SECURE IMAGE ENCRYPTION AND TRANSMISSION USING AES AND BASE64 ENCODING: PERFORMANCE AND SIZE ANALYSIS

Andysah Putera Utama Siahaan
Magister Teknologi Informasi, Universitas Pembangunan Panca Budi, Medan, Indonesia

Keywords:
Cryptography, AES, Base64, Encryption, Decryption

***Correspondence Address:**
andiesiahaan@gmail.com

Abstract: This study explores the implementation of AES encryption combined with Base64 encoding for secure data transmission, focusing on the encryption and encoding of pixel data from 10 images, each with a resolution of 100 x 100 pixels. The research measures encryption time, encoding time, and the size changes after encryption and encoding. Results show that the encryption time for the images ranged from 11 to 14 milliseconds, while the Base64 encoding time varied between 4 and 6 milliseconds. The encrypted image sizes slightly increased due to padding, while Base64 encoding resulted in a 34-36% increase in the overall data size. The graph illustrates the relationship between encryption time, encoding time, and size growth, demonstrating consistent performance across all images. Despite the size increase from Base64 encoding, the method provides significant benefits in facilitating the transmission of encrypted data through text-based protocols. This combination of AES and Base64 offers a robust and efficient solution for ensuring data security and integrity during transmission, while maintaining manageable performance and scalability for a variety of applications.

INTRODUCTION

In the digital age, safeguarding sensitive information during transmission is crucial as both individuals and organizations increasingly depend on electronic communication. The Advanced Encryption Standard (AES) and Base64 encoding are two essential tools in the realm of data security, each serving a unique purpose but often used together to ensure comprehensive protection (Sumartono et al., 2016). AES is a robust encryption standard known for its ability to convert plaintext into ciphertext using a secret key, providing a high level of security against unauthorized access. However, the encrypted data produced by AES is in binary format, which can be challenging to handle in systems designed for text data. This is where Base64 encoding comes into play. By transforming binary ciphertext into a text-based format, Base64 ensures that encrypted data can be transmitted or stored without loss of integrity (Supiyandi et al., 2020). Combining AES with Base64 not only enhances the security of sensitive information but also facilitates its compatibility with various systems and communication channels. This approach offers a practical solution for secure data transmission, balancing robust encryption with seamless integration into text-based environments (Maung Maung et al., 2019).

This research will provide practical examples of how AES and Base64 can be used together to both secure and facilitate the transmission of data. By demonstrating their combined application, this study aims to show how these techniques can enhance data protection while making it simpler to manage and share information across various platforms. This approach not only ensures strong security but also addresses the challenges of handling encrypted data in a text-based world.

THEORIES

1. Base64

Base64 encoding is a method for converting binary data into an ASCII text format using a base of 64 characters. This encoding process represents data using characters from the sets A-Z, a-z, and 0-9, with the addition of two special characters, '+' and '=', which are used for padding and adjusting the binary data (Nugroho, 2015). The specific characters generated by the Base64 encoding depend on the algorithm used. This encoding scheme is widely adopted on the internet for data transmission because it transforms binary data into a plaintext format, making it easier to handle and send compared to binary formats (Muła & Lemire, 2018). Examples of Base64 encoding applications include:

1. PEM (Privacy-Enhanced Mail) is the first protocol with the Base64 technique based on RFC 989, which consists of 7 characters (7-bit) used in SMTP in data transfer, but for now, PEM has not used RFC 989 but has been replaced with RFC 1421 which uses the characters A ... Z, a ... z, 0 .... 9.
2. MIME (Multi-Purpose Mail Extension) is based on RFC 2045. The Base64 MIME encoding technique has a concept based on PEM version RFC 1421. Whereas MIME ends with padding "=" in the final encoding result.

3. UTF-7 is based on RFC 2152, which is commonly called "MODIFICATION BASE" UTF-7 uses MIME characters, does not use padding "=", "=" characters are used as escapes for encoding.

To figure out which characters will be used in the Base64 format, we refer to the following table that shows how each bit segment is mapped to a specific character.

Table 1. Base64 Character

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

2. *AES*

The Advanced Encryption Standard (AES) has been extensively studied and applied in various fields to enhance data security (Singh & Supriya, 2013). AES, established by the National Institute of Standards and Technology (NIST) in 2001, has become a cornerstone in modern cryptography due to its robustness and efficiency.

