Journal of Computer Engineering, Electronics and Information Technology (COELITE)

Journal homepage: https://ejournal.upi.edu/index.php/COELITE

Digital Data Security Using a Combination of Base64 Encoding, Rail Fence Cipher, and GZIP Compression

Deden Pradeka[1], Zahra Khaerunnisa[2], Syifa Aqila Humaira[3], Aidha Salsa Billa[4], Murnawan[5], Budiman[6]*

[1,2,3,4]Department of Computer Engineering, Universitas Pendidikan Indonesia, Indonesia
[5]Department of Information Systems, Universitas Widyatama, Indonesia
[6]Department of Information System, Universitas Informatika dan Bisnis Indonesia, Indonesia
Correspondence E-mail: dedenpradeka@upi.edu

ABSTRACT

Digital data security is essential in protecting sensitive information in the current information era. This research implements a hybrid security approach by combining Base64 encoding, the Rail Fence cipher, and GZIP compression. Base64 encoding converts digital files into a standardized text format for easier processing, while the Rail Fence cipher applies transposition to further scramble the encoded data, enhancing confidentiality. GZIP compression reduces file size while adding a layer of complexity, making unauthorized data recovery more difficult. The experiment involved ten trials using files with extensions such as XLSX, DOCX, JPG, PDF, and PPTX. Pearson correlation analysis revealed coefficients close to 0.00 or slightly negative, indicating no significant correlation between plaintext and ciphertext, which signifies strong data randomness and security. Additionally, GZIP compression achieved an average file size reduction of approximately 1 KB, although some smaller files showed minimal compression effects. Overall, the combined method effectively enhances both data security and storage efficiency.

ARTICLE INFO

Article History:
Submitted/Received 22 Mar 2025
First Revised 29 Mar 2025
Accepted 30 Mar 2025
First Available online 1 Apr 2025
Publication Date 1 Apr 2025

Keyword:
Data Security, Base64 Encoding, Rail Fence Cipher, GZIP Compression, Cryptography

1. INTRODUCTION

The development of information technology has had a significant impact on the increase in the flow of digital data transmitted over the internet. This intensive data exchange presents its challenges in maintaining the security and privacy of sensitive information from various threats such as theft, manipulation, or data breaches [1]. This issue is crucial given the increasing number of data security breaches, which result in both material and non-material losses for individuals and institutions.

Various data protection techniques have been developed to safeguard digital information, including encryption, encoding, and data compression methods. However, relying on a single technique alone is sometimes insufficient to ensure optimal security. An integrative solution that combines multiple methods is needed to strengthen security layers against digital data threats. For instance, Base64 encoding has a weakness due to the availability of many online generators [2], making it necessary to further scramble the encoded data using the Rail Fence Cipher method [3]. From another perspective, handling large files can be challenging for transmission or storage, making GZIP compression a suitable alternative [4].

This study proposes a solution that combines three methods: Base64 Encoding, Rail Fence Cipher, and GZIP Compression. Base64 Encoding is used to convert files into a standardized text format, making them more flexible for encryption processing. The Rail Fence Cipher provides additional security through a simple yet effective transposition technique to obfuscate the original message content. Finally, GZIP compression reduces data size while also making unauthorized reading more difficult due to the compression process. By integrating these three methods, this approach is expected to enhance data security and improve efficiency in digital data transmission.

2. METHODS

This research employs the Design and Development (D&D) research method. This method consists of several key stages: problem identification, solution design, solution development, and final product evaluation [5][6], You can see **Figure 1**.

In the initial stage, an analysis is conducted to identify the needs for digital data security and assess the weaknesses of existing methods. Subsequently, the design of the combination of Base64 Encoding, Rail Fence Cipher, and GZIP Compression is systematically structured. The development stage involves implementing this design into an application or prototype that can be tested directly.

