# A Study of Z-Transform Based Encryption Algorithm

Mohammed N. Alenezi [1*] and Fawaz S. Al-Anzi[2]

[1]Computer Science & Information System Department, Public Authority for Applied Education & Training, Kuwait,
[2]Computer Engineering Department, Kuwait University, Kuwait,
[*]Corresponding author

***Abstract*:** It has become increasingly important to ensure the protection of information, especially data in transit. Therefore, it is the primary goal of any encryption algorithm to safeguard information protection against security attacks. It is equally important to design high-performance solutions with affordable cost of implementation. Encryption algorithms are used to transform plain text to ciphertext to protect privacy, prevent data fraud, and prevent unauthorized access of data in daily transactions. There are multiple types of encryption algorithms, each with its niche tactics to enhance security. This paper contributes an efficient and secure encryption algorithm for information security based on Z-transformation and XOR function known as the Z-Transformation Encryption (ZTE) technique. ZTE converts the encryption from the regular discrete-time domain into a complex frequency domain representation. It implements the concept of Z-transformation and XOR operations at the source. The reverse process is applied at the receiving end of the transaction, wherein the inverse of Z-Transformation and XOR are applied to reveal the original plain text message. The simulation of the proposed algorithm is conducted using the **R** language. The results show a promising performance comparing to other symmetric algorithms.

***Keywords*:** Information security, Encryption, Decryption, Cryptography, Z-transformation, Inverse Z-transformation.

## 1. Introduction

As the Internet nowadays dominates the ways of exchanging information, the need for a secure mechanism that assures the privacy of the data and provides protection for the data against attacks has increased. Cryptography is a Greek word that is divided into two parts. The first part is "crypto," which means "secret," and the second part is "graphy," which means "writing." In other words, Cryptography can be defined as the study of secrets. The main goal of Cryptography is to hide confidential information to provide a high level of privacy. Mainly, there are five goals behind using Cryptography [1], [2]: *Authentication*, *Confidentiality*, *Integrity*, *Non-Repudiation*, and *Availability*. The genuine identity of both the sender and the receiver can be verified by using authentication, while confidentiality assures only the intended parties (sender and receiver) can read the content of the message. Confidentiality protects the information from unauthorized access. Integrity is used to guarantee that the content has not been altered; therefore, the message is protected. Non-repudiation provides a means to prevent both parties from denying the ownership of a sent message. The digital signature is an excellent example of non-repudiation. Many new commercial and real-time systems are designed to be available 24/7 and cannot tolerate disruption. Therefore, the need for a secure system that provides a protection mechanism against service disruption, such as Denial of Service (DoS) attack, becomes a necessity. Availability assures timely access to information by the authorized people. Cryptography is inclusive of encryption and decryption methods, in addition to digital signatures. The process of encryption ensures confidentiality by converting plain text into ciphertext. On the other hand, properties such as authentication, integrity, and non-repudiation can be attributed to digital signatures.

Many Encryption algorithms have been proposed to enhance the encryption process [3]-[9]. Generally, encryption can be categorized into two types: symmetric and asymmetric. The cryptographic process of encrypting and decrypting information is met by utilizing public and private keys. The symmetric key algorithm uses a private pre-shared key between the two ends to encrypt and decrypt. On the other hand, asymmetric encryption uses two keys: private and public keys, to encrypt and decrypt the data. This contribution presented a reliable symmetric key encryption architecture primarily based on Z-Transformation Encryption (ZTE). ZTE converts the encryption from the regular discrete-time domain into a complex frequency domain representation. Using the frequency domain adds an extra layer of security to the encrypted text and acts as an encapsulation form by converting the encrypted text to the frequency domain. For simplicity, we have used XOR to encrypt the text before ZTE encryption. However, ZTE is not limited to a specific encryption algorithm, and any well-known encryption algorithm can be used. This will harden the process for Cryptoanalysis and strengthen the ZTE algorithm.

The rest of the paper is organized as follows. Related work is served in Section 2, while section 3 describes the new proposed cryptographic algorithm. Section 4 presents the performance and analysis of the proposed technique. The conclusion and future work are presented in Section 5 and Section 6, respectively.

## 2. Related Work

In this section, a basic introduction about encryption is presented and followed by related work to existing symmetric encryption. The Internet can be available to most of the world nowadays, and people stay connected to the Internet most of the time. Therefore, the need for a reliable network is highly appreciated, whether by ISPs or clients. The security of the networks is strongly relying on cryptography [10], which explains the high demand of encryption algorithms. With the available powerful machines and the Graphics Processing Unit (GPU) usage, the processing load is not considered a significant obstacle to crack the encryption algorithm from the

hackers' perspective. Therefore, a secure and fast encryption algorithm becomes a necessity. Many different encryption techniques such as DES, 3DES, AES, BlowFish, and others [11] have been proposed to secure the communications and provide the required privacy. Each technique has its advantages and disadvantages.