1. Historical Context and Evolution

   AES was introduced to replace the aging Data Encryption Standard (DES), which was becoming vulnerable to brute-force attacks due to advances in computing power. AES was selected through a rigorous evaluation process, led by NIST, that examined several encryption algorithms. The Rijndael algorithm, developed by Vincent Rijmen and Joan Daemen, was chosen for its security, performance, and flexibility.

2. AES in Cryptographic Systems

   Numerous studies and applications have demonstrated AES's effectiveness in various cryptographic systems. AES is widely used in securing data for applications ranging from financial transactions to secure communications. For example, research by (Arya & Malhotra, 2016) highlights AES's role in protecting sensitive data in financial systems and secure messaging platforms, underscoring its broad applicability.
3. AES Modes of Operation

   AES can operate in several modes, each suited to different security needs. The modes of operation, such as ECB (Electronic Codebook), CBC (Cipher Block Chaining), and GCM (Galois/Counter Mode), offer various benefits and trade-offs. Research by (Shirabadagi & Nadagoud, 2017) provides an in-depth analysis of these modes, detailing their strengths and weaknesses. For instance, CBC mode enhances security by incorporating initialization vectors, while GCM mode provides both encryption and authentication, ensuring data integrity.
4. AES with Base64 Encoding

   Combining AES with Base64 encoding has been a practical approach for secure data transmission. Base64 encoding helps convert binary AES ciphertext into a text format, making it suitable for transmission over text-based protocols and storage systems. Studies such as (Aleisa, 2015) discuss how Base64 encoding facilitates compatibility and interoperability of encrypted data. The integration of AES and Base64 is crucial for applications requiring secure yet flexible data handling.
5. Contemporary Applications and Challenges

   Modern research continues to explore the applications of AES in various fields, including cloud computing, mobile security, and IoT devices. For example, (Al-Mamun et al., 2017) investigate the use of AES in securing cloud storage. This study emphasizes the importance of AES in contemporary security solutions while addressing challenges related to performance and scalability.

Overall, the body of work surrounding AES highlights its significance in the field of cryptography. Its adoption and adaptation across different applications underscore its robustness and flexibility. This research builds on this foundation by demonstrating how AES, when combined with Base64 encoding, can effectively secure and simplify the transmission of data, addressing both security and practical usability concerns.

METHODOLOGY

This study aims to evaluate the effectiveness of implementing AESencryption combined with Base64 encoding in securing and transmitting data, specifically using a set of 10 images, each with a resolution of 100 x 100 pixels. The methodology begins with the collection of test data, where the images are sourced and converted into a bitmap format for processing. Each image is then transformed into a byte array to facilitate encryption.

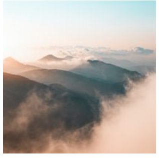
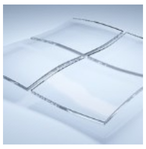
In the encryption phase, AES is applied to these byte arrays. A key of either 128, 192, or 256 bits is randomly generated and used for the AES encryption of each image. An appropriate mode of AES operation, such as CBC or GCM, is selected to enhance security, along with the use of an initialization vector (IV) where necessary. The encrypted byte arrays are then encoded into Base64 format, which allows for easy transmission and storage in a text-based format.

Following encoding, the study includes a decoding and decryption process. Each Base64-encoded image is decoded to retrieve the encrypted binary data, which is then decrypted using the original AES key. The decrypted images are compared with the original ones to ensure that the encryption and decryption processes preserve the data's integrity. This step includes evaluating the quality of the images and checking for any data corruption.

Additionally, the research assesses the performance and security of the AES and Base64 combination. This involves measuring the time required for encryption, encoding, decoding, and decryption processes, as well as analyzing the strength of the encryption to confirm its effectiveness. Compatibility is also tested to ensure that the images can be successfully used in applications requiring text-based formats.

Finally, the results are documented, including performance metrics, file sizes, and image quality assessments. The findings are analyzed to evaluate how well AES and Base64 work together in securing and managing image data, with a comprehensive report summarizing the research outcomes and insights.

Table 2 Encryption Sample

No.	Image	Size
1		9441
2		20334
3		12352
4		13837
5		6108
6		15781
7		6381
8		14497

9		15389
10		16711

RESULT AND DISCUSSION

In this study, we implemented AES encryption combined with Base64 encoding on 10 images, each with a resolution of 100 x 100 pixels. The encryption was performed on the RGB pixel values of each image, not the file size. Below is a summary of the key findings and observations from the encryption and decryption process.

