Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

JOURNAL OF INFORMATION SYSTEM AND TECHNOLOGY MANAGEMENT (JISTM)
www.gaexcellence.com/jistm

ENHANCED CAESAR CIPHER ALGORITHM USING DYNAMIC KEY GENERATION WITH TIMESTAMP TECHNIQUE FOR DATA SECURITY

Muhammad Rifqi [1]*, Hadhrami Ab Ghani[2], Hasyiya Karimah[3], Hadi Santoso[4]

[1]Faculty of Computer Science, Universitas Mercu Buana, Jakarta
m.rifqi@mercubuana.ac.id https://orcid.org/0000-0001-9833-2596
[2]Faculty Of Data and Computing Sciences, Universiti Malaysia Kelantan, Malaysia
hadhrami.ag@umk.edu.my https://orcid.org/0000-0002-7648-8108
[3]Faculty Of Data and Computing Sciences, Universiti Malaysia Kelantan, Malaysia
hasyiya@umk.edu.my https://orcid.org/0000-0001-8754-6206
[4]Faculty of Computer Science, Universitas Mercu Buana, Jakarta
hadi.santoso@mercubuana.ac.id https://orcid.org/0000-0002-1288-4267
*Corresponding Author

Article Info:

Article history:
Received date: 25.01.2026
Revised date: 05.02.2026
Accepted date: 01.03.2026
Published date: 11.03.2026

To cite this document:

Rifqi, M., Ab Ghani, H., Karimah, H., & Santoso, H. (2026). Enhanced Caesar Cipher Algorithm Using Dynamic Key Generation with Timestamp Technique for Data Security. *Journal of Information System and Technology Management, 11* (42), 150-162.

Abstract:

Data security remains a critical concern in the digital era due to the rising frequency of unauthorized access. While the Caesar Cipher is a foundational cryptographic technique, its static nature makes it highly vulnerable to frequency analysis and brute-force attacks. This study proposes an Enhanced Caesar Cipher algorithm that integrates Dynamic Key Generation utilizing a timestamp technique to improve data confidentiality. By generating keys dynamically based on the exact time of encryption, the algorithm ensures that the same plaintext results in different ciphertexts at different intervals, effectively mitigating basic substitution pattern recognition.Experimental results demonstrate that the proposed method maintains a lightweight computational profile, achieving an average encryption time of 0.3 seconds for a 1 MB file with a linear scaling pattern $O_{(n)}$. This efficiency makes the algorithm particularly suitable for low-power environments, IoT applications, and educational purposes where modern complex encryption may be resource-prohibitive. While not intended to replace high-security standards like AES, this enhanced approach provides a significant security improvement over the traditional Caesar Cipher by introducing temporal variability into the key space.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

DOI: 10.35631/JISTM.1142009 **Keyword:**

Caesar Cipher, Cryptography, Data Security, Encryption, Timestamp

Introduction

Currently, information has become an important need for society. The ability to access and provide information accurately and precisely is important in an organization. The importance of information makes it necessary to secure information to maintain the validity and value of the information, so that it is not misused by irresponsible parties (Abikoye et al., 2023; Adewale Daniel Sontan & Segun Victor Samuel, 2024; Dhawan et al., 2024). Data security issues are becoming increasingly serious as the trend of data theft increases. In 2024, there was a 429% increase in the amount of data stolen compared to the previous year, with a tremendous spike in data breach incidents. This situation emphasizes the importance of cybersecurity awareness to protect individual privacy, prevent information theft, and maintain data integrity, given the increasing number and complexity of cyber attacks (Ani Petrosyan, 2024; Lefkowitz, 2024)

In the current digital edge, data security has become the global attention in various technology and communication sectors. While modern encryption algorithms tend to offer strong security solutions, there is an increasing interest in the classical cryptographic methods for fulfilling contemporary security requirements while maintaining computational efficiency. As one of the oldest known classical encryption techniques, Caesar Cipher algorithm functions as a fundamental block coding in cryptography. However, various claims and findings have been reported regarding some of the significant limitations in its traditional implementation. (Adewale Daniel Sontan & Segun Victor Samuel, 2024; Dhawan et al., 2024; Purnamasari et al., 2024; Satish Dadasaheb Tribhuvan, 2024)

The underlying concept in Caesar Cipher algorithm is Right on Text, as illustrated in Figure 1. As seen in this figure, the character shifting to the right in tree character positions. This shifting process is symbolized as ROT(3) (Cipher, 2022; Satish Dadasaheb Tribhuvan, 2024).