Finally, the evaluation process is carried out through a series of tests that assess security effectiveness, file size efficiency, and data processing time. The evaluation results will be used to measure the success of the proposed solution in enhancing digital data security and provide recommendations for future improvements.

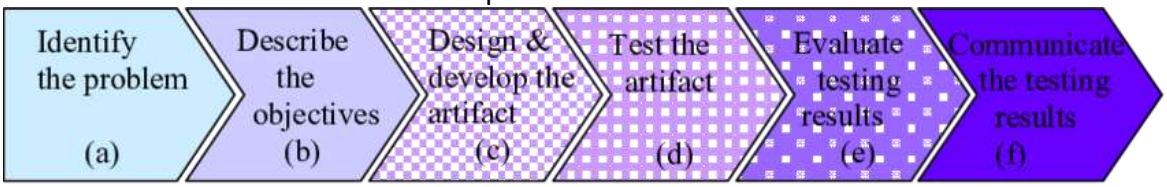

Figure 1. Design and Development (D&D) [7]

2.1. Data Security

Data security is a set of measures, practices, and technologies used to protect digital information from threats such as theft, damage, or unauthorized access [8]. This includes the

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

use of encryption to maintain confidentiality, user authentication to ensure that only authorized individuals can access information, and the implementation of firewalls and antivirus software to protect against cyber threats such as malware or ransomware.

The primary goal of data security is to preserve data integrity, confidentiality, and availability, thereby reducing the risks of information leaks, misuse of personal data, and protecting user privacy. Due to the rapid advancement of information technology and increasingly complex threats, the implementation of strict data security policies and educating users on safe data management practices has become crucial.

2.2. Cryptography

Cryptography is the science and technique of securing information by transforming original data (plaintext) into an unreadable format (ciphertext) through the encryption process, ensuring that only authorized parties with the secret key can revert it to its original form via decryption [9][10].

This technique is used to protect confidentiality, integrity, and authentication, and ensure that data is not accessed or altered by unauthorized entities during transmission or storage, can you see in **Figure 2**. Modern cryptography utilizes complex algorithms, such as symmetric encryption (which uses a single key for both encryption and decryption) and asymmetric encryption (which employs a pair of public and private keys) to provide a high level of security against cyber threats.

With the advancement of digital technology, cryptography has become a vital component of data security in various aspects of life, including financial transactions, electronic communications, and user privacy protection in the digital world.

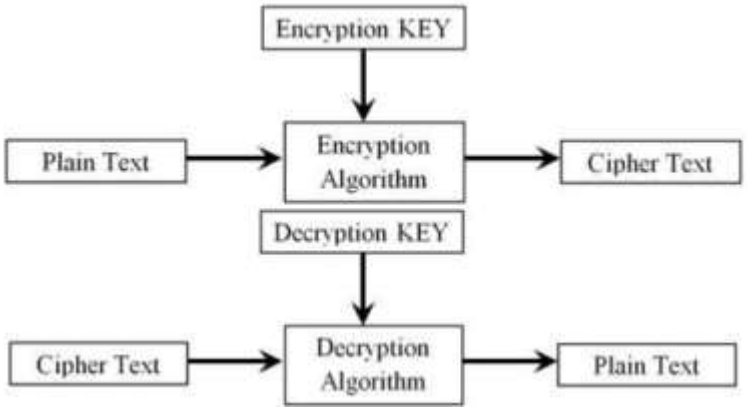

Figure 2. Encryption and Decryption Mechanism

2.3. Data Compression

Data compression is a technique or method used to reduce the size of files or digital information, making storage more efficient and data transmission faster [11]. This process works by eliminating redundancy or unnecessary information, so the remaining data becomes more compact without significantly reducing quality.

There are two main types of data compression:

1. Lossless compression, which allows the original data to be fully restored, as seen in formats like ZIP or PNG.

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

2. Lossy compression, which sacrifices some non-essential information to achieve a higher compression rate, is used in formats like JPEG or MP3.

