40713862566d312452362a547a
9	enc_image_05.bin	dec_image_05.raw	M9^yAz!o2#Ks7tPx (kunci benar)	4d395e79417a216f32234b7337745078
10	enc_image_05.bin	dec_image_05.raw	Px7tKs#2o!zAy^9M (kunci salah)	507837744b7323326f217a41795e394d
11	enc_image_06.bin	dec_image_06.raw	qL#3rB@5nF8x!ZwV (kunci benar)	714c2333724240356e463878215a7756
12	enc_image_06.bin	dec_image_06.raw	VwZ!x8Fn5@Br3#Lq (kunci salah)	56775a217838466e3540427233234c71

Figure 6 shows an example of a successful decryption using the correct key, while Figure 7 illustrates the failure message obtained when using an invalid key.

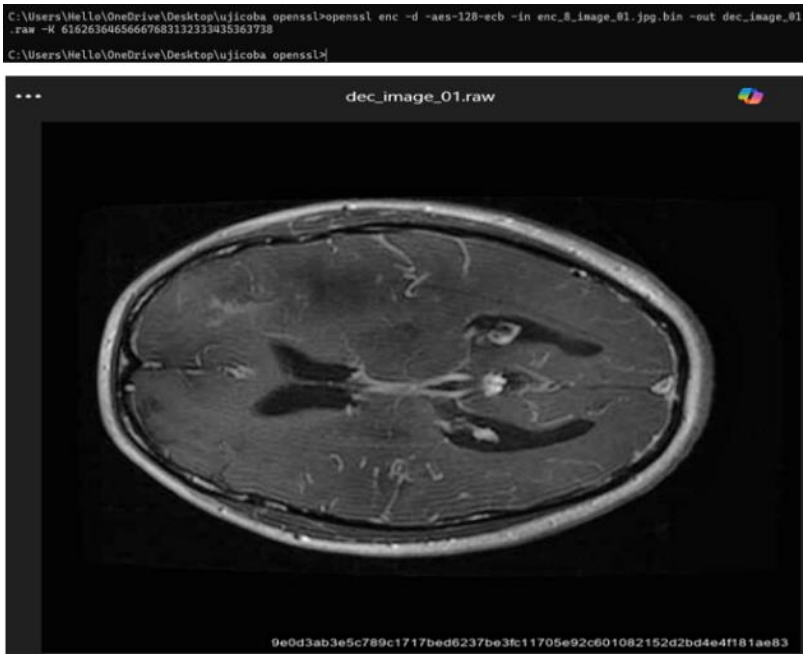

Figure 11: Succesful Decryption (Correct Key)

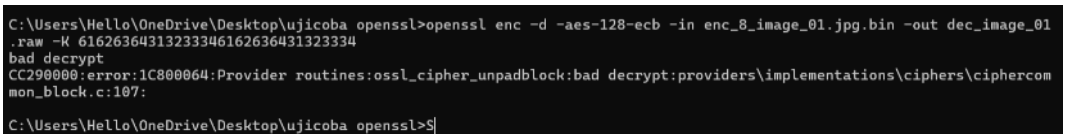

Figure 12: Failed Decryption (Invalid Key)

4.3. Performance Testing

Performance testing was conducted to evaluate how the proposed system performs in terms of speed and storage efficiency. Since medical image applications demand not only high security but also responsiveness and resource efficiency, the evaluation focused on two aspects: the time required for encryption and decryption operations, and the changes in file size at different processing stages. Both factors are essential to determine whether the dual-encryption and watermarking approach can be deployed in real-world scenarios without introducing excessive computational or storage overhead.

4.3.1 Processing Time

Processing time was measured to assess how long the system takes to execute encryption and decryption on medical images of varying sizes. Six MRI images ranging from 28 KB to 78 KB were processed through the complete pipeline, starting with watermark embedding, followed by AES-128 encryption of the watermarked image, Blowfish encryption of the AES key, and finally the decryption stage to restore the watermarked file. The duration of each operation was recorded using Python's time.time() function, providing millisecond precision.

The results, as presented in Table 4, indicate that encryption times ranged between 66.88 ms and 97.44 ms, while decryption was consistently faster, taking between 9.37 ms and 22.51 ms. On average, the system achieved an encryption speed of 82.02 ms and a decryption speed of 13.68 ms. These results demonstrate that the proposed design maintains high efficiency, even when dealing with images that nearly double in size after watermark embedding.

