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

A POSITION INDEXED AFFINE TRANSFORMATION FAMILY OVER MODULAR RING Z_{26}

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

Received: 04 Jan 2026 | Accepted: 26 Jan 2026 | Published: 29 Jan 2026

Abstract

This paper presents a new modification of the classical Affine Cipher called the Double Position Affine Cipher. The traditional Affine Cipher uses two fixed keys to encrypt every letter in a message. Because the same transformation is applied to all characters, identical letters in different positions produce the same ciphertext. This makes the cipher vulnerable to frequency analysis and brute-force attacks. The proposed method improves security by allowing both keys to change according to the position of each character in the plaintext. In this approach, the encryption formula becomes position-dependent, meaning that the same letter appearing in different locations will usually encrypt to different ciphertext letters. This introduces controlled variability while preserving the mathematical structure of modular arithmetic. The cipher remains reversible as long as the position-based multiplicative key satisfies the necessary invertibility condition in $\mathbb{Z}_{26}$. The method significantly increases the key space and reduces predictable patterns found in monoalphabetic substitution ciphers. This modification maintains simplicity in implementation while providing stronger resistance against classical cryptanalytic attacks. The Double Position Affine Cipher therefore offers an improved and structured alternative to the traditional Affine Cipher for educational and research purposes.

Keywords: cryptography, encryption, decryption, modular arithmetic, Affine cipher.

1. Introduction

Cryptography is the study of methods used to protect information from unauthorized access. Over the years, many encryption techniques have been developed, ranging from simple classical systems to highly advanced modern algorithms (Hassan *et al*., 2023; Hassan *et al*., 2021; Hassan., 2024; Hassan & Abdullahi., 2024; Hassan *et al*., 2025; Kashish & Supriya., 2013; Pooja & Pintu., 2017; Shahid., 2014). One of the well-known classical methods is the Affine Cipher. The Affine Cipher is a substitution cipher that uses a mathematical function to transform each letter of the plaintext into another letter using modular arithmetic. Because of its clear structure and simple formula, it is often introduced in beginner cryptography courses (Fahrul *et al*., 2017; Mishra., 2013; Umar *et al*., 2024). In the classical Affine Cipher, two fixed keys are used for the entire message. This means that every occurrence of the same letter in the plaintext will always produce the same ciphertext letter. Although this design makes the cipher easy to implement and understand, it also makes it vulnerable to frequency analysis and brute force attacks (Alhassan *et al*., 2021). The small key

space further reduces its security strength (Azzam & Sumarsono., 2017; Hassan *et al*., 2023). To address these weaknesses, this research proposes a modified version. The main idea of this improvement is to allow both encryption keys to change depending on the position of each character in the plaintext. Instead of using fixed values, the multiplicative and additive keys vary according to a defined position-based rule. As a result, the same letter appearing in different positions will usually encrypt into different ciphertext letters.

This position-dependent structure transforms the cipher from a simple monoalphabetic substitution system into a more dynamic and structured polyalphabetic scheme. Despite this modification, the cipher still operates within modular arithmetic, which preserves its mathematical clarity and reversibility under proper conditions (Kasturia & Maheswa., 2017; Sharma & Gupta., 2017).

The goal of this study is to present a simple yet stronger alternative to the classical Affine Cipher while keeping the design easy to understand and implement. This makes the proposed method suitable for educational purposes and for researchers interested in structured improvements of classical encryption systems.

2. Methodology

Affine Cipher

The Affine Cipher is a classical encryption method based on substitution and modular arithmetic. In this method, each letter of the plaintext is converted into a number using the rule:

$A = 0, B = 1, C = 2, \dots, Z = 25.$

After converting the letter to a number, a mathematical formula is applied to produce the ciphertext. The encryption formula is:

$E(x) = (ax + b) mod 26$

Here,

x is the numerical value of the plaintext letter,

a and b are the secret keys,

a must satisfy $\gcd(a, 26) = 1$ so that decryption is possible.