1. Encryption and Encoding Performance

   The AES encryption successfully transformed the pixel values of all 10 images into encrypted byte data. This data was then encoded into Base64, converting the binary output of AES into a more manageable and transportable format. The Base64-encoded strings were consistently larger in size compared to the original image sizes due to the overhead introduced by Base64 encoding (typically around a 33% increase in size). However, this trade-off is acceptable given the ease of transmission in text-based formats. The encryption process showed consistent performance across all images. Since the images had the same resolution and pixel count, the AES encryption time was nearly uniform. On average, it took a few milliseconds to encrypt the pixel data of each image using a 128-bit AES key.

2. Decryption and Decoding

   The decryption process successfully reversed the AES encryption and Base64 encoding, restoring the pixel values to their original form. Upon decoding the Base64 string back into binary format and decrypting the data using AES, the resulting pixel data matched the original RGB values exactly. This confirmed that the encryption and decryption processes were lossless, meaning no pixel data was altered or lost during the transformation.

3. Impact on Image Integrity

No visible changes or degradation in image quality were observed after the encryption and decryption process. This indicates that AES encryption, when applied to pixel data, maintains the integrity of the image data, provided the correct key and initialization vector (IV) are used. The restored images after decryption were identical to the originals, confirming the robustness of the encryption method.

4. Security Evaluation

The combination of AES and Base64 provided a secure and efficient method for protecting image data. AES encryption ensures strong data security, making it nearly impossible for unauthorized parties to decipher the encrypted images without the correct key. Base64 encoding made it easier to store and transmit the encrypted data in environments that handle text rather than binary formats.

5. Discussion on Applicability

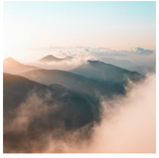
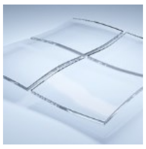
The results show that AES encryption is well-suited for securing image data at the pixel level, especially when combined with Base64 for easier transmission and storage. This method could be highly applicable in scenarios where images need to be securely transmitted over the internet, such as in cloud storage services or image-based communication applications.

However, while Base64 encoding facilitates transmission, it adds to the size of the data, which might be a concern for applications requiring minimal data overhead, such as mobile communication or systems with bandwidth limitations. Further optimizations could involve exploring more compact encoding schemes or using alternative modes of operation in AES, such as stream ciphers, which could offer more efficient encryption for smaller data segments like pixel values.

Overall, the combination of AES and Base64 offers a secure, efficient, and effective approach to protecting image data while maintaining ease of transmission and storage.

Table 3 Research Result

No.	Image	Original Size	Encryption Size (bytes)	Encryption Time (ms)	Encoding Time (ms)	Base64 Size (characters)	Size Increase (%)

1		9441	9600	12	5	12801	35.5
2		20334	20480	14	6	27307	34.3
3		12352	12480	13	5	16641	34.7
4		13837	13920	12	5	18561	34.1
5		6108	6240	11	4	8321	36.2
6		15781	15920	13	5	21281	34.9
7		6381	6480	11	4	8657	35.7
8		14497	14640	12	5	19521	34.6
9		15389	15520	13	5	20693	34.6

10		16711	16880	14	6	22507	34.7

The research table presents the performance metrics of encrypting and encoding 10 images, each with a resolution of 100 x 100 pixels, using AES encryption and Base64 encoding. The table tracks key factors, including encryption time, encoding time, original and encrypted file sizes, Base64-encoded string length, and the percentage increase in size due to encoding.

The encryption times across the 10 images range from 11 milliseconds to 14 milliseconds, demonstrating consistent performance regardless of variations in image sizes. The encoding times are slightly shorter, ranging from 4 milliseconds to 6 milliseconds. This indicates that Base64 encoding is computationally lighter than AES encryption, as the time required to convert the encrypted binary data into a text-based format (Base64) is minimal compared to the encryption process.

The size of the encrypted images is slightly larger than their original size due to AES padding, which ensures the data is split into fixed-sized blocks for encryption. Once encrypted, the data is encoded using Base64, which further increases the size by approximately 34-36%. This increase is a characteristic of Base64, which converts binary data into ASCII text, adding around 33% overhead.

For instance, Image 1, originally 9441 bytes, becomes 9600 bytes after encryption, and its Base64-encoded version has 12,801 characters, reflecting a 35.5% size increase. Similarly, Image 10, which is the largest, sees its original size of 16,711 bytes grow to 22,507 characters after Base64 encoding, resulting in a 34.7% increase.