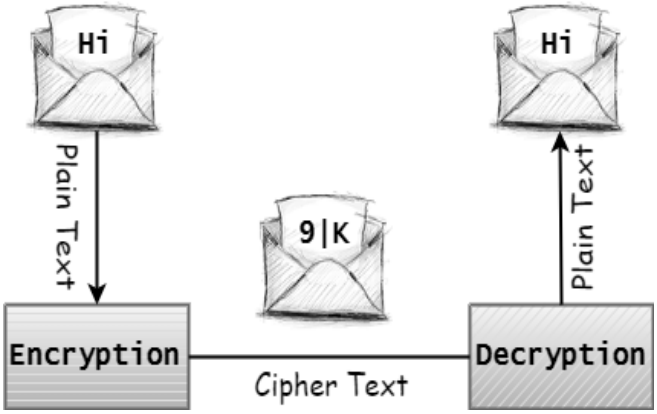


**Figure 1.** Encryption-Decryption process.

### 2.1 Encryption Basics

As mentioned in section 1, Cryptography is the science of hiding secrets and assuring both intended parties can receive the message. In Cryptography, the encryption algorithm is used to hide the information, while the Decryption algorithm is used to reverse the process and reveal the hidden information. Figure 1 shows the general idea of the Encryption and Decryption process. The information before the encryption process is called *plaintext*. Once the output is produced from the encryption process, an unreadable format of the plain text called *ciphertext* is produced. Decryption is used to decipher the text from an unreadable format to the original plaintext [12]. The system, which applies the whole process, starting from encryption and ending by decryption, is called the Cryptographic system. Cryptoanalysis or code-breaking is used to break secure communication and the cryptographic system [2] [13].

The classification of cryptographic algorithms is subjected to many different aspects. It depends on what is the criteria used in classifications. However, the most common classification of cryptographic algorithms is based on a number of keys involved in the process. Cryptographic algorithms can be classified into two categories: *Symmetric* and *Asymmetric* algorithms.

Symmetric encryption, also called *secret* or *private* key encryption, uses the same key for encryption and decryption. As shown in Figure 2, the sender and the receiver should have the same key (secret key) before the encryption process starts. The sender uses the key to encrypt the plaintext, and the receiver uses the same key to decrypt the ciphertext. Symmetric encryption algorithms are generally categorized based on input data handling into *Block Cipher* and *Stream Cipher*. Block cipher essentially encrypts and decrypts one block at a time, while stream cipher encrypts and decrypts one byte at a time. The most common classification for cryptographic algorithms is shown in Figure 3.

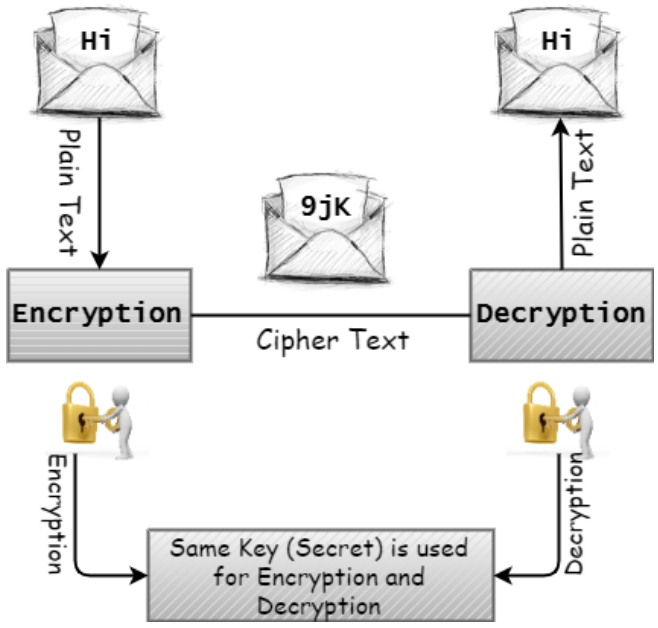


**Figure 2.** Symmetric Algorithms.

### 2.2 Literature Review

Extensive research has been conducted recently in the field of cryptography and the study of data encryption. An efficient symmetric encryption cipher called YC1 was proposed by Panford et al. [14] that makes use of the concept of bit rotation, where key spaces of different lengths are used to encrypt and decrypt the suggested plaintext. The 95-character permutation function, coupled with unlimited key space, makes it a lucrative cipher, as it is exhaustive to opt for brute-forcing. In addition to being susceptible to crypto-analytic attacks, YC1 is challenging to implement on a more considerable amount of text due to its proportionally large execution time. It also seems to observe some weak vital elements, with an inability to find key spaces. With data security as the primary focal point, an algorithm that involves multiple access circular queues (MACQ) was proposed. Primarily, in a circular queue, data operations are performed in a first-out manner. However, like a circle, the last position is linked back to the first.