Data compression is essential in various digital applications, particularly for image, audio, and video storage, as well as internet data transmission, to enhance efficiency, speed, and reduce bandwidth usage.

2.4. Base64

Base64 is an encoding scheme used to convert binary data (such as image, audio, or document files) into a text format consisting of 64 ASCII characters, making it easier to transmit or store through text-based protocols such as email or HTTP [12], as shown in **Figure 3**. Base64 works by dividing data into small blocks (each 3 bytes or 24 bits) and then converting each block into 4 ASCII characters by referring to the Base64 character table [13].

The Base64 encoding steps are as follows:

1. Take the binary data and divide it into 3-byte (24-bit) blocks.
2. Convert each 3-byte block into 4 groups of 6 bits (since 24 bits = 4 groups × 6 bits).
3. Convert each 6-bit group into a corresponding character from the Base64 table, which consists of uppercase A-Z, lowercase a-z, digits 0-9, and the symbols + and /.
4. If the number of bytes is not a multiple of 3, padding characters ('=') are added at the end to complete the block.

The reverse process is called Base64 decoding, which converts Base64 text back into its original binary form by following the opposite steps. Base64 is widely used in internet data transmission, particularly for embedding images in HTML or sending email attachments. However, in this study, the author identifies a weakness in Base64 encoding—the availability of numerous online encoders and decoders, which creates a security vulnerability when someone obtains the encoded data.

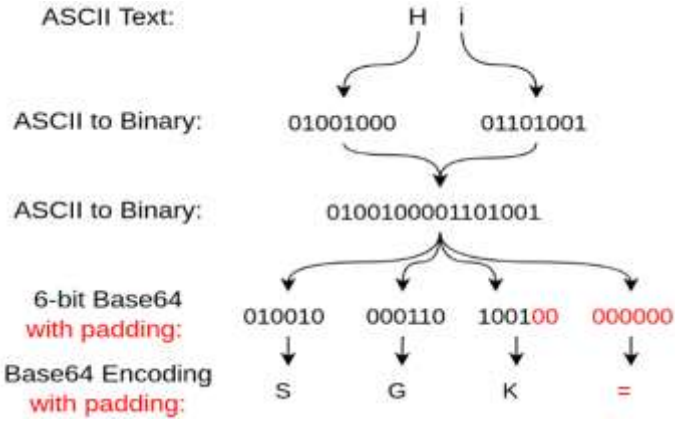

Figure 3. Base64 process [14]

2.5. Rail Fence Cipher

The Rail Fence Cipher is a classical cryptographic method categorized as a transposition cipher, which encrypts a message by altering the position or arrangement of its characters rather than replacing them [15]. This method works by writing the text in a zig-zag pattern across a specified number of rows and then reading it horizontally to produce the ciphertext, in **Figure 4**. Rail Fence Cipher Steps:

a. Encryption:

1. Determine the number of rows (rails) to be used, for example, 3 rails.

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

2. Write the message diagonally in a zig-zag pattern, moving downwards and then upwards, repeating until all characters are placed.
3. After completing the zig-zag pattern, read the text horizontally, row by row, from top to bottom to obtain the ciphertext.
4. Example encryption with the message "KEAMANANDATA" and 3 rails, and then the ciphertext obtained: KMNTEAAADAAA:

```
Rail 1: K . . . M . . . N . . . T .
Rail 2: . E . A . A . A . D . A . A
Rail 3: . . A . . . N . . . A . . .
```

Figure 4. Encryption of a Message through the Rail Fence Cipher Technique

b. Decryption:

1. Determine the same number of rows (rails) used during encryption.
2. Mark the zig-zag positions where the ciphertext characters will be placed.
3. Fill the ciphertext into the marked positions horizontally.
4. Read the message diagonally in a zig-zag pattern to reconstruct the original plaintext.

With this technique, Base64-encoded information is further secured by scrambling the character order. However, this method is relatively simple and not secure enough for critical communications without additional encryption techniques.