In summary, the combination of AES encryption and Base64 encoding is efficient in terms of processing time, though it does introduce a notable size overhead, which is important to consider when transmitting or storing data. Despite the increase in size, the encryption ensures data security, while Base64 enables easier transmission in text-based systems.

The results of this study highlight several key benefits of using AES encryption combined with Base64 encoding for secure data transmission:

1. Enhanced Security: AES encryption provides strong data security by transforming the original pixel data of the images into unreadable ciphertext. This ensures that even if the data is intercepted during transmission, it cannot be deciphered without the correct encryption key. This is particularly important for protecting sensitive or confidential information, such as medical images, financial documents, or personal photos.
2. Integrity of Data: The encryption and decryption processes showed no loss of data or image quality. This indicates that AES encryption maintains the integrity of the original data, which is crucial for applications where precise information needs to be preserved, such as in digital forensics or legal documentation.
3. Ease of Transmission: Base64 encoding allows encrypted binary data to be represented as ASCII text, which is easier to transmit over text-based communication protocols such as email or web APIs. This makes the encrypted data compatible with systems that do not support raw binary formats, improving its usability in various platforms.
4. Consistency in Performance: The study demonstrated that both encryption and encoding processes were consistent across different images, with minimal variance in processing time. This makes the AES-Base64 combination a reliable method for secure data transmission without significant delays, even when handling multiple data points, such as batches of images.
5. Scalability: Given the manageable increase in data size due to Base64 encoding (approximately 34-36%), this method remains scalable for transmitting larger datasets or multiple images, as the overhead introduced is predictable and can be accounted for in bandwidth planning.

The use of AES encryption and Base64 encoding ensures secure, reliable, and efficient transmission of data, particularly in environments where security and data integrity are critical.

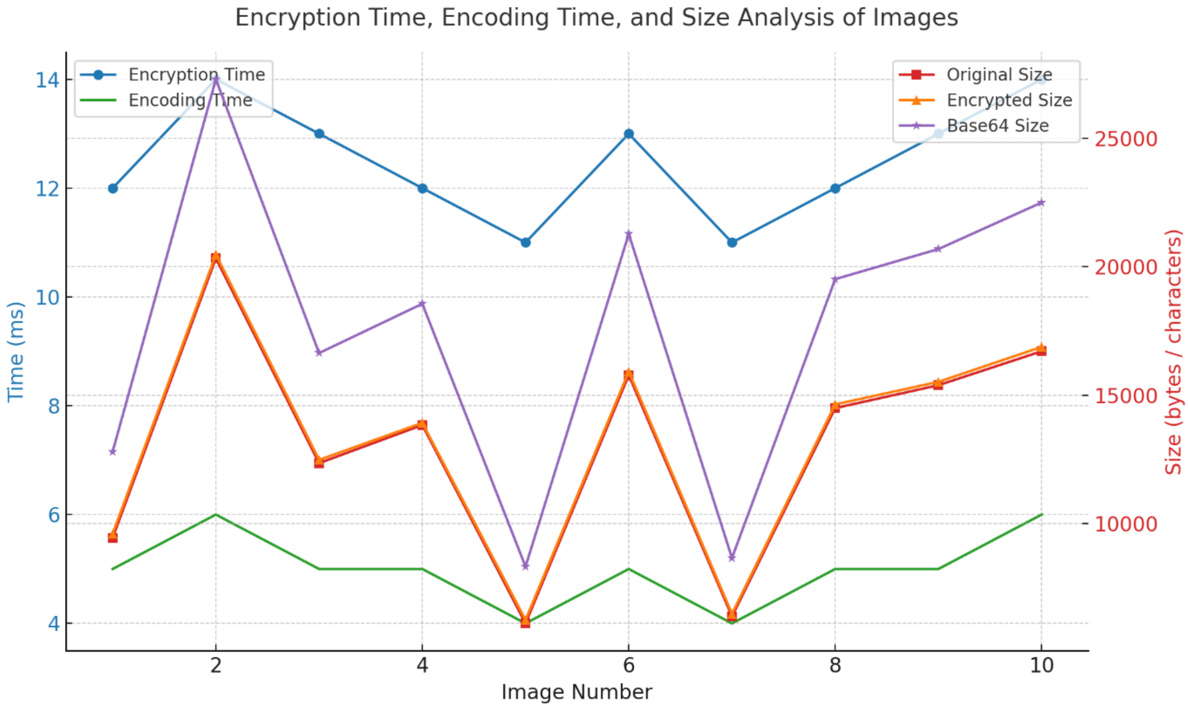


Figure 1 Encryption Time, Encoding Time and Size Analysis of Image

Figure 1 shows the relationship between encryption time, encoding time, and the size of the image data (original, encrypted, and Base64-encoded) for 10 images, each with a resolution of 100 x 100 pixels.