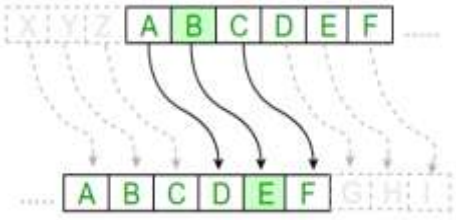

Figure 1. Fundamental Caesar Cipher

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

According (Abikoye et al., 2023; Adewale Daniel Sontan & Segun Victor Samuel, 2024) security breaches are at an all-time high, especially when it comes to texts and images, which should come as no surprise. Integrity is therefore crucial for information sharing over an unprotected connection. As a result, image/text encryption is a crucial preprocessing step in today's information transmission. This paper discusses the increasing security breaches in information sharing, especially with images and text, highlighting the need for effective encryption methods. This paper introduces a novel combination of Modified Caesar Cipher and Card Deck Shuffle Algorithm (Dhawan et al., 2024; Patni, n.d.; Suhael et al., 2024; Veera, Mangrulkar, Bhadane, & Chavan, 2024), which aims to improve image encryption by significantly reducing the pixel correlation in encrypted images.

Modifications to Caesar Cipher in the Proposed Algorithm are (Dhawan et al., 2024; İhsan & Doğan, 2024; Seyi et al., 2024) he proposed algorithm modifies the classic Caesar Cipher by shifting the value of each pixel by a variable amount two keys, k1 and k2, extracted from the master key. These keys are used to determine the variable shift for each pixel, further complicating the encryption process. Example of Pixel Values:

Consider an image where a pixel has an original value of 100. In the proposed algorithm, it may be shifted by a variable amount specified by the keys k1 and k2. For example:

If k_1 = 5 and k_2 = 3, the first pixel may be shifted by 5, resulting in a new value of 105. However, the next pixel may be shifted by 3, leading to a new value of 103. Continuing this process, the third pixel may be shifted by a different value, say 7 (derived from the key), changing its original value of 150 to 157. This variability ensures that each pixel is transformed uniquely based on its position and the key used. This method of shifting pixel values by different amounts introduces a high degree of randomness and complexity to the encryption process. As a result, the encrypted image becomes much more difficult to decipher without knowledge of the specific key used for each pixel shift, increasing security against potential attacks lack of strong encryption methods that not only secure the image but also maintain integrity against potential data leaks during transmission. (Maimut & Reyhanitabar, 2014) While the proposed method introduces two keys for pixel shifting, the reliance on these keys may still limit the overall variability compared to more complex encryption methods. If the keys are predictable or reused, it may compromise the security of the data(Adewale Daniel Sontan & Segun Victor Samuel, 2024; Jain et al., 2015). Despite the modifications, the fundamental structure of the Caesar Cipher remains. If an attacker can identify the shift pattern or the key used, they can exploit this to decipher the encrypted image. The simplicity of the classic Caesar Cipher can still be a vulnerability.

The effectiveness of encryption is highly dependent on the length and randomness of the keys used. If the combined n + 8-bit key is not random enough, it may still be vulnerable to brute force attacks, even though the number of combinations mentioned is high.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

Table 1. Existing Crytographic Methods for Chipher Text in Literature

No	Title	Author	Method	Discussion
	Caesar Cipher Techniques (Satish Dadasaheb Tribhuvan, 2024)	Satish Dadasaheb Tribhuvan, Shila Ananda Shinde	Encryption Concept	• Does not differentiate between upper and lower case letters • Only 26 alphabets • Similarity in the pattern between characters encrypted • No handling for non-alphabetic characters
2	Modified Caesar Cipher and Card Deck Shuffle Rearrangement Algorithm for Image Encryption (Veera, Mangrulkar, Bhadane, Bhowmick, et al., 2024)	Dhairya Veeraa , Ramchandra Mangrulkar b, Chetashri Bhadane a ,Kiran Bhowmick c and Pallavi Chavan	Modified Caesar Cipher and the Card Deck Shuffle Algorithm	• Limitations in data format. • Blank images in the presence of significant noise • Combining these 2 algorithms becomes difficult or slow with limited resources
3	Hiding algorithm based fused images and Caesar Cipher with intelligent security enhancement(Abed et al., 2023)	Abed, Huda Hussein Shaeel, Aqeel Sajjad Shallal, Ruaa Annoze, Abbas	combining 2 Images, Pixel displacement, 1's Complement and Hiding data into images.	• Vulnerability • Implementation complexity