2.6 Gzip (GNU zip)

Gzip (GNU zip) is a digital data compression method that uses the lossless DEFLATE algorithm to reduce file size, making data more efficient for storage and transmission [16]. This format is widely used on web servers to accelerate data transfer over the HTTP protocol. Steps in the Gzip Compression Process:

a. Compression (Encoding) Steps:

1. Identify Input Data:
    a) The original data or file (text, images, or other files) to be compressed.
2. Apply the DEFLATE Algorithm:
    b) Analyze the data to detect redundancy (repeated patterns).
    c) Use the LZ77 method (pattern matching) and Huffman coding to significantly reduce file size.
3. Store in Gzip Format:
    a) The compressed data is saved with an added header and checksum for validation during extraction.
4. Generate the Output File:
    1. The resulting file has the extension .gz.

b. Decompression (Decoding) Steps:

1. Retrieve the .gz file.
2. Extract header information to validate data integrity.
3. Use the DEFLATE algorithm to restore the data to its original form without loss of quality (lossless).
4. Recover the original file exactly as it was before compression.

2.7 Pearson Correlation

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

The Pearson correlation, or Pearson Correlation Coefficient, is a statistical measure used to assess the strength and direction of a linear relationship between two numerical variables [17]. However, in cryptography, the best value is one where there is no correlation between the plaintext and the ciphertext. This is because if there is no correlation, it can be said that there is a random transformation from plaintext to ciphertext, which implies enhanced security. Equation 1 presents the Pearson correlation formula, which can be seen in equation 1. Pearson Correlation Formula:

$$r = \frac{\sum (x_i - \underline{x})(y_i - \underline{y})}{\sqrt{\sum (x_i - \underline{x})^2 . \sum (y_i - \underline{y})^2}} \quad (1)$$

Wherex_i and y_i: $are\ the\ i-th\ \ data\ points\ of\ variable\ X\ and\ Y$.
Then $\underline{x}$ and $\underline{y}$: $represent\ the\ mean\ of\ X\ and\ Y$.
Next, r: $is\ the\ koefisien\ korelasi\ pearson\ correlation\ coefficient$.
From the Pearson correlation value, the interpretation is as follows:
if ***r*** = 1, there is a perfect positive relationship (as X increases, Y also increases linearly).
if ***r*** = -1, there is a perfect negative relationship (as X increases, Y decreases linearly). And
if ***r*** = 0, here is no linear relationship between X and Y.

Table 1 provides a comprehensive interpretation of Pearson correlation coefficients, outlining the commonly accepted thresholds used to classify the strength and direction of relationships between variables. This framework allows researchers to evaluate the degree of linear association observed in the data, facilitating more informed statistical inferences and discussions within the context of empirical analysis.

Table 1. General Interpretation of Pearson Correlation Values

Value of *r*	Interpretation
0.90 – 1.00	Very strong (positive) correlation
0.70 – 0.89	Strong (positive) correlation
0.50 – 0.69	Moderate (positive) correlation
0.30 – 0.49	Weak (positive) correlation
0.00 – 0.29	Very weak or no correlation
Negative values	Same interpretation, but in the negative direction

In addition to statistical evaluation, recent advancements in cryptographic methods have demonstrated the effectiveness of combining multiple security layers. For example, Zala and Khan [18] proposed an integrated approach using Base64 encoding and Fernet encryption to protect textual data more robustly. Similarly, Speidel and Manoharan [19] explored individualized puzzle-based learning to enhance conceptual understanding of security mechanisms. From a cybersecurity infrastructure standpoint, Finck and Dohrmann [20] revealed that even more than a decade after the Stuxnet incident, critical systems such as

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

Siemens S7 remain vulnerable, emphasizing the urgent need for layered and adaptive data protection strategies.