To decrypt, we use:

$D(x) = a^{-1}(x - b) mod 26$

Where a^{-1} is the modular inverse of $a \; modulo \; 26$.

The Affine Cipher is simple and easy to understand, which makes it useful for learning the basics of cryptography. However, since it uses fixed keys for the entire message, it is not very secure against modern attacks.

Example 1 (Encryption)

Let:

$a = 5, b = 8$

Encrypt the word: ***CAT***

Step 1: Convert letters to numbers

$C = 2$

$A = 0$

$T = 19$

Step 2: Apply formula

$E(x) = (5x + 8) mod 26$

For C

$E(2) = (5 \times 2 + 8) mod 26$

$= (10 + 8)\ mod\ 26$

$= 18 \rightarrow S$

For A

$E(0) = (5 \times 0 + 8) mod 26$

$= 8 \rightarrow I$

For T

$E(19) = (5 \times 19 + 8) mod 26$

$= (95 + 8) mod 26$

$= 103 mod 26$

$= 25 \rightarrow z$

= (95 + 8) mod 26

= 103 mod 26

= 25 → Z

Ciphertext: ***SIZ***

Example 2 (Decryption)

Let:

$a = 7, b = 3$

Ciphertext: ***XQ***

Step 1: Find modular inverse of $7\ mod\ 26$

$7 \times 15 = 105 \equiv 1\ mod\ 26$

So $a^{-1} = 15$

Step 2: Convert letters to numbers

$X = 23$

$Q = 16$

Step 3: Apply formula

$D(x) = 15(x - 3) mod 26$

For X:

$D(23) = 15(23 - 3) mod 26$

$= 15 \times (20)\ mod\ 26$

$= 300\ mod\ 26$

$= 14 \rightarrow O$

For Q:

$D(16) = 15(16 - 3) mod 26$

$= 15 \times (13) mod\ 26$

$= 195\ mod\ 26$

$= 13 \rightarrow N$

Plaintext: ***ON***

Proposed algorithm

The Double Position Affine Cipher is a modified version of the classical Affine Cipher. In the traditional Affine Cipher, two fixed keys a and b are used to encrypt every letter in a message. Because the keys do not change, the same plaintext letter always produces the same ciphertext letter. This creates patterns that attackers can study and break. This proposed algorithm improves this idea by allowing both keys to depend on the position of the letter in the message. This means the encryption rule changes as we move from one character to another. As a result, the same letter appearing in different positions will usually be encrypted into different letters.

We still use the alphabet rule:

$A = 0, B = 1, C = 2, \ldots, Z = 25.$

The position index starts from $i = 0$.

The encryption formula is:

$E_i(x) = (a_i x + b_i) mod 26$

where

$b_i = b + 2i$

The Decryption formula is given by

$D_i(x) = a_i^{-1}(x - b_i) mod 26$

For decryption to work, each a_i must satisfy $\gcd(a_i, 26) = 1$.

This method increases security because the transformation changes at every position. It remains simple to compute but reduces repeated ciphertext patterns.

Example 3

Let

$a = 5, b = 3$

Encrypt and decrypt: ***HI***

Encryption

Position 0 (H):

H = 7

$a_0 = 5$

$b_0 = 3$

$E_0(7) = (5 \times 7 + 3) mod 26$

$= 38 \, mod \, 26$

$= 12 \rightarrow M$

Position 1 (I):

I = 8

$a_1 = 6$

$b_1 = 5$

$E_0(8) = (6 \times 8 + 5) mod 26$

$= 53 \, mod \, 26$

$= 1 \rightarrow B$

Ciphertext: ***MB***

Decryption

Position 0 (M):

M = 12

Inverse of $5 \, mod \, 26$ is 21

$D_0(12) = 21(12 - 3) \, mod 26$

$= 189 \, mod 26$

$= 7 \rightarrow H$

Position 1 (B):

B = 1

Inverse of $6 \, mod \, 26$ is not available

Since $\gcd(6, 26) = 2,$ then, 6 has no inverse.

So this choice is invalid.

This shows we must always ensure $\boldsymbol{a_i}$ is coprime with 26.

