International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

TEXT ENCRYPTION AND DECRYPTION USING THE LOGISTIC FUNCTION: MKTEXT-6

[1] Mohammed Abdulhameed Jassim, [2] Mohammed Alsaeedi, [3] Abbas Jadi, [4] Hasan Abdulhussein Ahmed

Workplace for all:- University of Kufa, Faculty of Basic Education, Department of Mathematics

E − mail [1]:- mohammeda.alkufi@uokufa.edu.iq
E − mail [2]:- mohammedk.alsaeedi@uokufa.edu.iq
E − mail [3]:- abbasa.jedi@uokufa.edu.iq
E − mail [4]:- hasana.alshishtarli@uokufa.edu.iq

Abstract

We continue to develop our algorithms, initially designed for image encryption, to ensure they are also valid for text encryption. This is a challenging task because text encoding does not tolerate any errors, unlike image encryption, where minor numerical inaccuracies may be acceptable. The numerical values of text components must be retrieved accurately. In this study, we build upon our previous algorithm (MK-11), which used Generalized Singular Value Decomposition (GSVD) technology combined with a logistic function for image encryption with a real-number encryption key. We introduce a new algorithm that relies solely on the logistic function, employing a method of dispersal and retrieval of numerical values to encrypt text effectively.

We assess the quality of this new algorithm by applying it to various text samples, as we did with our previous algorithm (MKTEXT-5). Our algorithm operates in two main steps: First, we convert the text to be encrypted into a vector of numerical values, using the ASCII values corresponding to the letters and symbols stored in the computer. Second, we disperse the numerical values of this vector using the encryption key (a real number) and the logistic function.

The results we obtained are excellent and error-free, outperforming our previous algorithms in terms of quality, speed, and accuracy. We reaffirm that these algorithms are suitable for use in various fields, such as information security in banks, universities, and public and private institutions. We will detail the algorithm, results, and related readings in the subsequent sections of our research.

Keywords: text encryption, logistic function, algorithm development, information security, numerical accuracy

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1. Introduction

The field of cryptography has evolved significantly in recent years, driven by advancements in information technology, computational power, and mathematical techniques. Previously, cryptographic methods were largely bas678

ed on classical templates that were relatively limited in scope and application. However, the rapid development in these areas has propelled cryptography into a new era, making it an integral part of modern digital information systems and a key component of matrix science.

The competition among specialists in this field has intensified, leading to continuous innovation and improvement in cryptographic techniques. As we reflect on the historical methods used in cryptography, it is important to note the three primary types of codes and symbolic writing systems that were prevalent in the past:

- **Modern Encryption Algorithms:**

- **Vestal Method**: Originally designed for texts of even length, this block cipher method was improved by Nizar Khalaf Hussein Al-Dakhil to handle odd-length texts.
- **Decipherable Text**: Used in educational programs, especially for teaching phonetics. It helps new readers decrypt words using phonemic skills. For example, children can decrypt "Pat the fat rat" if taught the phoneme combinations of the letters.

- **Encryption Goals:**

1. **Confidentiality**: Ensures that information is accessible only to authorized individuals.
2. **Integrity**: Prevents unauthorized addition or modification of data.
3. **Authentication**: Verifies the identity of the data's originator.

- **Logistic Function:**

We previously discussed the logistic function in our research titled "Image Encryption with General Singular Value Decomposition via Logistic Function by Encryption Key (Real Number) (Method-2)." It is a simple chaotic function, studied since the 1960s, known for its sensitivity to initial values, making it suitable for encryption. Its mathematical representation is:

$$x_{p+1} = tx_p(1 - x_p) \quad X_p \in (0,1) \qquad \text{and } t > 0$$

We have provided models for the graphics of this function, such as the logistic model of population growth. Additionally, the logistic map has several dimensions. In this algorithm, we will focus on the one-dimensional logistic map, which is a one-dimensional representation of a discrete and simple non-linear system. Its mathematical model is:

$X_{p+1} = R(t, X_p) = t * X_p * (1 - X_p)$ where t is parameter and $X_p \in [0,1]$.

2. Methodology for Text Encryption by Logistic Function

We have provided a useful summary of the logistic function. For more details, readers can refer to the research and sources mentioned earlier. Now, we will explain the algorithm with

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

a numerical example to demonstrate its strength and accuracy. Suppose we have a specific text that we wish to encrypt into a vector of numerical values.

Algorithm Steps:

1. **Initialization**: Choose an initial value *x*0 and a growth rate *r*.
2. **Text Conversion**: Convert the text into a vector of numerical values (e.g., ASCII values).
3. **Encryption Process**: Apply the logistic function iteratively to generate a sequence of values and use these to encrypt the numerical vector.
4. **Output**: The encrypted text is represented as a vector of numerical values.

Example:

First step: Encryption:-

Let B_1 be the text vector we want to encryption.

