International Journal of Engineering Innovations and Technology
ISSN : XXXX-XXXX | Vol 1, Issue 1, Jan – Feb, 2026
Published by RGA Research Publications

A NON-LINEAR EXTENSION OF THE CLASSICAL SHIFT CAESER CIPHER USING PARITY BASED DIRECTIONAL ENCRYPTION

[1*]Hassan Aliyu, [1]Abdullahi Aminu, [1] Aliyu Ummulkhairi Buhari
[1]*Federal University Birnin Kebbi, Nigeria*

Received: 02 Jan 2026 | Accepted: 25 Jan 2026 | Published: 28 Jan 2026

Abstract.

The Caesar cipher, one of the oldest and simplest encryption techniques, shifts each letter in a message by a fixed number of positions. Despite its historical significance, it offers minimal security by modern standards because the same plaintext letter always encrypts to the same ciphertext letter, preserving frequency patterns that attackers can easily use. This paper proposes an enhancement called the Index-Modulated Shift Cipher, which introduces position-dependent directional shifting to overcome this fundamental weakness. In the proposed algorithm, the character's index determines the direction of the shift letters at even positions shift forward by a fixed key, while letters at odd positions shift backward by the same key. This simple modification ensures that identical plaintext letters appearing at different positions encrypt to different ciphertext letters, effectively flattening frequency distributions and resisting pattern-based cryptanalysis. The mathematical formulation is presented, along with step-by-step encryption and decryption examples for validation. A security analysis compares the proposed method with the original Caesar cipher, demonstrating significant improvements in resistance to frequency analysis and brute-force attacks while maintaining the same linear time complexity. The Index-Modulated Shift Cipher offers a practical, lightweight upgrade for scenarios where the simplicity of the Caesar cipher is desired but its security flaws are unacceptable.

Keywords: Caesar cipher, index modulation, bi-directional shift, cryptography, frequency analysis, parity-based encryption.

1. Introduction.

Cryptography is the practice of securing communication by converting readable messages into forms that cannot be understood by unauthorized parties. Throughout history, humans have developed many methods to protect their secrets, from simple hand ciphers used by ancient rulers to complex mathematical algorithms that power modern digital security (Stallings., 2017; Azzam & Sumarsono., 2017; Fahrul *et all*., 2017; Hassan *et all*., 2021). Among these methods, the Caesar cipher holds a special place as one of the oldest and most well-known encryption techniques in recorded history. The Caesar cipher operates on a straightforward principle. To encrypt a message, the sender shifts each letter forward by a fixed number of positions in the alphabet. For example, with a shift of three, the letter A becomes D, B becomes E, and so on. To decrypt, the receiver simply shifts each letter backward by the same number using the representations $(A = 0, B = 1, C = 2, \dots, Z = 25)$. This method is named after Julius Caesar, who reportedly used

it to communicate with his military generals (Singh., 1999; Kuriakkottu., 2021; Kashish & Supriya., 2013; Mishra., 2013; Pooja & Pintu., 2017; Shahid., 2014). The beauty of the Caesar cipher lies in its simplicity. It is easy to understand, easy to implement, and requires no special tools or training.

However, this simplicity comes with serious drawbacks. Because every letter shifts by the same amount, the original frequency patterns of the language remain visible in the encrypted text. In English, for instance, the letter E appears more often than any other letter. In a message encrypted with the Caesar cipher, the letter that replaces E will also appear more often than any other letter in the ciphertext (Kahn., 1996; Fahrul *et all*., 2017; Hassan *et all*., 2021). Attackers can use this weakness to break the code without knowing the shift value. This technique, known as frequency analysis, makes the Caesar cipher completely insecure for any serious application.

Despite this vulnerability, the Caesar cipher remains valuable as a teaching tool and as a foundation for understanding more complex cryptographic concepts. Many modern encryption methods build upon the basic idea of shifting or substituting letters, adding layers of complexity to overcome the weaknesses of simple ciphers (Bellovin., 2011; Hassan *et all.,* 2023; Alhassan *et all.,* 2021; Hassan., 2024a; Hassan., 2024b; Hassan *et all.,* 2025; Umar *et all.,* 2024a; Umar *et all.,* 2024b). This paper follows that tradition by proposing a modification that preserves the simplicity of the Caesar cipher while addressing its most significant flaw.

The proposed method, called the Index-Modulated Shift Cipher, introduces a new rule: the position of each letter determines whether it shifts forward or backward. Letters at even positions shift forward by a fixed key, while letters at odd positions shift backward by the same key. This simple change ensures that identical plaintext letters appearing at different positions encrypt to different ciphertext letters. As a result, frequency patterns become flattened and much harder for attackers to exploit.