3. RESULTS AND DISCUSSION

a. Pearson Correlation Results

Table 2 presents the results of the Pearson correlation analysis based on 10 trials involving various file extensions, including XLSX, DOCX, JPG, PDF, and PPTX, with the Rail Fence cipher applied using a key size of 3. The analysis compares the Base64-encoded plaintext with the ciphertext generated by the Rail Fence cipher algorithm. The findings consistently demonstrate that the average Pearson correlation coefficient is approximately 0.00 or slightly negative (-0.00), indicating a very weak or essentially no correlation between the two datasets. In the field of cryptography, such an outcome is highly favorable, as it signifies a minimal statistical association between the plaintext and the ciphertext. This absence of correlation suggests that the encryption process effectively transforms the plaintext into a randomized sequence that is statistically independent of its original structure, thereby enhancing the unpredictability and security of the ciphertext. The randomness indicated by this lack of correlation contributes to the cipher's resistance to statistical and cryptanalytic attacks, ultimately strengthening the confidentiality and integrity of the protected data.

Table 2. Pearson Correlation Analysis Result

No	File	Rail (key)	Base64	Rail Fence Cipher	Pearson Correlation
1.	excel_one.xlsx	3	UesDB… AAAA	UBBAI… cAA4A	0.0054762160302451
2.	excel_two.xlsx	3	UesDB… AAA==	UBBAI… oI2A=	-0.0015123902837401
3.	jpg_one.jpg	3	/9j/4…B/9k=	/4SRA… d8Vfk	-0.01659774571885
4.	jpg_two.jpg	3	/9j/4… f/9k=	/4RZS… Diomk	-0.0029870957094411
5.	msword_one.docx	3	UesDB… AAA==	UBBAI… wMSA=	0.0023166717367943
6.	msword_two. docx	3	UesDB… AAAAA	UBBAI… IDAMA	0.013524173737449
7.	pdf_one.pdf	3	JVBER… PRgo=	JRLJz… YkoVo	0.00057255410312938
8.	pdf_two.pdf	3	JVBER… FT0Y=	JRLCt… EI0VY	-0.0031939823667947
9.	ppt_one.pptx	3	UesDB… AAAA	UBBAI… EAApA	0.00018997790720693

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

10.	ppt_two.pptx	3	UesDB… WAAAA	UBBAI… MDABA	0.00050871635587214

In the subsequent file compression experiments summarized in **Table 3**, ten trials were conducted using files with various extensions, including XLSX, JPG, DOCX, PDF, and PPTX. The results consistently show that, on average, the file sizes were reduced by approximately 1 KB after applying the Gzip compression algorithm. This reduction demonstrates the capability of Gzip to achieve a modest degree of compression, which can contribute positively to storage efficiency and optimize resource utilization, particularly when processing multiple files or managing limited storage environments.

However, it is noteworthy that in the third trial, no file size reduction was observed following the compression process. This outcome can be attributed to the characteristics of the Gzip algorithm, which may yield minimal or no compression gains when applied to files that are already small or contain limited redundancy. In such cases, the compression overhead may equal or exceed the potential savings, resulting in unchanged file sizes. Despite these limitations, the overall compression performance remains beneficial, especially for larger datasets where redundancy and data patterns offer greater opportunities for compression.

Table 3. The Result of Gzip Compression

No	File Name	Before Extention	Before Compress	After Compress	After Extention
1.	excel_one	.xlsx	82 KB	78 KB	.gz
2.	excel_two	.xlsx	42 KB	41 KB	.gz
3.	jpg_one	.jpg	7 KB	7 KB	.gz
4.	jpg_two	.jpg	183 KB	177 KB	.gz
5.	msword_one	.docx	36 KB	34 KB	.gz
6.	msword_two	.docx	27 KB	24 KB	.gz
7.	pdf_one	.pdf	1.608 KB	1.495 KB	.gz
8.	pdf_two	.pdf	129 KB	121 KB	.gz
9.	ppt_one	.pptx	5.707 KB	5.630 KB	.gz
10.	ppt_two	.pptx	5.559 KB	5.468 KB	.gz