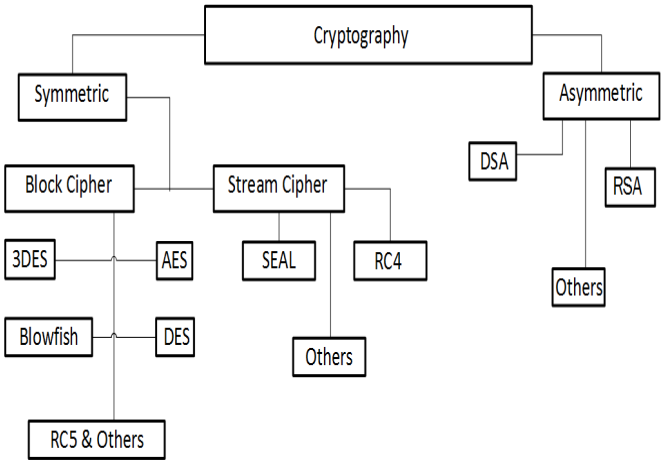


**Figure 3.** Classification of Cryptographic algorithms.

Phull and Som [15] introduced a symmetric block cipher that widely distributed information using MACQs of varied lengths. Despite the keys being generated randomly, elements in the circular queue are subjected to multiple iterations of substitutions and XORing. Therefore, the cipher can be deemed relatively complex, ensuring security, efficiency, and speed.

Muhammad N. et al. [16] proposed an algorithm that studied the effectiveness of Security Increment to Symmetric Data Encryption through AES Methodology. The technique is based on a symmetric key algorithm analogous to Rijndael, using AES to improve the security aspect of the algorithm. The method proposed in this paper seemed to be more secure than Rijndael in terms of the size of the key but comparatively low efficiency, which the authors deemed negligible in the large scale of events.

Mahalakshmi. J and Kuppuswamy [17] used a Symmetric encryption tactic to ensure data security with the storage medium. The proposed algorithm mainly operates as a block cipher, wherein every alphanumeric value is substituted with its corresponding block value, accessed in the form of a matrix. The proposed method excels in terms of data security and is complex enough to withstand brute-forcing. However, the method could be susceptible to side-channel attacks. Also, large-scale real-time applications were in dire need of an apt encryption strategy.

Alsharani and Walker [18] proposed an algorithm that adopted a cubicle technique. The method subjected the plain text to an 8 x 8 x 8 cube matrix, with binary inputs on individual cells. This method is relatively light and affordable. Another advantage is that the technique used to generate keys is quite complicated, adding on the long key size. Although it is deemed to be faster than AES, the number of subjected rounds for keys generation increases the delay. This can be identified as a significant limitation of this algorithm. Therefore, one needs to keep this in mind while designing the algorithm for real-time use.

It seems essential to understand standard but necessary level encryption like Ceasar to establish a benchmark for performance evaluation of the algorithm proposed in this paper. Jain et al. [19] devised a technique to enhance the security of elementary level Ceasar cipher facilitating secure communication. The conventional Ceasar cipher follows a linear shift substitution pattern. Alternatively, the enhanced Ceasar algorithm uses a randomly shifting process in substituting characters in conjunction with permutation box techniques [18]. Ciphertext = ($PlainText * key1$) + $key2$ [19], also called the technique of Affine Ciphers, is used to enable the substitution box. A unique factor to this method is that ASCII and extended ASCII can be encrypted using this method. Additionally, this technique also uses double columnar transposition to improve the strength of the algorithm.

Panhwar et al. [20] discussed various symmetric and asymmetric encryption algorithms used in mobile computing-based workstations. They compared and evaluated various cryptographic algorithms' performance based on encryption and decryption time, speed, memory usage, validity, etc., in mobile computing platforms. Aburass and Qatawneh [21] evaluated the performance of one of the best-known symmetric key encryption algorithms in terms of running time, parallel efficiency, etc. They demonstrated a parallel implementation of AES on a Supercomputer, IMAN1, with the help of a message passing interface (MPI) library. The evaluation of the performance was based on different data sizes and a different number of processors. Based on their study, parallel AES performance increases with the increase in processors from 2 to 16. Buhari et al. [22] performed a performance analysis of two symmetric encryption algorithms, Advanced Encryption Standard (AES) and Blowfish, for various data types: image, video, audio, and text. They evaluated the performance in terms of encryption time and throughput.

A new encryption algorithm based on rotation–translation equation and numerical methods is proposed by Stoyanov and Nedzhibov [23]. They aimed to improve the performance and security of already available encryption algorithms with a faster convergent iterative method and rotation–translation formula. The space contraction equation used in the new algorithm added additional randomness. They performed a security analysis on this new encryption scheme, and it can be successfully applied to information security applications.

Vilardy O. et al. [24] proposed new encryption and decryption algorithms for images based on iterative cosine transform and Jigsaw transform (JT) to prevent attacks such as the differential, brute force, statistical, and entropy. The encrypted images are protected from attacks with the help of three security keys. JT added extra security to the images by adding two keys and performed two random permutations. The use of cosine transform over a finite field helps to decrypt the images at high quality. Ramasamy et al. [25] proposed a new encryption scheme Enhanced Logistic Map (ELM), for images based on chaotic maps and simple encryption techniques. They used encryption techniques like modified zigzag transformation, block scrambling, diffusion, keystream generation, and permutation to overcome the attacks.