2. Methodology

Caeser cipher: The Caesar cipher is one of the oldest and most straightforward encryption methods known to humanity. It is named after Julius Caesar, who used it to protect private military communications. The core idea is simple: each letter in the message is replaced by another letter that is a fixed number of positions ahead or behind in the alphabet. This fixed number is called the key or shift value. To use the cipher, the sender and receiver must agree on a secret key beforehand. The sender shifts every letter forward by the key to create the ciphertext. The receiver shifts every letter backward by the same key to recover the original plaintext. If shifting reaches the end of the alphabet, it wraps around to the beginning. This wrapping feature ensures that all letters remain within the alphabet.

Despite its simplicity, the Caesar cipher has a major weakness. Because the shift is constant, the frequency patterns of the original language remain visible in the encrypted text. This makes it vulnerable to frequency analysis, where attackers use the statistical properties of language to break the code.

The **encryption** formula of Caeser cipher is given by

$$C_i = (P_i + k) mod 26$$

Where

C_i is the ciphertext to be produced.

P_i is the plaintext letters to be transformed.

k is the secret key to be used when performing the transformation

$mod26$ is the modulation that will allow wrap around of the values between 0 to 26.

The **Decryption** formula of Caeser cipher is given by

$P_i = (C_i - k) mod 26$

Where

P_i is the plaintext letters to be produced.

C_i is the ciphertext to be transformed.

k is the secret key to be used when performing the transformation

$mod 26$ is the modulation that will allow wrap around of the values between 0 to 26.

Example 1: Encrypting and Decrypting "***HELLO***" with Key **3**

Encryption Steps: using the encryption formula $\boldsymbol{C_i = (P_i + k) mod 26}$

$H\ (position\ 7) + 3 = 10 \rightarrow K$

$E\ (position\ 4) + 3 = 7 \rightarrow H$

$L\ (position\ 11) + 3 = 14 \rightarrow O$

$L\ (position\ 11) + 3 = 14 \rightarrow O$

$O\ (position\ 14) + 3 = 17 \rightarrow R$

Result: HELLO becomes KHOOR

Decryption Steps: using the decryption formula $\boldsymbol{P_i = (C_i - k) mod 26}$

$K (position\ 10) - 3 = 7 \rightarrow H$

$H\ (position\ 7) - 3 = 4 \rightarrow E$

$O\ (position\ 14) - 3 = 11 \rightarrow L$

$O\ (position\ 14) - 3 = 11 \rightarrow L$

$R\ (position\ 17) - 3 = 14 \rightarrow O$

Result: KHOOR becomes ***HELLO***

Example 2: Encrypting and Decrypting "***ZEBRA***" with Key **5**

Encryption Steps:

$Z\ (position\ 25) + 5 = 30 \rightarrow 30\ mod\ 26 = 4 \rightarrow E$

$E\ (position\ 4) + 5 = 9 \rightarrow J$

$B\ (position\ 1) + 5 = 6 \rightarrow G$

$R\ (position\ 17) + 5 = 22 \rightarrow W$

$A\ (position\ 0) + 5 = 5 \rightarrow F$

Result: ***ZEBRA*** becomes ***EJGWF***

Decryption Steps:

$E\ (position\ 4) - 5 = -1 \rightarrow -1 + 26 = 25 \rightarrow Z$

$J\ (position\ 9) - 5 = 4 \rightarrow E$

$G\ (position\ 6)\ -\ 5\ =\ 1\ \rightarrow\ B$

$W\ (position\ 22)\ -\ 5\ =\ 17\ \rightarrow\ R$

$F\ (position\ 5)\ -\ 5\ =\ 0\ \rightarrow\ A$

Result: ***EJGWF*** becomes ***ZEBRA***

These examples demonstrate how the Caesar cipher works in practice. Notice how identical letters, such as the two ***L***'s in ***HELLO***, encrypt to the same ciphertext letters. This uniformity is the weakness that the Index-Modulated Shift Cipher aims to correct.

The Index Modulated Shift Cipher: The Index-Modulated Shift Cipher builds upon the classical Caesar cipher by introducing a simple but powerful change: the direction of the shift depends on the position of each letter. This modification breaks the uniform shifting pattern that makes the original cipher vulnerable to frequency analysis. In this method, the sender and receiver agree on a secret key, just as in the original Caesar cipher. However, instead of shifting every letter in the same direction, the algorithm examines the index of each character. Letters at even positions (0, 2, 4, ...) shift forward by the key. Letters at odd positions (1, 3, 5, ...) shift backward by the same key. This alternating pattern ensures that identical plaintext letters appearing at different positions encrypt to different ciphertext letters.