Example 4 (Valid Case)

Let:

$a = 7, b = 4$

Encrypt and decrypt: $\boldsymbol{GO}$

Encryption

Position 0 (G):

G = 6

$a_0 = 7$

$b_0 = 4$

$E_0(6) = (7 \times 6 + 4) mod 26$

$= 20 \rightarrow U$

Position 1 (O):

O = 14

$a_1 = 8\ (invalid\ because \gcd(8, 26) = 2)$

So we adjust rule slightly:

Let

$a_i = 7 + 2i$

Now:

Position 0: $a_0 = 7$

Position 1: $a_1 = 9$ (valid)

Now compute:

$E_1(14) = (9 \times 14 + 6) mod 26$

$= 132\ mod\ 26$

$= 2 \rightarrow C$

Ciphertext: $\boldsymbol{UC}$

Decryption

Position 0:

Inverse of $7\ mod\ 26$ is 15

$= 240 \ mod \ 26$

$= 6 \rightarrow G$

Position 1:

Inverse of $9 \ mod \ 26$ is 3

$= -12 \ mod \ 26$

$= 14 \rightarrow O$

Plaintext: ***GO***

This example shows that the improve version of the Affine cipher works correctly when all position based keys satisfy the invertibility condition.

Implementation

The Double Position Affine Cipher (DPAC) can be implemented easily using a programming language such as Python. The idea is to follow the mathematical formula step by step while making sure that each position uses its own keys. In this improved version, both the multiplicative key and the additive key depend on the character position.

Before implementing the algorithm, we must decide three things:

The alphabet system $(A = 0 \ to \ Z = 25)$.

The base values of a and b.

The position rule for generating a_i and b.

In this work, we use:

$a_i = a + 2i$

$b_i = a + 2i$

The reason for using $2i$ instead of just i is to reduce the chance that a_i shares a common factor with 26. However, we still check that $\gcd(a_i, 26) = 1$ before encryption or decryption. If this condition is not satisfied, the program stops and warns the user.

The implementation process includes:

a. Converting letters to numbers.
b. Applying the position-based formula.
c. Converting numbers back to letters.
d. Computing the modular inverse for decryption.

The structure remains simple, but the output becomes more dynamic because each letter uses different keys depending on its position.

Python implementation algorithm

```
import math

# Function to find modular inverse of a number under mod 26
```

```
def mod_inverse(a, m=26):
    for x in range(1, m):
        if (a * x) % m == 1:
            return x
    return None  # If no inverse exists
#For Encryption function
def encrypt(text, a, b):
    result = ""
    text = text.upper()
    for i, char in enumerate(text):
        if char.isalpha():
            x = ord(char) - ord('A')  # Convert letter to number (0-25)
            # Position-based keys
            a_i = a + 2 * i
            b_i = b + 2 * i
            # Check invertibility condition
            if math.gcd(a_i, 26) != 1:
                raise ValueError(f"a_i = {a_i} is not coprime with 26 at position {i}")
            # Apply encryption formula
            encrypted_value = (a_i * x + b_i) % 26
            # Convert number back to letter
            result += chr(encrypted_value + ord('A'))
        else:
            result += char  # Keep spaces or symbols unchanged
    return result
# For Decryption function
def decrypt(cipher, a, b):
    result = ""
    cipher = cipher.upper()
    for i, char in enumerate(cipher):
        if char.isalpha():
            y = ord(char) - ord('A')
            # Position-based keys
```

```
        a_i = a + 2 * i
        b_i = b + 2 * i
        # Find modular inverse of a_i
        inv = mod_inverse(a_i)
        if inv is None:
        raise ValueError(f"No modular inverse for a_i = {a_i} at position {i}")
        # Apply decryption formula
        decrypted_value = (inv * (y - b_i)) % 26
        result += chr(decrypted_value + ord('A'))
    else:
        result += char
return result
```