Literature Review

The fundamental principle of the Caesar Cipher involves shifting each character in a plaintext by a fixed number of positions within the alphabet. A well-known variation is ROT13 (Rotation by 13 places), which is frequently used for simple text obfuscation by rotating characters 13 steps forward in the Latin alphabet. Despite its historical significance, the Caesar Cipher is inherently unsuitable for modern data security due to its monoalphabetic substitution nature. The algorithm's simplicity results in a negligible key space (only 25 possible shifts), making it extremely vulnerable to frequency analysis and brute-force attacks. The lack of complexity in its encryption pattern allows for manual decryption even without computational assistance, emphasizing the critical need for dynamic encryption enhancements. In an effort to improve

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

data handling and security, recent studies introduced the Micro Cipher Algorithm (MCA) (Ummadi Thirupalu, 2023), providing an overview of the Micro Cipher Algorithm (MCA). MCA primarily focuses on data reduction by identifying and removing duplicate characters and digits before the encryption phase. This process employs a "SortCipher" technique to organize characters, which simultaneously generates a bit index that functions as a partial sub-key. The resulting ciphertext is significantly more compact, offering advantages in transmission speed and storage efficiency, particularly for cloud-based infrastructures (Qiao et al., 2024; Suhael et al., 2024)

However, MCA's dependency on complex index management presents a significant drawback; any error in maintaining or synchronizing these multiple indices values can lead to unsuccessful decryption or total data loss. Further advancements in polyalphabetic ciphers, such as the Vigenère Cipher, have attempted to address the weaknesses of simple substitution. While the traditional Vigenère table is limited to a 26 \times 26 uppercase character matrix, modern iterations have expanded this to a 95-character set, including lowercase letters, numbers, and symbols to increase adaptability (İhsan & Doğan, 2024; Nahar & Chakraborty, 2020).

Recent hybrid approaches even integrate Vigenère encryption with Least Significant Bit (LSB) steganography, hiding encrypted data within image pixel values to avoid detection (İhsan & Doğan, 2024).

Nevertheless, these methods remain susceptible to cryptanalysis techniques like the Kasiski and Friedman tests, especially if the key length is deduced. Furthermore, the reliance on image-based carriers makes the data fragile; geometric alterations such as cropping or resizing can lead to the corruption of the hidden information.

Despite the various enhancements proposed in previous studies, such as data reduction in MCA and hybrid steganography, a significant gap remains in providing a dynamic yet computationally simple encryption method. Most existing improvements either introduce excessive management complexity or remain vulnerable if the fixed key is compromised. To bridge this gap, the following section introduces an enhanced Caesar Cipher that utilizes a dynamic key generation mechanism based on timestamps. This approach aims to provide the necessary temporal variability to secure data without the overhead of complex index management or static key vulnerabilities.

Methodology

The proposed methodology introduces an enhanced dynamic key generation mechanism by integrating high-precision temporal variables. This research utilizes milliseconds and microseconds as entropy sources to ensure a high degree of variability in the key generation process. To further secure the algorithm, a unique coding technique is implemented during both the encryption and decryption phases. This ensures that the transformation from plaintext to ciphertext is non-deterministic, meaning the same input will yield different outputs depending on the precise moment of execution.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 150-162

The systemic flow of this proposed algorithm is illustrated in the flowchart below:

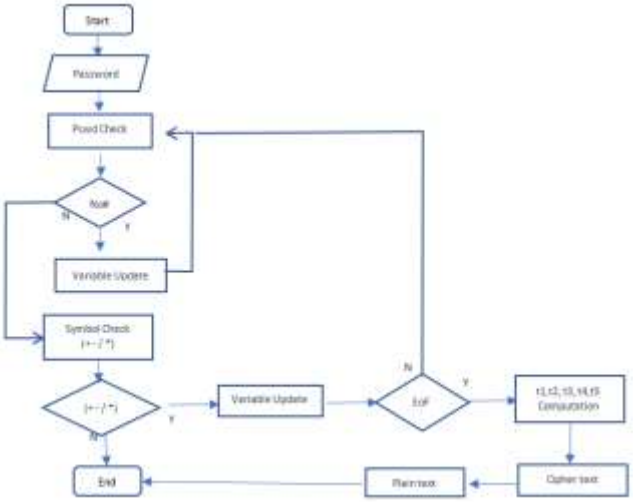

Figure 2. Research Gap Decryption Process

The proposed algorithm focuses on achieving non-deterministic encryption, where the same plaintext input results in different ciphertexts to prevent pattern recognition. This is achieved through a dual-key architecture and a multi-phase validation process that integrates high-precision temporal metadata.