The mathematical formulas are straightforward. For encryption, if the index i is even, we add the key. If i is odd, we subtract the key. All calculations use modulo 26 to wrap around the alphabet. For decryption, we simply reverse the direction: even indices subtract the key, while odd indices add the key.

This cipher uses the character's position to determine the direction of the shift

Direction rule is given by

$$\begin{cases} +1\ if\ i\ is\ even\ (shift\ forward) \\ -1\ if\ i\ is\ odd\ (shift\ backward) \end{cases}$$

The **Encryption** formula is given by

$C_i = (P_i + (k \times direction_i) mod 26$

Which expands to

$if\ i\ is\ even:\ \ C_i = (P_i + k) mod 26$

$if\ i\ is\ odd:\ \ C_i = (P_i - k) mod 26$

Where

C_i is the ciphertext to be produced.

P_i is the plaintext letters to be transformed.

k is the secret key to be used when performing the transformation

$mod 26$ is the modulation that will allow wrap around of the values between 0 to 26.

The **Decryption** formula is given by

$P_i = (C_i - (k \times direction_i) mod 26$

Which expands to

$if\ i\ is\ even:\ \ P_i = (C_i - k) mod 26$

$if\ i\ is\ odd:\ P_i = (C_i + k) mod 26$

Where

P_i is the plaintext letters to be produced.

C_i is the ciphertext to be transformed.

k is the secret key to be used when performing the transformation

$mod26$ is the modulation that will allow wrap around of the values between 0 to 26.

Generally: for odd indices during encryption, if $P_i - k$ is negative, add 26. For odd indices during decryption, if $C_i + k$ exceeds 25, subtract 26.

Example 3: Encrypting and Decrypting "HELLO" with Key 3

Encryption Steps: using the encryption formula $C_i = (P_i + (k \times direction_i) mod 26$

Index 0 (even, forward): $H\ (position\ 7)\ +\ 3\ =\ 10\ \rightarrow\ K$

Index 1 (odd, backward): $E\ (position\ 4)\ -\ 3\ =\ 1\ \rightarrow\ B$

Index 2 (even, forward): $L\ (position\ 11)\ +\ 3\ =\ 14\ \rightarrow\ O$

Index 3 (odd, backward): $L\ (position\ 11)\ -\ 3\ =\ 8\ \rightarrow\ I$

Index 4 (even, forward): $O\ (position\ 14)\ +\ 3\ =\ 17\ \rightarrow\ R$

Result: HELLO becomes KBOIR

Decryption Steps: using the decryption formula $P_i = (C_i - (k \times direction_i) mod 26$

Index 0 (even, subtract): K (position 10) - 3 = 7 → H

Index 1 (odd, add): B (position 1) + 3 = 4 → E

Index 2 (even, subtract): O (position 14) - 3 = 11 → L

Index 3 (odd, add): I (position 8) + 3 = 11 → L

Index 4 (even, subtract): R (position 17) - 3 = 14 → O

Result: KBOIR becomes HELLO

Example 4: Encrypting and Decrypting "ZEBRA" with Key 5

Encryption Steps:

Index 0 (even, forward): $Z\ (position\ 25)\ +\ 5\ =\ 30\ \rightarrow\ 30\ mod\ 26\ =\ 4\ \rightarrow\ E$

Index 1 (odd, backward): $E\ (position\ 4)\ -\ 5\ =\ -1\ \rightarrow\ -1\ +\ 26\ =\ 25\ \rightarrow\ Z$

Index 2 (even, forward): $B\ (position\ 1)\ +\ 5\ =\ 6\ \rightarrow\ G$

Index 3 (odd, backward): $R\ (position\ 17)\ -\ 5\ =\ 12\ \rightarrow\ M$

Index 4 (even, forward): $A\ (position\ 0)\ +\ 5\ =\ 5\ \rightarrow\ F$

Result: ZEBRA becomes EZGMF

Decryption Steps:

Index 0 (even, subtract): $E\ (position\ 4)\ -\ 5\ =\ -1\ \rightarrow\ -1\ +\ 26\ =\ 25\ \rightarrow\ Z$

Index 1 (odd, add): $Z\ (position\ 25)\ +\ 5\ =\ 30\ \rightarrow\ 30\ mod\ 26\ =\ 4\ \rightarrow\ E$

Index 2 (even, subtract): $G\ (position\ 6)\ -\ 5\ =\ 1\ \rightarrow\ B$

Index 3 (odd, add): $M\ (position\ 12)\ +\ 5\ =\ 17\ \rightarrow\ R$

Index 4 (even, subtract): $F\ (position\ 5)\ -\ 5\ =\ 0\ \rightarrow\ A$

Result: EZGMF becomes ZEBRA

These examples demonstrate how the Index-Modulated Shift Cipher successfully breaks the uniform encryption pattern of the original Caesar cipher. Notice how the two L's in HELLO encrypt to different letters (O and I), and the Z in ZEBRA encrypts differently depending on its position. This simple modification provides enhanced security while maintaining the elegance and simplicity of the original method.