Nissar et al. [26] aimed to enhance the security of AES encryption by including Dynamicity in the S-Box of AES for preventing Biclique and Brute Force Attacks. They included a hashing technique to the existing AES for achieving better security with the help of SHA3, SHA5, and MD4. They used novel key dispersion to AES to increase the avalanche effect of the encryption algorithm. Singh et al. [27] performed an analysis of basic and modified working Caesar ciphers' and performance. They evaluate the basic Caesar cipher, XOR Caesar cipher, and Delta formation Caesar cipher in terms of frequency test, Brute force attack, and Avalanche Effect. Caesar ciphers are simple encryption methods and can enhance security by making some advancements to the basic model.

Sirivaram uses Z-Transformation alongside a finite state machine to improve the security of ciphers [28]. Z-Transformation can be described as how a sequence of discrete data is converted to a complex domain [29]. The data sequence in itself may be real or complex. The paper

compares its proposed methodology to one that uses Laplace transformation, which was discovered to lack security. The proposed Z transform method addresses this demerit [30].

Any study of encryption algorithms is considered incomplete without understanding AES [31] [32]. Advances Encryption System or AES has been regarded widely as a means of secure encryption among the many available types of encryption. AES has also proved its mettle in a variety of applications. It can also be considered the go-to set of instructions preferred by CPU manufacturers, resulting in improved performance. It is a form of symmetric encryption that primarily uses the same key for its encryption and decryption process. This method encrypts and decrypts in blocks of 128 bits (block cipher). The plain text is initially split into blocks, post which key expansion takes place. A primary key is used to develop a series of 128-bit round keys using Rijndael's key schedule during the key expansion process. Every round signifies every layer of operation executed on the plain text. The block is subjected to a series of operations that include XOR operations, substitutions, transformations, permutations, etc. The inverse process is applied to convert cipher text to plain text. This text is subjected to multiple rounds of operations. The higher the number of rounds, the more complex the cipher is. This kind of system is considered best to protect data and information at rest as it is faster and requires comparatively less computational power to obtain the result. Therefore, it is maintaining a perfect balance between performance and safety. AES cipher keys can be 128, 192, or 256 bits making it entirely secure against brute force discovery. Even if the key or the plain text message is modified by a single byte, the ciphertext output obtained will be changed entirely. As a result, this algorithm is chosen over most traditional stream cipher schemes. The difficulty level of breaking this encryption is what ensures the protocol is safe to use.

## 3. Z-Transform Encryption Algorithm

This work proposes a new reliable and efficient encryption algorithm based on Z-transform and XOR function for ensuring better security. The proposed method enhances both encryption speed and security. This work also evaluates and compares the proposed model's performance with AES-16, AES-24, AES-32, and Caesar cipher.

In this section, a brief introduction about the Z- transformation concept is presented, along with our proposed technique ***ZTE***.

### 3.1 Z-transformation Overview

Z Transform is the mathematical building block used to design, analyze, and monitor any system. Z transform provides a systematic method of obtaining exact solutions of linear difference equations and is directly analogous to the Laplace technique applied to differential equations [33]. The z-transform of a sampled sequence or $X(n)$, where $n$ represents nonnegative integers, and $T$ is the sampling period, is defined by

$$X(z) = Z[x(n)] = \sum_{n=0}^{\infty} x(n)z^{-n} \quad (1)$$

Where $z$ is a complex variable and $X(z)$ is a new representation of $X(n)$. If you want to return to the original sequence, you have to find a coefficient associated with the $n^{th}$ power known as inverse Z transform.

$$X[nT] = \mathcal{Z}^{-1}[X(z)] = \frac{1}{2\pi j}\oint X(z)z^{n-1}dz \quad (2)$$

### 3.2 ZTE illustration

Due to the size limitation and illustration in this example, it is assumed that a plain text with 16 bits (2 bytes) long and a key with 8 bits long.

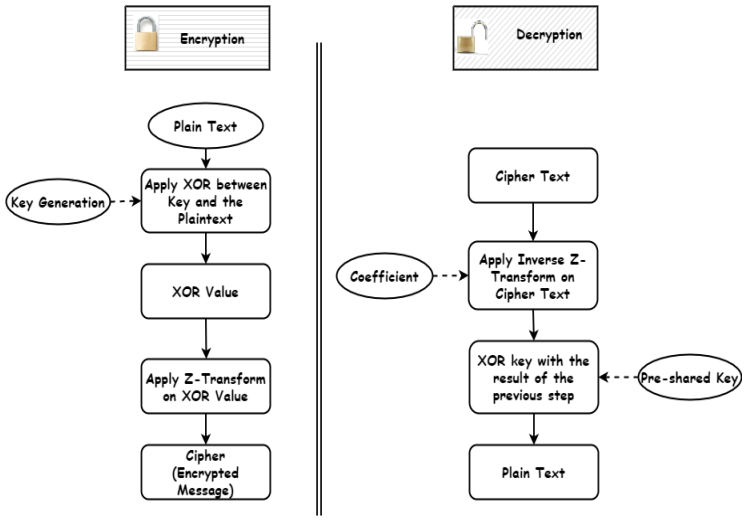


**Figure 4.** Encryption and Decryption process of ZTE algorithm.

Plain Text:

| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Key:

| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|

We will reverse these key digits, make another 8-digit key, and then combine eight bits and get the 16-bit key.