Table 4: Processing Time

Nama Gambar	Ukuran Asli	Watermarked	Encrypted (.bin)	Decrypted	Encrypt Time	Decrypt Time
image_01.jpg	28.87 KB	57.09 KB	57.11 KB	57.09 KB	90.28 ms	12.98 ms
image_02.jpg	38.96 KB	76.35 KB	76.36 KB	76.35 KB	81.96 ms	22.51 ms
image_03.jpg	48.96 KB	94.71 KB	94.72 KB	94.71 KB	85.33 ms	14.46 ms
image_04.jpg	58.62 KB	112.70 KB	112.72 KB	112.70 KB	70.21 ms	9.37 ms
image_05.jpg	68.49 KB	132.51 KB	132.52 KB	132.51 KB	97.44 ms	13.28 ms
image_06.jpg	78.55 KB	150.13 KB	150.14 KB	150.13 KB	66.88 ms	9.52 ms
		Average			82.02 ms	13,68 ms

From these findings, it can be observed that the AES–Blowfish dual-encryption scheme introduces minimal computational overhead. The additional Blowfish encryption of the AES key did not significantly impact the overall time, confirming the system's scalability. Moreover, the decryption process was consistently faster than encryption, which is advantageous for real-world applications where retrieval of medical images is expected to be quick and reliable. Even at its peak, the encryption process completed in less than 100 ms, a threshold that is acceptable for cloud-based health record systems and telemedicine services where near real-time access is crucial.

4.3.2 Storage Efficiency

In addition to speed, storage efficiency was analyzed by observing the file size changes at each processing stage. The images were evaluated in four conditions: the original file, the watermarked image after SVD embedding, the encrypted binary file, and the decrypted result. These measurements are summarized in Table 5.

Table 5: Storage Efficiency

Nama Gambar	Ukuran Asli	Watermarked	Encrypted (.bin)	Decrypted	Encrypt Time	Decrypt Time
image_01.jpg	28.87 KB	57.09 KB	57.11 KB	57.09 KB	90.28 ms	12.98 ms
image_02.jpg	38.96 KB	76.35 KB	76.36 KB	76.35 KB	81.96 ms	22.51 ms
image_03.jpg	48.96 KB	94.71 KB	94.72 KB	94.71 KB	85.33 ms	14.46 ms
image_04.jpg	58.62 KB	112.70 KB	112.72 KB	112.70 KB	70.21 ms	9.37 ms
image_05.jpg	68.49 KB	132.51 KB	132.52 KB	132.51 KB	97.44 ms	13.28 ms
image_06.jpg	78.55 KB	150.13 KB	150.14 KB	150.13 KB	66.88 ms	9.52 ms
		Average			82.02 ms	13,68 ms

The results show that watermark embedding caused the most significant increase in file size, often nearly doubling the original. For instance, image_01 grew from 28.87 KB to 57.09 KB after watermark insertion, while image_06 expanded from 78.55 KB to 150.13 KB. This is expected since the SVD embedding modifies the singular value matrix and introduces additional hash data derived from the watermark. While this increase might be considered substantial in general image compression contexts, it remains acceptable within medical data storage environments, where preserving authenticity and integrity outweighs the demand for minimal storage size.

In contrast, the AES encryption stage introduced only a negligible increase of about 0.01–0.02 KB, which results from block padding to fit the AES encryption standard. Importantly, once decrypted, the images returned exactly to their watermarked sizes without any loss or corruption. This confirms that the encryption–decryption cycle preserves both the fidelity and the integrity of the data.

Taken together, the storage efficiency analysis demonstrates that while watermarking introduces some overhead, the encryption itself remains lightweight in terms of storage consumption. This makes the proposed scheme feasible for large-scale applications, such as cloud-based medical archives or hospital imaging systems, where hundreds or thousands of images may be processed daily. Compared to conventional single-encryption approaches, this system adds robustness and authenticity features without imposing significant storage costs, thereby achieving an effective balance between security and efficiency.

5. Conclussion

Based on the experiments and analysis that have been conducted, this research successfully developed and evaluated a secure system for protecting digital medical images through a combination of double encryption and watermarking. The functional tests confirmed that the system consistently performed all processes as intended, including watermark embedding, AES-128 encryption of the image, Blowfish encryption of the AES key, and accurate decryption to restore the original image and watermark hash. Security evaluations further validated the robustness of the design: entropy analysis showed values ranging from 7.99 to 8.00, indicating that the ciphertexts exhibit strong randomness and resistance to statistical attacks, while penetration testing using incorrect keys consistently failed with "bad decrypt" errors, proving that the system cannot be bypassed without valid credentials.

From a performance perspective, the system demonstrated high efficiency with an average encryption time of approximately 81 ms and a decryption time of about 13 ms, making it suitable even for real-time scenarios. File size analysis revealed that the most significant increase occurred during the watermark embedding stage, which nearly doubled the original size; however, the encryption itself introduced only negligible overhead of about 0.01–0.02 KB. Importantly, the decryption process preserved both image quality and watermark integrity, ensuring that the extracted SHA-256 hash matched the original watermark.

Overall, the proposed system successfully meets the objectives of ensuring confidentiality, authenticity, and efficiency in digital medical image protection. By integrating layered encryption with AES and Blowfish, watermarking through SVD, and integrity verification via SHA-256, the system offers a comprehensive and reliable

solution. Compared to existing approaches that often focus solely on encryption or watermarking, this research provides a balanced method that addresses security, performance, and integrity simultaneously. The results suggest that the proposed approach is not only effective but also practical for implementation in healthcare environments where the protection of sensitive visual data is paramount.