3. Results and Analysis

Security Analysis: This Double Position Affine Cipher provides stronger security than the original Affine Cipher because its transformation changes at every character position. In the classical Affine Cipher, the encryption rule is fixed for the entire message. This means that if a letter such as “E” appears many times, it will always be encrypted to the same ciphertext letter. Such repetition creates patterns that can be detected using frequency analysis. Since English and many other languages have predictable letter frequencies, an attacker can study these patterns and recover the keys with relatively low effort. In contrast, the improved cipher uses position-dependent keys. The multiplicative and additive values change as the index increases, so the same plaintext letter can encrypt to different ciphertext letters depending on where it appears. This reduces visible repetition and weakens the effectiveness of simple statistical attacks. Although the cipher still operates within modular arithmetic and remains linear in structure, the changing keys make direct pattern recognition much harder.

When comparing key space, the difference is significant. The original Affine Cipher has only 12 valid values for the multiplicative key (those coprime with 26) and 26 possible values for the additive key, giving a total of 312 possible key pairs. This is small and can be tested quickly using brute force. In this modified version, the keys vary with position. Even if the base values a and b are known, the effective keys used across a message form a sequence rather than a single pair. For longer messages, this produces many more possible key combinations, especially if different position rules are allowed. As a result, the search space becomes larger and the cipher becomes more resistant to straightforward brute force attempts compared to the original Affine Cipher.

Computational Efficiency: The Double Position Affine Cipher remains computationally efficient because it still relies on simple arithmetic operations such as multiplication, addition, subtraction, and modular reduction. The only extra work compared to the classical Affine Cipher is the calculation of position-based keys for each character. This adds a small overhead, but it does not significantly increase processing time. The algorithm runs in linear time, meaning the encryption and decryption time grows proportionally with the length of the message. Since it does not require

complex matrices or heavy computations, it can be implemented easily on basic systems while maintaining reasonable performance.

4. Conclusion and Recommendations

This Double Position Affine Cipher is a structured improvement of the classical Affine Cipher. By allowing both keys to change according to the position of each character, the cipher reduces repeated patterns that are common in simple substitution systems. This makes it stronger than the traditional Affine Cipher, especially against basic frequency analysis and simple brute-force attacks. At the same time, the method keeps the mathematical foundation of modular arithmetic, which makes it easy to understand and implement for students and beginner researchers.

However, even with this improvement, this method is still a classical style cipher. Modern encryption systems such as block ciphers and stream ciphers use more advanced techniques, including multiple rounds of substitution and permutation, large key sizes, and nonlinear transformations. Compared to these modern systems, this method is not designed for high-level security in real-world applications. Instead, its strength lies in education, experimentation, and research development.

It is recommended that future work may extend this idea further by introducing nonlinear position functions, larger character sets, or multi round encryption. Researchers may also study its resistance against more advanced cryptanalysis methods. In conclusion, this work serves as a meaningful bridge between classical cryptography and modern encryption concepts, helping learners understand how simple mathematical modifications can improve security structure.

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. 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.
4. Hassan, A (2024) Analysis and modification of Vigenere cipher. Journal of mathematical science and computational mathematics (JMSCM). 5(4):502-510
5. Hassan, A and Abdullahi, U (2024) security analysis and modification of a Ceaser cipher, journal of mathematical science and computational mathematics (JMSCM). 5(4):480-487
6. 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
7. 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
8. 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
9. Kashish G and Supriya K. (2013) Modified Ceaser Cipher for a Better Security Enhancement. International Journal of Computer Application.73:26-31
10. Kasturia, P and Maheswa, K (2017). Critical Analysis of Various Cryptographic Algorithms. Journal of Ultra Computer & Information Technology. 8(1), pp 5-9
11. Mishra A. (2013), Enhancing security of Ceaser cipher using different methods. International Journal of Research in Engineering and Technology 2(09):327-332.
12. 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.
13. 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.
14. Sharma, S and Gupta, Y (2017). Study on Cryptography and Techniques. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2(1)-ISSN: 2456-3307. pp 249-252
15. Umar, M., Hassan, A., Abdullahi, I & Muhammad Shehu, Z (2024) 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