Reversed Key:

| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|

Now combine both eight bits to make 16 bits key.

Key combined (16 bits):

| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

XOR operation is performed: message $\oplus$ key.

XOR operation:

| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Now applying Z-Transform on the above XOR value using equation (1):

$$X(z) = Z[x(n)] = \sum_{n=0}^{\infty} x(n)z^{-n}, \text{ n from 0 to 15}$$
$$= \sum_{n=0}^{n=15} x(n)z^{-n}$$

By substituting the XOR value and using Z as 1:

$$X(z) = 0Z^{-0} + 1Z^{-1} + 0Z^{-2} + .. + 0Z^{-15}$$
$$X(z) = 1Z^{-1} + 1Z^{-3} + 1Z^{-8}$$
$$ChiperText = 111$$

At the receiving end, the coefficient associated with the $n^{th}$ power of $Z^{-n}$ is calculated. The resulted value from the inverse Z-Transform is XOR with the pre-shared key to get an original message.

Figure 4 illustrates the flow chart of the ZTE algorithm encryption and decryption process. In summary, to encrypt, the generated key and plain text are subjected to XOR value. Z transform is applied to the XORed output. Alternatively, the decryption process involves applying inverse Z transform to the ciphertext using the coefficient calculated. The output is XORed using the pre-shared key to get the original plain text message. The entire algorithm is presented in Algorithm 1.

**Algorithm 1.** ZTE Encryption/Decryption process

1 **Encryption mode:**

**Input :** Plain Text;

**Output:** Cipher(encrypted message);

2 **Decryption mode:**

**Input :** Cipher (encrypted message);

**Output:** Plain Text;

3 $P \leftarrow$ plain text;

4 $K_1 \leftarrow$ first part of the Key;

5 $K_2 \leftarrow$ second part of the Key (Reversed part);

6 $Z_{Transform} \leftarrow \sum_{n=0}^{\infty} x(n)z^{-n}$;

7 $Z \leftarrow$ the constant value used in the Z-transformation;

8 **if** Encryption mode, **then**

a $K_{full} = K_1 + K_2$;

b $XOR_{operation} = P \oplus K_{full}$

c $Cipher = \sum_{n=0}^{n=15} x(n)z^{-n}$

9 **else**

//Decryption mode

a Apply the Z- Inverse

b $X[nT] = Z^{-1}[X(z)] = \frac{1}{2\pi j}\oint X(z)z^{n-1}dz$

10 **end**

## 4. Performance and Analysis

### 4.1 Simulation and System Setup

In order to test the ZTE-algorithm performance, the R programming language was used to implement the ZTE-algorithm and other related algorithms. R is a platform-independent programming language and an ideal environment for statistical and graphics computing [34]. An essential key metric in evaluating any encryption algorithm is the encryption time. The ZTE Algorithm consists of three parts. The first part is the 16 bytes key generation, and the second part is the XOR process between the message and the key. The last part is applying the Z-transform to the output of the second part. The same process is reversed for the decryption process. In order to have a fair comparison and avoid platform differences, it was decided to use the same parameters for all experiments.

Moreover, due to the changes in the results for each simulation run, the execution time for each encryption algorithm is repeated 100 times. The presented results are based on the average of 100 runs. The following components were used in the simulation process:

- Programming language: R.
- Application platform: R-Studio Version 1.0.143
- Operating system: Windows 10, 64-bit.
- Hardware computer: 2.6 GHz 6-core, Intel Core i7 MacBook Pro, and 16.00 GB for RAM.

### 4.2 Results & Discussion

The main focus of the experiment was the execution time of the encryption algorithm. It can be deemed that, for the most part, encryption and decryption times are the same [35]. The execution time of different encryption algorithms is shown in figure 5.

The ZTE algorithm has been compared with the classic Caesar algorithm and the approved AES in the simulation. Although the Caesar algorithm is not recommended to be used as an encryption algorithm due to its weaknesses, it was implemented and compared with the ZTE algorithm to give an impression and establish a baseline for the ZTE performance. On the other hand, AES, a.k.a Rijndael was implemented as it is the currently approved replacement algorithm of DES by

the US- National Institute of Standards and Technology (NIST) due to the flaws in DES [32]. AES was simulated with different key sizes (16 bytes, 24 bytes, 32 bytes) and Electronic Code Book (ECB) as a mode. Table 1 presents the encryption speed and throughput of various encryption algorithms under consideration along with the ZTE encryption.