Implementation. In this section, the python code will be implemented in order for the cipher to be implemented without following the manual methods practiced by the older ciphers.

#**encryption part**

```
def encrypt(text, key):
    Encrypts text using the Index-Modulated Shift Cipher.
     Rules:
    - Even indices (0,2,4...): Shift FORWARD by +key
    - Odd indices (1,3,5...): Shift BACKWARD by -key
     Args:
       text (str): The plaintext to encrypt (A-Z only, case insensitive)
       key (int): The shift key (0-25)
     Returns:
       str: The encrypted ciphertext
  text = text.upper().replace(" ", "")  # Remove spaces, convert to uppercase
   result = ""
     for i, char in enumerate(text):
       if char.isalpha():
          # Convert letter to number (A=0 to Z=25)
          p = ord(char) - 65
            # Apply directional shift based on index parity
          if i % 2 == 0:  # Even index - shift forward
             c = (p + key) % 26
          else:  # Odd index - shift backward
             c = (p - key) % 26
           # Convert back to letter
          result += chr(c + 65)
       else:
          # Keep non-alphabetic characters
          result += char
```

```
    return result
#decryption part
def decrypt(text, key):
    Decrypts text encrypted with the Index-Modulated Shift Cipher.
     Args:
        text (str): The ciphertext to decrypt
        key (int): The shift key used for encryption
      Returns:
        str: The decrypted plaintext
    text = text.upper()
     result = ""
      for i, char in enumerate(text):
        if char.isalpha():
            # Convert letter to number
            c = ord(char) - 65
              # Reverse the directional shift
            if i % 2 == 0:  # Even index - was shifted forward, so shift backward
                p = (c - key) % 26
            else:  # Odd index - was shifted backward, so shift forward
                p = (c + key) % 26
              # Convert back to letter
            result += chr(p + 65)
        else:
         result += char
    return result
```

3. **Results and analysis.**

Security analysis: Security Analysis and Key Space

The Index-Modulated Shift Cipher offers significant security improvements over the original Caesar cipher while maintaining its elegant simplicity. These improvements can be understood by examining two key areas: resistance to frequency analysis and the effective key space. The original Caesar cipher fails because it preserves the frequency patterns of the plaintext. The letter E, which appears most often in English, always encrypts to the same ciphertext letter. An attacker can simply count letter frequencies in the ciphertext and match them to known language patterns. The Index-Modulated Shift Cipher breaks this pattern entirely. Because letters shift forward on even positions and backward on odd positions, the same plaintext letter encrypts to different ciphertext letters depending on its location. This flattens the frequency distribution, making statistical attacks ineffective. The original Caesar cipher has only 25 possible keys. An attacker can try every key in seconds and look for readable output. The Index-Modulated Shift Cipher also uses a single key between 1 and 25. However, the effective complexity is higher because the attacker must also understand the directional rule. Even if an attacker tries all 25 keys, the resulting text will remain

scrambled because the direction of shift for each position must also be correct. This doubles the effective search space when the directional pattern is unknown. The method remains vulnerable to known-plaintext attacks. If an attacker obtains even one matching pair of plaintexts and ciphertext, they can calculate the key and decrypt the entire message.

Computational efficiency: The Index-Modulated Shift Cipher maintains the same excellent computational efficiency as the original Caesar cipher. Both algorithms process each character individually, resulting in a time complexity of $O(n)$, where n is the length of the message. The only additional operation in the proposed method is a simple parity check to determine whether the index is even or odd. This check requires minimal processing power and adds no noticeable delay, even for long messages. Memory usage remains identical, requiring only storage for the input and output strings. These qualities make the cipher suitable for resource-constrained environments such as embedded systems, mobile devices, or applications where speed and simplicity are priorities.