4. CONCLUSION

In conclusion, this study successfully demonstrates the implementation and performance analysis of a hybrid digital data security approach that integrates Base64 encoding, the Rail Fence cipher algorithm, and Gzip compression. The combination of these three techniques offers a multi-layered security framework that enhances data confidentiality, integrity, and storage efficiency. The Pearson correlation analysis reveals that the ciphertext generated by the Rail Fence cipher exhibits an average correlation coefficient of approximately 0.00 or slightly negative when compared to the Base64-encoded plaintext, indicating a statistically insignificant relationship between the two datasets. This absence of correlation reflects the effectiveness of the encryption process in producing a randomized ciphertext that resists statistical and cryptanalytic attacks. Furthermore, the Gzip compression trials show an

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

average file size reduction of approximately 1 KB across multiple file types, including XLSX, JPG, DOCX, PDF, and PPTX. Although some trials demonstrated no change in file size due to the limitations of Gzip when compressing already small or less redundant files, the overall compression performance remains beneficial for optimizing storage resources, particularly for larger datasets.

For future research, it is recommended to explore the integration of more advanced or modern encryption algorithms alongside Base64 and Rail Fence to further enhance the security robustness of the system. Additionally, examining the performance of alternative compression algorithms such as BZIP2, LZMA, or Zstandard may provide better compression efficiency, especially for diverse file types and larger datasets. Future studies may also consider evaluating the proposed method under real-world scenarios involving network transmission, cloud storage, and resistance against various cryptographic attacks, to validate its practical applicability and scalability in modern data security infrastructures.

5. ACKNOWLEDGMENT

The authors would like to express their deepest gratitude to the Computer Engineering Study Program at Universitas Pendidikan Indonesia (UPI) Cibiru Campus for the continuous academic guidance, research facilities, and valuable support throughout the completion of this study. Special appreciation is also extended to Universitas Widyatama and Universitas Informatika dan Bisnis Indonesia (UNIBI) for their collaborative contributions, insightful discussions, and institutional support that greatly enhanced the quality and scope of this research. The constructive feedback and academic resources provided by all parties involved have played a significant role in the successful execution and completion of this research project.

6. AUTHORS' NOTE

All authors contributed equally to the development and completion of this research. The first author is affiliated with the Computer Engineering Study Program, Universitas Pendidikan Indonesia (UPI) Cibiru Campus. The second author is affiliated with Universitas Widyatama, and the third author is affiliated with Universitas Informatika dan Bisnis Indonesia (UNIBI). The authors declare no conflict of interest related to this study. Correspondence concerning this article should be addressed to dedenpradeka@upi.edu

7. REFERENCES

[1] Pradeka, D., Adiwilaga, A., Agustini, D. A. R., Suheryadi, A., & Nuriman, R. (2023). Design and Build an Assessment Platform by Inserting Moodle-Based Cryptographic Methods. Jurnal Nasional Teknologi dan Sistem Informasi, 9(3), 264-270.

[2] Kurniawan, M. S., Putra, I. G. A. S., Maheswara, I. M. A., Labamaking, R. Y. M. N., Listartha, I. M. E., & Saskara, G. A. J. (2023). Analisis Efektivitas Dan Efisiensi Metode Encoding Dan Decoding Algoritma Base64. *Jurnal Informatika Dan Tekonologi Komputer (JITEK)*, *3*(1), 24-34.

[3] Rachmawati, D., Budiman, M. A., & Yusuf, A. (2020, May). Combination of Rail Fence Cipher Algorithm and Least Significant Bit Technique to Secure The Image File. In *IOP Conference Series: Materials Science and Engineering* (Vol. 851, No. 1, p. 012069). IOP Publishing.

[4] Sulistyo, F. P. (2018). Aplikasi Server Web Dengan Kompresi GZIP. *Jurnal Ilmu Teknik dan Komputer*, *2*(1), 2621-1491.