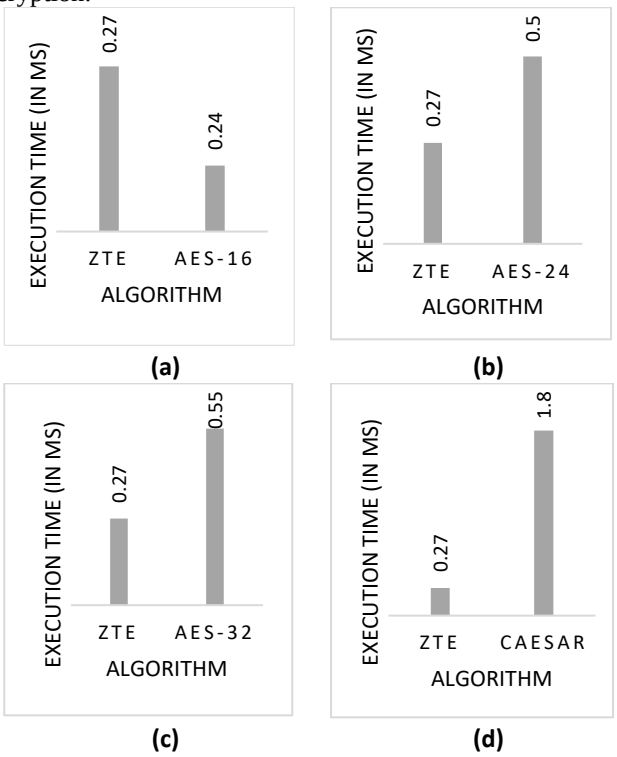


**Figure 5.** Execution Time of different encryption algorithms

**Table 1.** Execution time (in Milliseconds) of various Encryption algorithms

| Encryption Algorithm | Execution Time (in ms) | Throughput (in MB/Sec) |
|---|---|---|
| AES-16 | 0.24 | 2083.33 |
| AES-24 | 0.5 | 1000 |
| AES-32 | 0.55 | 909.09 |
| Caesar | 1.8 | 277.78 |
| ZTE | 0.27 | 1851.85 |

As shown in figures 5 and 6, the ZTE encryption algorithm shows a good and relatively fast encryption time and good throughput compared to AES and Caesar. The results show a close performance to AES (16 bytes) with a difference of 0.03 ms. This difference is acceptable as ZTE is performing the encryption process after the XOR process overhead. On the other hand, ZTE shows a better performance than AES (24 bytes), AES (32 bytes), and Caesar, with a difference of 0.23 ms, 28 ms, and 1.53 ms, respectively. We believe that AES (24 bytes) and AES (32 bytes) show a slower performance due to the key size difference.

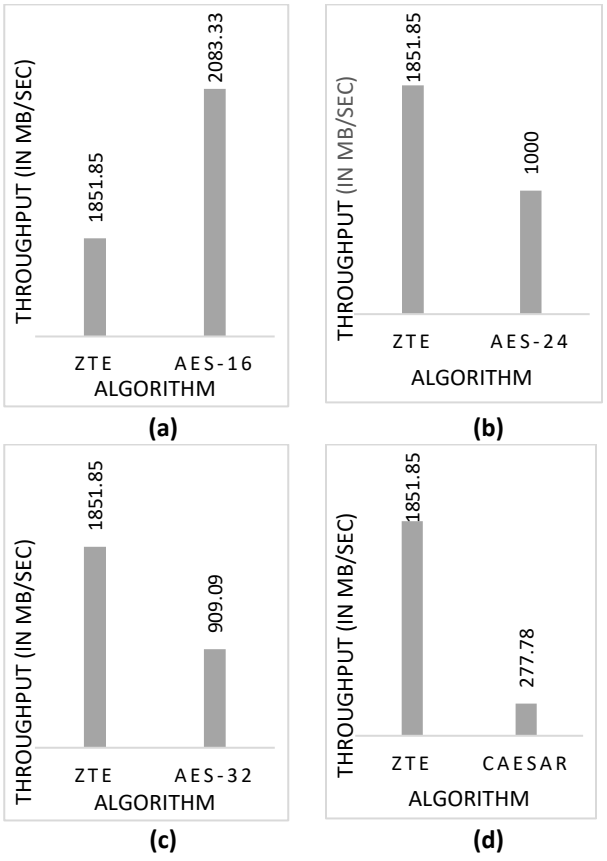


**Figure 6.** Throughput of different encryption algorithms

### 4.3 Z-Value

As mentioned in Section 3, Z is the constant value used in the Z-transformation that needs to be set during the simulation. Therefore, to choose the suitable value of Z for the simulation, we have studied a small range of Z values. Figure 7 shows the performance of ZTE for different Z values. When Z=1, the ZTE execution time is the smallest, i.e., fast encryption and better performance. However, it should be noted that there is no clear pattern to the Z values included in the range. Table 2 lists the time taken to encrypt the plaintext using proposed ZTE encryption with various Z values. There is a need for further evaluation on how the Z value is associated with execution time. This could be attributed to the fact that for this experiment, only discrete values were utilized. Z value could be a decimal number, which increases the size of the range. This construct could be studied further. Unfortunately, understanding this correlation and optimizing the Z value is out of the scope of this paper.