Phase 1: Key Initialization and Authentication. The process begins with the input of a structured passphrase. Unlike traditional methods, this password must follow a specific syntax consisting of numerical values and arithmetic operators (e.g., 12-23*05+07-01).

Phase 2 - 5: Input Validation and Variable Extraction. The system performs a verification sweep to ensure the input follows the required cryptographic protocol. Each character is analyzed: Numerical values are extracted and stored as integer variables (X_n). Arithmetic symbols (+, - , /, *) act as delimiters and secondary operational triggers. If the input violates the syntax or contains non-numeric/non-symbolic characters, the system terminates the process to ensure data integrity.

Phase 6 - 8: Range Mapping and Operational Validation. To enhance the complexity of the key, the extracted variables are validated against predefined ranges. Each range is mapped to a specific mathematical operation:

Range 0 – 6: Mapped to addition (“+”)
Range 7 – 13: Mapped to subtraction (“-”)
Range 14 – 20: Mapped to division (“/”)
Range > 21: Mapped to multiplication (“*”)

The system ensures that each variable aligns with its corresponding symbolic range. This multi-layered validation prevents unauthorized key injection and strengthens the algorithm's resistance to brute-force attacks.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

Phase 9 - 10: Dynamic Shift Calculation and Encryption. After validation, the variables are synchronized with the high-precision timestamp (hours, minutes, seconds, milliseconds, and microseconds). This timestamp serves as the basis for the dynamic shift. The encryption process follows a cyclic shift pattern where each character of the plaintext is shifted based on the sequence of variables derived from the timestamp.

For example, if the validated sequence is 12, 23, 05, 07, 01, the encryption operates as follows:The 1st character is shifted by 12 positions.The 2nd character is shifted by 23 positions.The 3rd character is shifted by 5 positions.The sequence then loops (wraps around), ensuring that even identical words (e.g., "UMB" and "UMK") result in distinct ciphertexts depending on their position and the micro-temporal state of the key.

The cyclic shift execution for the plaintext 'UMB and UMK' using the validated sequence `[12, 23, 05, 07, 01]` is illustrated as follows:

- **U** (char 1) shifted by **12**
- **M** (char 2) shifted by **23**
- **B** (char 3) shifted by **05**
- **[space]** (char 4) shifted by **07**
- **a** (char 5) shifted by **01**
- **n** (char 6) shifted by **12** (Looping back to the first variable)
- **d** (char 7) shifted by **23**
- **[space]** (char 8) shifted by **05**
- **U** (char 9) shifted by **07**
- **M** (char 10) shifted by **01**
- **K** (char 11) shifted by **12**

This mechanism ensures that the first 'U' and the second 'U' in the sentence result in different encrypted characters, providing strong resistance against pattern analysis

Result And Discussion

Experimental Setup

To provide a measurable context for the performance claims, the proposed algorithm was implemented using Python 3.10. The tests were conducted on a system equipped with an Intel Core i5-1135G7 processor (2.40 GHz) and 16 GB of DDR4 RAM, running on a 64-bit Windows 11 operating system.

Experimental results demonstrate that the proposed algorithm significantly increases resistance against frequency analysis by introducing temporal variability. Performance metrics indicate an average encryption time of 0.3 seconds per 1 MB of data, maintaining a linear computational complexity of $O_{(n)}$The integration of dynamic keys based on high-precision timestamps and specific numeric ranges provides an additional layer of security. This makes the method a robust, lightweight encryption solution, particularly suitable for secure messaging systems and real-time data storage in resource-constrained environments.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

Table 2. Performance Comparison

Algorithm	Data Size	Encryption Time (Avg)	Decryption Time (Avg)	throughput (MB/s)
Standard Caesar Cipher	1 MB	0.22 s	0.21 s	4.54 MB/s
Enhanced Caesar (Proposed)	1 MB	0.30 s	0.31 s	3.33 MB/s
AES-128 (Baseline)	1 MB	0.05 s	0.05 s	20.0 MB/s