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149

[5] Ellis, T. J., & Levy, Y. (2010, June). A guide for novice researchers: Design and development research methods. In *Proceedings of Informing Science & IT Education Conference (InSITE)* (Vol. 10, No. 10, pp. 107-117). Italy, Cassino.

[6] Cotton, W. (2008). Supporting the Use of Learning Objects in The K-12 Environment: A Design-Based Research Project.

[7] Ellis, T. J., & Levy, Y. (2010, June). A Guide for Novice Researchers: Design and Development Research Methods. In *Proceedings of Informing Science & IT Education Conference (InSITE)* (Vol. 10, No. 10, pp. 107-117). Italy, Cassino.

[8] Barmawi, A. M., & Pradeka, D. (2017). Information Hiding Based on Histogram and Pixel Pattern. *Journal of Cyber Security and Mobility*, 397-426.

[9] Pradeka, D., Adiwilaga, A., Agustini, D. A. R., Suheryadi, A., & Nuriman, R. (2023). Design and Build an Assessment Platform by Inserting Moodle-Based Cryptographic Methods. *Jurnal Nasional Teknologi dan Sistem Informasi*, *9*(3), 264-270.

[10] Pradeka, D. (2019). Implementasi Aplikasi Kriptografi Berbasis Android menggunakan Metode Substitusi dan Permutasi. *In Search (Informatic, Science, Entrepreneur, Applied Art, Research, Humanism)*, *18*(1), 161-168.

[11] Aruan, M. C., & Rahayu, W. (2023). Analisis Performa Algoritma Kompresi Data dalam Penyimpanan dan Transfer Data. *LANCAH: Jurnal Inovasi dan Tren*, *1*(2), 228-232.

[12] Wen, S., & Dang, W. (2018, June). Research on base64 encoding algorithm and PHP implementation. In *2018 26th International Conference on Geoinformatics* (pp. 1-5). IEEE.

[13] Mesran, M., Abdullah, D., Hartama, D., Roslina, R., Asri, A., Rahim, R., & Ahmar, A. S. (2018, June). Combination Base64 and Hashing Variable Length for Securing Data. In *Journal of Physics: Conference Series* (Vol. 1028, p. 012056). IOP Publishing.

[14] Red Hat. (n.d.). *Base64 encoding*. Red Hat. Retrieved January 15, 2025, from https://www.redhat.com/en/blog/base64-encoding

[15] Godara, S., Kundu, S., & Kaler, R. (2018). An improved Algorithmic Implementation of Rail Fence Cipher. International Journal of Future Generation Communication and Networking, 11(2), 23-32.

[16] Gailly, J. L., & Adler, M. (1992). Gnu gzip. *GNU Operating System*, 8-18.

[17] Beddolo, C. A., & Budiman, H. (2024). Kriptografi Hill Cipher Menggunakan Matriks Fibonacci. *Jurnal Matematika Komputasi dan Statistika*, *4*(3), 790-800.

[18] Zala, D. K., & Khan, M. A. (2024). *Compressive Method for Securing Text Data Using Base64 Encoding and Fernet Cryptography*. 2024 2nd International Conference on Cyber Security and Digital Forensics. IEEE.https://ieeexplore.ieee.org/document/10625646

[19] Speidel, U., & Manoharan, S. (2022). Strengthening Puzzle-based Learning with Individualization. IEEE Global Engineering Education Conference (EDUCON). https://ieeexplore.ieee.org/document/9820231

[20] Finck, C., & Dohrmann, T. (2023). A Decade After Stuxnet: How Siemens S7 is Still an Attacker's Heaven. Blackhat Conference Europe. https://colinfinck.de/files/EU-23-Finck-A-Decade-After-Stuxnet-How-Siemens-S7-is-Still-an-Attackers-Heaven-wp.pdf

DOI: https://doi.org/10.17509/coelite.v4i1.82498
p- ISSN 2829-4157 e- ISSN 2829-4149