4. Conclusion and Recommendations

This paper introduced the Index Modulated Shift Cipher, a simple but effective enhancement of the classical Caesar cipher. By using character position to determine the direction of shifting, the proposed method successfully addresses the primary weakness of the original cipher: its vulnerability to frequency analysis. Identical plaintext letters encrypt to different ciphertext letters depending on their location, flattening frequency patterns and making statistical attacks ineffective. The algorithm maintains the same linear time complexity as the original, adding only a simple parity check per character. These qualities make it a practical upgrade for scenarios requiring lightweight encryption with improved security. Future research could explore several promising directions. First, investigators might examine non-linear index functions, such as using prime numbers to determine direction changes. Second, combining this method with a keyword-based system could add another layer of security. Third, researchers could extend the algorithm to support larger character sets, including digits and symbols. Fourth, formal cryptanalysis using information theory metrics would help quantify the confusion and diffusion properties of the cipher. Finally, developing a hybrid model that switches between encryption modes based on message length could optimize the balance between simplicity and security.

Article Publication Details

This article is published in the **International Journal of Engineering Innovations and Technology**, ISSN XXXX-XXXX (Online). In Volume 1 (2026), Issue 1 (January - February)

The journal is published and managed by **RGA Research Publications**.

References

1. Alhassan, M. J; Hassan, A; Sani, S. and Alhassan, Y. (2021). A Combined Technique of an Affine Cipher and Transposition Cipher *Quest Journals Journal of Research in Applied Mathematics Volume 7. Issue 10 (2021) pp: 08-12*
2. Azzam A and Sumarsono (2017), A Modifying of Hill Cipher Algorithm with 3 Substitution Ceaser Cipher. *Proceedings International Conference of Science and Engineering, Indonesia*.**1**: 157-163.
3. Bellovin, S. M. (2011). Frank Miller: Inventor of the One-Time Pad. Cryptologia, 35(3), 203-222.
4. Fahrul I, K., Hassan F, S., Toras P and Rahmat W. (2017), Combination of Ceaser Cipher Modification with Transposition Cipher. *Advances in Science Technology and Engineering Systems Journal*. **2**(5): 22-25.
5. Hassan, A (2024a) Analysis and modification of Vigenere cipher. *Journal of mathematical science and computational mathematics* (JMSCM). **5(4):**502-510
6. Hassan, A and Abdullahi, U (2024b) security analysis and modification of a Ceaser cipher, *journal of mathematical science and computational mathematics* (JMSCM). **5**(4):480-487
7. Hassan, A., Garko, A., Sani, S., Abdullahi, U and Sahalu, S (2023). Combined Techniques of Hill Cipher and Transposition Cipher, *Journal of Mathematics Letters*. **1**(822) pp 1-8, DOI:10.31586/jml.2023.822
8. Hassan, A., Mallam, N. J., Umar, A., Dakingari, A. U., Illo, Z. Z., (2025) a modified Atbash cipher with special characters and stack implementation, *journal of mathematical science and computational mathematics* (JMSCM). **6**(1):44-51
9. Hassan, A; Alhassan, M. J; Alhassan, Y. and Sani, S. (2021). Cryptography as a Solution for a Better Security *International Journal of Advances in Engineering and Management* (IJAEM :**3**(12). pp: 849-853
10. Kahn, D. (1996). The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet. Scribner.
11. Kashish G and Supriya K. (2013) Modified Ceaser Cipher for a Better Security Enhancement. International Journal of Computer Application. **73**:26-31
12. Kuriakkottu A. R. (2021). Use of Transposition Cipher and its Types. International Journal of Research and Engineering, Science and Management **4**(11), 164-165
13. Mishra A. (2013), Enhancing security of Ceaser cipher using different methods. International Journal of Research in Engineering and Technology **2**(09):327-332.
14. Pooja S and Pintu S. (2017), Enhancing security of Ceaser cipher using "Divide and Conquer Approach". International Journal of Advance Research in Science and Engineering. **06**(02):144-150.
15. Shahid B. D. (2014), enhancing the security of Ceaser cipher using double substitution method. International Journal of Computer Science and Engineering Technology. **5**:772-774.
16. Singh, S. (1999). The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. Doubleday.
17. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.
18. Umar, M., Hassan, A., Abdullahi, I & Muhammad Shehu, Z (2024a) Permutation of National Identification Number for a Better Security in Communication Channel. International Journal of Innovative Science and Research Technology ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1655. **9**(4) Pp:3096-3099
19. Umar, M., Hassan, A., Abdullahi, I & Muhammad Shehu, Z (2024b) Transposition Cipher as a Solution for a Better Bank Verification Number (BVN) Security in Communication Channel. International Journal of Innovative Science and Research Technology ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24MAY1654. Volume **9**, Issue 5, pp:3090-3095