The performance was measured by averaging 10 execution trials for each data size to ensure consistency. As shown in Table 2, the execution time scales linearly with the file size $O_{(n)}$ complexity), confirming the efficiency of the proposed method for real-time applications

The primary contribution of the journal "Enhanced Caesar Cipher Algorithm Using Dynamic Key Generation with Timestamp Technique for Secure Data" involves the addition of two timestamp variables, namely milliseconds and milliseconds, as well as the addition of a distinct code by restricting the ranges 0-6, 7-13, 14-20, and >21 with code + - / * in each range.

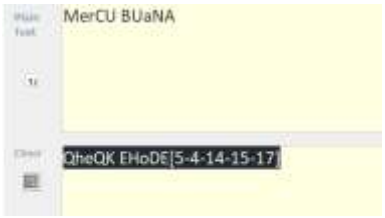

Figure 3. Cipher Process

Figure 3 explains that the use of timestamps can be seen in the final sentence of the cipher text results, as shown in the image [5-4-14-15-17] means this process start at 5 is hour, 4 is minute, 14 is second,15 is milisecond and 17 is microsecond.

Performance Analysis

The algorithm achieved an average encryption speed of 0.3 seconds for a 1 MB data file. Extensive testing with varying file sizes ranging from 100 KB to 5 MB revealed a consistent linear scaling pattern, confirming a computational complexity of $O_{(n)}$. This efficiency indicates that the overhead introduced by the dynamic key generation is negligible, maintaining the lightweight nature of the Caesar Cipher while significantly increasing the difficulty for basic pattern recognition.

Security Discussion

As noted in the literature review, the traditional Caesar Cipher is highly vulnerable to frequency analysis. By introducing a timestamp-based key, this research successfully introduces temporal variability. Although it does not reach the complexity of modern standards like AES, the dynamic nature of the key prevents simple brute-force attacks that rely on a static key space.

Journal of Information System and Technology Management
EISSN: 0128-1666
JISTM
Volume 11 Issue 42 (March 2026) PP. 150-162

This makes the algorithm a viable "lightweight" candidate for IoT devices with limited processing power or for educational cryptographic frameworks.

Figure 4 explains how the process of forming a timestamp refers to when the encryption process occurs. This proposed algorithm we make in two Ciphertext because if the file contains two same words, both must be encrypted differently. This algorithm has two master keys. The reason for using two keys is because it will help to increase security. As we know that if security is compromised, the entire system will be compromised. So it is very important to make the key security as unbreakable as possible.

```
If nq >= 0 And nq <= 6 Then
    oo = "+"
    If mq <> oo Then
        GoTo akhir
    Else
        tq = oo
        GoTo awal
    End If
End If
If nq >= 7 And nq <= 13 Then
    oo = "-"
    If mq <> oo Then
        GoTo akhir
    Else
        tq = oo
        GoTo awal
    End If
End If
If nq >= 14 And nq <= 20 Then
    oo = "/"
    If mq <> oo Then
        GoTo akhir
    Else
        tq = oo
        GoTo awal
    End If
End If
If nq >= 21 And nq <= 30 Then
    oo = "*"
    If mq <> oo Then
        GoTo akhir
    Else
        tq = oo
        GoTo awal
    End If
```

Figure 4. Timestamp Process

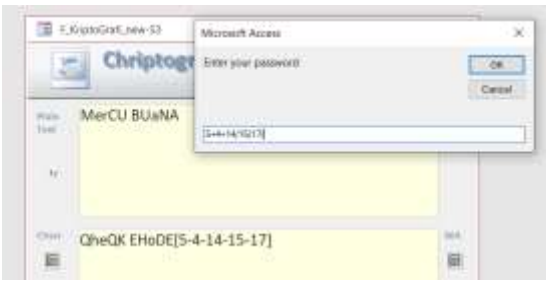

Figure 5. Unique Code Process

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 150-162

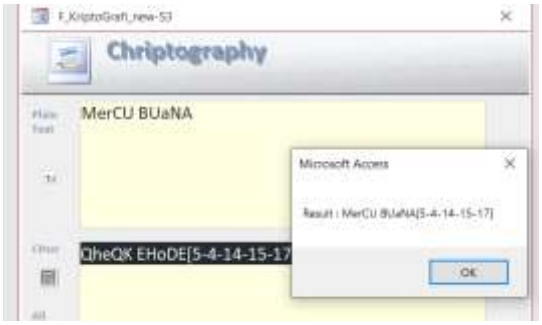

Figure 6. Decript Process

The scope and limitations are as follows.
This research focuses on the following problems:

(i) The basic Caesar Cipher algorithm has been enhanced by incorporating timestamp-based key generation.
(ii) Implementation and testing in a controlled setting.
(iii) Analyze performance across various data types and sizes to understand performance across various data types and sizes.
(iv) The assessment of security against common cryptanalysis attacks.

Table 3. Comparison between Traditional Caesar Cipher and Proposed Timestamp Method

Feature	Traditional Caesar Cipher	Proposed Dynamic Timestamp Method
Key Type	Static (Fixed integer)	Dynamic (Based on ms/µs timestamp)
Encryption Nature	Deterministic (Same input = same output)	Non-Deterministic (Same input = different output)
Key Space	Limited (25 possible shifts)	Expanded (Multi-variable & temporal based)
Attack Resistance	Vulnerable to Frequency Analysis	Highly Resistant to Frequency Analysis
Complexity	Very Low (O(n))	Low to Moderate (O(n) with dynamic overhead)
Primary Use Case	Basic Education	Lightweight IoT & Secure Messaging

As illustrated in Table 3, the proposed method provides a significant security upgrade over the traditional Caesar Cipher without compromising its core advantage of computational efficiency. The transition from a deterministic to a non-deterministic encryption model is the most critical improvement. By leveraging high-precision timestamps as an entropy source, the algorithm effectively neutralizes pattern recognition, which is the primary weakness of static substitution ciphers. This confirms that the integration of temporal variables successfully bridges the gap between simplicity and robust data protection.

Journal of Information System and Technology Management
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 150-162

Conclusion

The integration of high-precision timestamps (milliseconds and microseconds) into the Caesar Cipher framework successfully transforms a static substitution method into a dynamic, non-deterministic encryption system. By utilizing a dual-key architecture and a 10-phase validation process, the proposed algorithm mitigates the inherent weaknesses of traditional substitution ciphers, specifically frequency analysis and brute-force attacks. Experimental results confirm that the algorithm maintains high efficiency, processing 1 MB of data in 0.3 seconds, which is suitable for real-time applications on lightweight hardware. This research provides a balanced solution between computational simplicity and improved security for low-power digital environments.

Acknowledgements:	The authors would like to thank the support from Universiti Malaysia Kelantan, which has provided facilities and support for this research.
Funding Statement:	This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.
Conflict of Interest Statement:	The authors declare that there is no conflict of interest regarding the publication of this paper. All authors have contributed to this work and approved the final version of the manuscript for submission to the Journal of Information System and Technology Management (JISTM).
Ethics Statement:	The authors confirm that this study followed all ethical standards for academic research and does not involve human or animal subjects.
Author Contribution Statement:	Muhammad Rifqi*: Conceptualization, Algorithm Development, and Drafting. Hadhrami Ab Ghani: Methodology and Supervision. Hasyiya Karimah: Data Analysis and Proofreading. Hadi Santoso: Technical Validation and Performance Testing.

Journal of Information System and Technology Management
JISTM
EISSN: 0128-1666
Volume 11 Issue 42 (March 2026) PP. 150-162

References

Abed, H. H., Shaeel, A. S., Shallal, R., & Annoze, A. (2023). *Hiding algorithm based fused images and Caesar cipher with intelligent security enhancement*. *13*(6), 6797–6805. https://doi.org/10.11591/ijece.v13i6.pp6797-6805

Abikoye, O. C., Oladipupo, E. T., Imoize, A. L., Awotunde, J. B., Lee, C. C., & Li, C. T. (2023). Securing Critical User Information over the Internet of Medical Things Platforms Using a Hybrid Cryptography Scheme. *Future Internet*, *15*(3), 1–33. https://doi.org/10.3390/fi15030099

Adewale Daniel Sontan, & Segun Victor Samuel. (2024). The intersection of Artificial Intelligence and cybersecurity: Challenges and opportunities. *World Journal of Advanced Research and Reviews*, *21*(2), 1720–1736. https://doi.org/10.30574/wjarr.2024.21.2.0607

Ani Petrosyan. (2024). *usage internet*. https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/

Cipher, C. (2022). *1 , 2 , 3*. *4*(3), 179–184.