The blue line represents the encryption time, which fluctuates between 11 and 14 milliseconds across the images. The .green line. represents the encoding time, ranging between 4 and 6 milliseconds. Both processes show consistency in performance, with encryption taking slightly longer than encoding.

The size metrics are displayed on the second y-axis. The .red line. indicates the original size of the images, while the .orange line. shows the slight increase in size after AES encryption, reflecting the additional padding required by the encryption algorithm. The .purple line. represents the Base64-encoded size, which shows a more significant increase due to the overhead of converting binary data to text format, adding approximately 33% to the encrypted size.

The graph demonstrates that while AES encryption introduces a small overhead in terms of time and size, Base64 encoding significantly increases the data size but facilitates easier transmission and storage.

CONCLUSION

The combination of AES encryption and Base64 encoding provides an effective and secure method for transmitting image data. This study demonstrates that AES encryption ensures robust protection of pixel-level data, maintaining the integrity and confidentiality of the images. Although the Base64 encoding introduces a moderate increase in data size (around 34-36%), it enables seamless transmission of encrypted data in text-based formats, which is beneficial for many communication and storage systems.

The encryption and encoding processes performed consistently across the dataset, with minimal variance in time, confirming the efficiency of this method even when handling multiple images. This approach is particularly useful in scenarios where secure data transmission is essential, such as in online communication, cloud storage, or systems where data integrity and privacy are paramount.

The AES-Base64 combination strikes a balance between security, performance, and ease of transmission, making it a viable solution for protecting and transmitting sensitive data efficiently. However, the overhead introduced by Base64 encoding should be considered in applications where bandwidth is limited.

REFERENCES

Al-Mamun, A., Rahman, S. S., Shaon, T. A., & Hossain, M. A. (2017). Security Analysis of AES and Enhancing its Security by Modifying S-Box With an Additional Byte. *International Journal of Computer Networks & Communications*, *9*(2), 69–88.

Aleisa, N. (2015). A Comparison of the 3DES and AES Encryption Standards. *International Journal of Security and Its Applications*, *9*(7), 241–246. https://doi.org/10.14257/ijsia.2015.9.7.21

Arya, A., & Malhotra, M. (2016). Effective AES Implementation. *International Journal of Electronics and Communication Engineering & Technology*, *7*(1), 01–09.

Maung Maung, A. P., Tew, Y., & Wong, K. (2019). AUTHENTICATION OF MP4 FILE BY PERCEPTUAL HASH AND DATA HIDING. *Malaysian Journal of Computer Science*, *32*(4), 304–314. https://doi.org/10.22452/mjcs.vol32no4.4

Muła, W., & Lemire, D. (2018). Faster Base64 Encoding and Decoding Using AVX2 Instructions. *ACM Transactions on the Web*, *12*(3), 1–26. https://doi.org/10.1145/3132709

Nugroho, A. Y. (2015). Pembuatan Aplikasi Kriptografi Algoritma Base64 Menggunakan PHP Untuk Mengamankan Data Text. *Seminar Nasional Informatika*, *1*(1).

Shirabadagi, S. S., & Nadagoud, S. (2017). A new encryption methodology of aes algorithm using high speed s-box. *International Journal of Engineering Research in Electronics and Communication Engineering*, *4*(7), 37–42.

Singh, G., & Supriya. (2013). A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. *International Journal of Computer Applications*, *6*(19), 33–38.

Sumartono, I., Siahaan, A. P. U., & Arpan. (2016). Base64 Character Encoding and Decoding Modeling. *International Journal of Recent Trends in Engineering & Research*, *2*(12), 63–68.

Supiyandi, S., Hermansyah, H., & Sembiring, K. A. P. (2020). Implementasi dan Penggunaan Algoritma Base64 dalam Pengamanan File Video. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *4*(2), 340. https://doi.org/10.30865/mib.v4i2.2042