References

And, I., & Expert, D. (2022a). Data Encryption on Video Files Using Android-Based Blowfish Algorithm. *Informatic and Digital Expert* (Vol. 4, Issue 1). https://e-journal.unper.ac.id/index.php/informatics

And, I., & Expert, D. (2022b). Cryptography for Double Encryption on Images Using AES (Advanced Encryption Standard) and RC5 (Rivest Code 5) Algorithms. *Informatic and Digital Expert* (Vol. 4, Issue 1). https://e-journal.unper.ac.id/index.php/informatics

Ariska, N. (2022). Implementation of Hash Function with SHA-256 Algorithm on a Duplicate Image Scanner Application. *Jurnal Sains Dan Teknologi Informasi*, *1*(4), 112–120. https://doi.org/10.47065/jussi.v1i4.2292

Ravida, R., & Santoso, H. A. (2017). Advanced Encryption Standard (AES) 128-Bit for Data Security of Hydroponic Plant Internet of Things (IoT). *Jurnal Rekayasa Sistem dan Teknologi Informasi*, *4*(6), 1157–1164.

Dzikri Azhari Ali, M., & Id Hadiana, A. (2024). Securing Network Log Data Using Advance Encryption Standard Algorithm and Twofish with Common Event Format. *International Journal of Quantitative Research and Modeling*, *5*(3), 341–353.

Enhancing video encryption: AES and blowfish algorithms with random password generation. (2023). *ACCENTS Transactions on Image Processing and Computer Vision*, *9*(25). https://doi.org/10.19101/tipcv.2023.924002

Fahriani, N., & Rosyid, H. (2019). Implementation of Encryption and Decryption Techniques on Video Files Using the Blowfish Algorithm. *6*(6), 697–702. https://doi.org/10.25126/jtiik.201961465

Fajriati Romli, S., Id Hadiana, A., & Rakhmat Umbara, F. (n.d.). Implementation of Advanced Encryption Standard (AES) Cryptography and Spread Spectrum Steganography to Secure Messages in Images. *DES 2023 Journal of Informatics and Communications Technology*, *5*(2), 196–209. https://doi.org/10.52661

Haddad, S. (n.d.). Protection of encrypted and/or compressed medical images by means of watermarking. https://theses.hal.science/tel-03157216v1

Jamaluddin, J., Saragih, N. F., Simamora, R. J., Siringoringo, R., & Purba, E. N. (2021). The Concept of Video Conference Security with AES-GCM Encryption in the Zoom Application. *Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, *4*(2), 109–113. https://doi.org/10.46880/jmika.Vol4No2.pp109-113

Malvi, A. (n.d.). Securing Image Files in Video Media with RSA Algorithm Cryptography and End of File (EOF) Algorithm Steganography.

Melina, M., Hadiana, A. I., Putra, E. K., Sukono, S., Napitupulu, H., Murniati, A., & Kusumangtyas, V. A. (2024). Digital signature authentication using Rivest-Shamir-Adleman cryptographic algorithm. *AIP Conference Proceedings*, *2867*(1), 020011. https://doi.org/10.1063/5.0225078

Monica, T., Hadiana, A. I., & Melina, M. (2024). Question Bank Security Using Rivest Shamir Adleman Algorithm and Advanced Encryption Standard. *JIKO (Jurnal Informatika Dan Komputer)*, *7*(3), 175–181. https://doi.org/10.33387/jiko.v7i3.8654

Nagamunthala, M., & Manjula, R. (2023). Implementation of a Hybrid Triple-Data Encryption Standard and Blowfish Algorithms for Enhancing Image Security in Cloud Environment. *Journal of Computer and Communications*, *11*(10), 135–149. https://doi.org/10.4236/jcc.2023.1110009

Rahayu, A., Abdillah, G., & Ashaury, H. (2024). Security Using AES (Advanced Encryption Standard) and Bcrypt (Blowfish Crypt) Algorithms on Document Files. In Journal of Informatics Engineering Students. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 8, Issue 6).

Solikhudin, A., Mohammad Yasin, dan, & Matematika Fakultas Matematika Dan, J. (n.d.). Information Hiding in Digital Images with Singular Value Decomposition (SVD).

Taj, R., Tao, F., Kanwal, S., Almogren, A., & Rehman, A. U. (2024). A SURF and SVD-based robust zero-watermarking for medical image integrity. *PLoS ONE*, *19*(9 September). https://doi.org/10.1371/journal.pone.0307619

Wayan Angga Wijaya Kusuma, I., Afriliana Kusumadewi, dan, Teknik, F., Studi Teknik Elektro, P., Widya Dharma, U., Ki Hajar Dewantara, J., Karanganom, D., Klaten Utara, K., & Klaten, K. (2021). Comparative Analysis of Segmentation Results Using K-Means and Fuzzy C-Means Methods on Compressed Input Images. (Vol. 13, Issue 2).

Zermi, N., Khaldi, A., Kafi, M. R., Kahlessenane, F., & Euschi, S. (2021). Robust SVD-based schemes for medical image watermarking. *Microprocessors and Microsystems*, *84*. https://doi.org/10.1016/j.micpro.2021.104134