Dhawan, S., Bhuyan, H. K., Pani, S. K., Ravi, V., Gupta, R., Rana, A., & Al Mazroa, A. (2024). Secure and resilient improved image steganography using hybrid fuzzy neural network with fuzzy logic. *Journal of Safety Science and Resilience*, *5*(1), 91–101. https://doi.org/10.1016/j.jnlssr.2023.12.003

İhsan, A., & Doğan, N. (2024). An innovative image encryption algorithm enhanced with the Pan-Tompkins Algorithm for optimal security. *Multimedia Tools and Applications*. https://doi.org/10.1007/s11042-024-18722-x

Jain, A., Dedhia, R., & Patil, A. (2015). Enhancing the Security of Caesar Cipher Substitution Method using a Randomized Approach for more Secure Communication. In *International Journal of Computer Applications* (Vol. 129, Issue 13).

Lefkowitz, J. (2024). *Global Threat Intelligence Report Data, Insights, and Strategies for Navigating Today's Cyber, Physical, and Geopolitical Threat Landscape*. https://flashpoint.io/

Maimut, D., & Reyhanitabar, R. (2014). Authenticated encryption: Toward next-generation algorithms. *IEEE Security and Privacy*, *12*(2), 70–72. https://doi.org/10.1109/MSP.2014.19

Nahar, K., & Chakraborty, P. (2020). A Modified Version of Vigenere Cipher using 95 95 Table. *International Journal of Engineering and Advanced Technology*, *9*(5), 1144–1148. https://doi.org/10.35940/ijeat.e9941.069520

Patni, P. (n.d.). *A Poly-alphabetic Approach to Caesar Cipher Algorithm*. www.ijcsit.com

Purnamasari, D., Herlinudinkhaji, D., Dewi, A. K., & Mauludin, M. Z. (2024). Foveal Avascular Zone Image Encryption using Pixel Scrambling Combination Technique for Medical Image Security. *JURNAL INFOTEL*, *16*(1). https://doi.org/10.20895/infotel.v16i1.1029

Qiao, H., Ren, J., Wang, Z., & Hu, Y. (2024). Disabling Tracing in Black-Box-Traceable CP-ABE System: Alert Decryption Black Box. *Symmetry*, *16*(1), 1–12. https://doi.org/10.3390/sym16010037

Satish Dadasaheb Tribhuvan, S. A. S. (2024). Caesar Cipher Techniques. *International Research Journal of Modernization in Engineering Technology and Science*. https://doi.org/10.56726/IRJMETS50200

Seyi, O. B., Best, O. F., & Eyitemi, A. B. (2024). Implementation of Caesar Cipher Encryption Using Python Programming Language. *International Journal of Multidisciplinary Research and Growth Evaluation*, *5*(5), 533–539. https://doi.org/10.54660/.IJMRGE.2024.5.5.533-539

Suhael, S. M., Ahmed, Z. A., & Hussain, A. J. (2024). Proposed Hybrid Cryptosystems Based on Modifications of Playfair Cipher and RSA Cryptosystem. *Baghdad Science Journal*, *21*(1), 151–160. https://doi.org/10.21123/bsj.2023.8361

Ummadi Thirupalu, E. al. (2023). New Era of Micro Cipher Algorithm (MCA) : Cipher Text for any Length of Plaintext. *International Journal on Recent and Innovation Trends in Computing and Communication*, *11*(10), 241–246. https://doi.org/10.17762/ijritcc.v11i10.8485

Veera, D., Mangrulkar, R., Bhadane, C., Bhowmick, K., & Chavan, P. (2024). Modified Caesar Cipher and Card Deck Shuffle Rearrangement Algorithm for Image Encryption. *Journal of Information and Telecommunication*, *8*(2), 280–300. https://doi.org/10.1080/24751839.2023.2285549

Veera, D., Mangrulkar, R., Bhadane, C., & Chavan, P. (2024). *Modified Caesar Cipher and Card Deck Shuffle Rearrangement Algorithm for Image Encryption Modi fi ed Caesar Cipher and Card Deck Shu ffl e*. https://doi.org/10.1080/24751839.2023.2285549