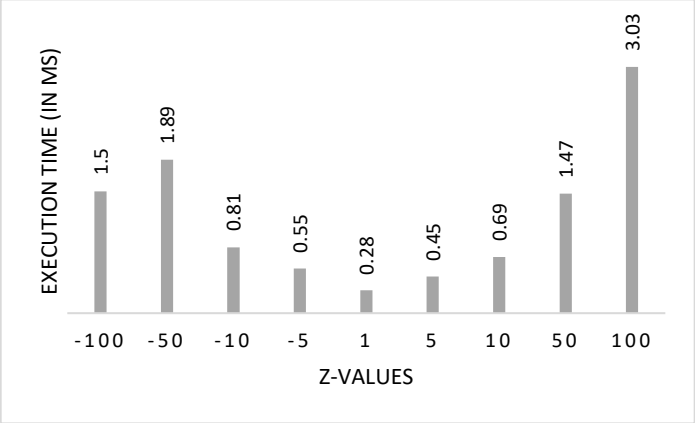


**Figure 7.** Execution Time of ZTE with different Z values.

**Table 2.** Execution time in Milliseconds of ZTE encryption using different Z-values

| Z-Values | Time (in ms) |
|---|---|
| -100 | 1.5 |
| -50 | 1.89 |
| -10 | 0.81 |
| -5 | 0.55 |
| 1 | 0.28 |
| 5 | 0.45 |
| 10 | 0.69 |
| 50 | 1.47 |
| 100 | 3.03 |

## 5. Conclusion and Future Work

With the existence of easy to run attacking tools on the Internet, the need for a secure mechanism that assures the privacy and protection of the data has increased. Encryption has been used over history to exchange information securely. The paper proposes a new secure, efficient, and reliable encryption algorithm. The algorithm applies XOR and Z-Transformation at the source. Alternatively, the ciphertext is converted to plaintext at the destination by reversing the process using inverse Z-transformation and applying the XOR operation. The proposed cipher's security is far enhanced with the use of Z transformation, which makes it quite difficult to brute force. Therein, improved security in optical encryption, digital signal processing, and even multiple image processing. The simulation of the proposed algorithm ZTE was conducted using R-language, and the results showed that the ZTE outperformed (in execution time) the other encryption algorithms such as AES with different key sizes, as it is the currently approved algorithm by US NIST.

In the future, it would be beneficial to conduct a comprehensive comparison of the proposed algorithm with its contemporaries that are commonly being used in standard applications. It would also aid in understanding how the algorithm stands against standard crypto-analysis tactics. Studying the Z value thoroughly as an optimization problem would improve the performance of ZTE significantly.

## References

[1] P. Krzyzanowski, "Cryptographic communication and authentication," *Rutgers University–CS*, vol. 417, pp. 1–2, 1997.

[2] M. N. Alenezi, H. Alabdulrazzaq, and N. Q. Mohammad, "Symmetric encryption algorithms: Review and evaluation study," *International Journal of Communication Networks and Information Security*, vol. 12, no. 2, pp. 256–272, 2020.

[3] K. Sharma and N. Bahl, "Taxonomy of cryptography techniques for network security," *International Journal of Engineering and Computer Science*, vol. 5, no. 8, pp. 17787–17793, 2016.

[4] M. Agrawal and P. Mishra, "A comparative survey on symmetric key encryption techniques," *Int. Journal of Computer Science Eng.*, vol. 4, 05 2012.

[5] M. Bellare, A. Desai, E. Jokipii, and P. Rogaway, "A concrete security treatment of symmetric encryption," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*, Oct 1997, pp. 394–403.

[6] H. AL-Hashimi and W. Hussein, "Implementation of symmetric encryption algorithms," *Computer Engineering and Intelligent Systems*, vol. 8, p. 6, 05 2017.

[7] D. Salama, H. M. A. Kader, and M. M. Hadhoud, "Performance evaluation of symmetric encryption algorithms," in *International Journal of Communications of the IBIMA*, Vol. 10, Issue 3, 2009.

[8] M. E. Haque, S. Zobaed, M. U. Islam, and F. M. Areef, "Performance analysis of cryptographic algorithms for selecting better utilization on resource constraint devices," in *2018 21st International Conference of Computer and Information Technology (ICCIT)*, 2018, pp. 1–6.

[9] H. A. Kholidy, "Towards a scalable symmetric key cryptographic scheme: Performance evaluation and security analysis," in *2019 2nd International Conference on Computer Applications Information Security (ICCAIS)*, 2019, pp. 1–6.

[10] W. Stallings, *Data and computer communications*. Pearson Education India, 2007.

[11] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: theory and implementation," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 79, 2018.

[12] R. Shirey, "Internet security glossary," RFC-2828, Tech. Rep., May 2000. [Online]. Available: https://www.ietf.org/rfc/rfc2828.txt

[13] R. Bhanot and R. Hans, "A review and comparative analysis of various encryption algorithms," *International Journal of Security and Its Applications*, vol. 9, no. 4, pp. 289–306, 2015.

[14] J. Panford, P. Yeng, J. Hayfron-Acquah, and F. Twum, "An efficient symmetric cipher algorithm for data encryption," *International Research Journal of Engineering and Technology*, vol. 3, pp. 1713 – 1732, 05 2016.

[15] S. Phull and S. Som, "Symmetric cryptography using multiple access circular queues (macq)," in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies*, 03 2016, pp. 1–6.

[16] M. N. Islam, M. M. H. Mia, M. F. Chowdhury, and M. A. Matin, "Effect of security increment to symmetric data encryption through aes methodology," in *2008 Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*. IEEE, 2008, pp. 291–294.

[17] K. Kuppusamy and D. Jeyabalu, "A block cipher based cryptographic algorithm to enhance the data security," in *International Journal of Computing and Technology*, vol. 10, 05 2015.

[18] A. Alshahrani and S. Walker, "New approach in symmetric block cipher security using a new cubical technique," *International Journal of Computer Science and Information Technology*, vol. 7, pp. 69–75, 02 2015.

[19] A. Jain, R. Dedhia, and A. Patil, "Enhancing the security of caesar cipher substitution method using a randomized approach for more secure communication," *International Journal of Computer Applications*, vol. 129, pp. 6–11, 11 2015.

[20] M. Panhwar, S. Ali Khuhro, G. Panhwar, and K. Ali, "Saca: A study of symmetric and asymmetric cryptographic algorithms," *IJCSNS: International Journal of Computer Science and Network Security(2019)*, vol. 19, no. 1, 01 2019.

[21] S. Aburass and M. Qatawneh, "Performance evaluation of aes algorithm on supercomputer iman1," *International Journal of Computer Applications*, vol. 179, no. 48, pp. 32–34, 06 2018.

[22] B. Buhari, A. Obiniyi, K. Sunday, and S. Shehu, "Performance evaluation of symmetric data encryption algorithms: Aes and blowfish," *Saudi Journal of Engineering and Technology*, vol. 4, pp. 407 – 414, 10 2019.

[23] B. Stoyanov and G. Nedzhibov, "Symmetric key encryption based on rotation-translation equation," *Symmetry*, vol. 12, p. 73, 01 2020.

[24] J. Vilardy O., L. Barba J., and C. O. Torres Moreno, "Image encryption and decryption systems using the jigsaw transform and the iterative finite field cosine transform," *Photonics*, vol. 6, p. 121, 11 2019.

[25] P. Ramasamy, V. Ranganathan, S. Kadry, R. Damasevicius, and T. Blazauskas, "An image encryption scheme based on block scrambling, modified zigzag transformation and key generation using enhanced logistic-tent map," *Entropy 2019*, vol. 21, 07 2019.

[26] G. Nissar, D. Garg, and B. Khan, "Implementation of security enhancement in aes by inducting dynamicity in aes s-box," *International Journal of Innovative Technology and Exploring Engineering (IJITEE-2019)*, vol. 8, 09 2019.

[27] R. Singh, N. Kumar, and Anuja, "A review paper on cryptography of modified caesar cipher," 08 2018.

[28] S. Sirivaram, "Information security using z transforms and finite state machine," *Research Journal of Science and Technology*, vol. 9, 12 2017.

[29] M. K. Soni, "Implementation of laplace transform in various science and engineering field and relation of laplace transform with z-transform," *Journal of Experimental & Applied Mechanics*, vol. 10, no. 2, pp. 36– 40, 2019.

[30] F. Al-Anzi, M. Al-Enezi, and J. Soni, "New proposed z-transform based encryption algorithm," in *New Proposed Z-Transform Based Encryption Algorithm*, 09 2016, pp. 1–5.

[31] F. J. D'souza and D. Panchal, "Advanced encryption standard (aes) security enhancement using hybrid approach," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017, pp. 647–652.

[32] N.-F. Standard, "Announcing the advanced encryption standard AES," *Federal Information Processing Standards Publication*, vol. 197, no. 151, pp. 3–3, 2001.

[33] S. Elaydi, "Z-transform," *Encyclopedia of Mathematics.*, 2014. [Online]. Available: https://www.encyclopediaofmath.org/index.php/Z-transform

[34] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2019. [Online]. Available: https://www.R-project.org/

[35] A. T. Abdel-Karim, "Performance analysis of data encryption algorithms," URL:" http://www.cse.wustl.edu/~jain/cse567-06/ encryption perf.htm", 2006, [Online; accessed on January 19 January, 2020].