$B_1 = \text{TEXT}_{(\text{befor encryption})} = [\text{mohammed}]$

$B_2 = \text{length}\ (B_1)$

$B_3 = \text{double}(B_1) = [109\ \ 111\ \ 104\ \ 97\ \ 109\ \ 109\ \ 101\ \ 100]$

$B_{31} = [B_3\ \ 10 * \max(B_3)] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (*)$

$= [109 \quad 111 \quad 104 \quad 97 \quad 109 \quad 109 \quad 101 \quad 100 \quad 1110]$

$B_4 = \max(B_3) = 1110$

$$B_5 = \frac{B_{31}}{B_4 + 1}$$

$= [0.0981 \quad 0.0999 \quad 0.0936 \quad 0.0873 \quad 0.0981 \quad 0.0981 \quad 0.0909 \quad 0.0900 \quad 0.9991]$

$B_6 = \text{key} = 52014$

$$B_{61} = \frac{|B_6| - 44}{|B_6|} * 44 = 43.9628$$

$B_7 = \max(B_5) = 0.9991$

$$B_8 = B_5 > \frac{B_7}{2} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$$

$B_{10} = B_{61} * B_5.*(1 - B_5)$

$= [3.8900 \quad 3.9535 \quad 3.7301 \quad 3.5032 \quad 3.8900 \quad 3.8900 \quad 3.6333 \quad 3.6009 \quad 0.0395]$

Now we save the index B_8through the numerical values of the vector blade first B_{10}.

$B_{11} = ((\text{floor}(B_{10}) * 10) + B_8) + (B_{10} - \text{floor}(B_{10}))$

$= [30.8900 \quad 30.9535 \quad 30.7301 \quad 30.5032 \quad 30.8900 \quad 30.8900 \quad 30.6333 \quad 30.6009 \quad 1.0395]$

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Now we keep the upper value $B_7$from the vector of the numerical values confined between zero and one as an additional column on the vector B_{11}.

$B_{12}[B_{11}\ B_7]$

$= [30.8900\ \ 30.9535\ \ 30.7301\ \ 30.5032\ \ 30.8900\ \ 30.8900\ \ 30.6333\ \ 30.6009\ \ \ 1.0395\ \ \ 0.9991]$

The vector B_{12} represents the encryption of the text vectorB_1.

Note / in the step (*) above, we added a new value to the vector values of the numerical values of the text components. Which is equal to ten times the organic value of the components of the numerical values vector, because we are dealing with a function of the second degree in encryption and this function is difficult for us to retrieve the correct numerical value during a process decryption because the roots of the equation are of the second order (two). Therefore, we have to define a flag based on it to create a guide vector that indicates the correct numerical value from among the roots of the quadratic equation, as we have noted through the details of this algorithm.

Second step: Decryption:-

We have the ciphered text vector to numerical values B_{12}

First, we extract the highest value from the vector of the numerical values of the original text, which is between zero and one, whose name wasB_7, and here we will call it B_{13}.

$B_{13} = B_{12}(\text{end}) = 0.9991$

$B_{12}(\text{end}) = [\ ]$

$B_{14} = B_{12}$

$= [30.8900\ \ 30.9535\ \ 30.7301\ \ 30.5032\ \ 30.8900\ \ 30.8900\ \ 30.6333\ \ 30.6009\ \ \ 1.0395]$

Now we extract the directory vector which we called $B_8$and in this step, we will call it B_{15}.

$$B_{15} = 10 * \left(\frac{\text{floor}\ (B_{14})}{10} - \text{floor}\frac{\text{floor}B_{14}}{10}\right) = [0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 0\quad 1]$$

Now we retrieve the ciphered text vector after extracting both B_{13} and B_{15}

$$B_{16} = \text{floor}\frac{\text{floor}B_{14}}{10} + (B_{14} - \text{floor}B_{14})$$

$= [3.8900\ \ \ 3.9535\ \ \ 3.7301\ \ \ 3.5032\ \ \ 3.8900\ \ \ 3.8900\ \ \ 3.6333\ \ \ 3.6009\ \ \ 0.0395]$

$B_{17} = \text{length}(B_{16}) = 9$

For each value from the vector B16, let B16 (i) generate a second-degree equation as follows: -

$$x^2 - x + \frac{B_{16}(i)}{B_{61}} = 0 \ldots\ldots B_{61} \text{ is incryption key}$$

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

This equation is written in the language of Matlab in the form of a vector that includes the transactions only so that its roots are calculated as follows: -

$$B_{18} = \left[1 - 1\ \frac{B_{16}(i)}{B_{61}}\right]$$

$$\text{roots}(B_{18})$$

For vector B16, the generalization of the work on all its components is as follows:

$$B_{18} = \left[\left(1 - 1\ \frac{3.89}{B_{61}}\right)\left(1 - 1\ \frac{3.9535}{B_{61}}\right)\left(1 - 1\ \frac{3.7301}{B_{61}}\right)\left(1 - 1\ \frac{3.5032}{B_{61}}\right)\left(1 - 1\ \frac{3.89}{B_{61}}\right)\left(1 - 1\ \frac{3.89}{B_{61}}\right)\left(1 - 1\ \frac{3.6333}{B_{61}}\right)\left(1 - 1\ \frac{3.6009}{B_{61}}\right)\left(1 - 1\ \frac{0.0395}{B_{61}}\right)\right]$$

Then we calculate the roots of each equation from equations B18 by instructing the roots.

$$r_1 = \text{roots}(B_{18})$$

The result will be two roots for each equation, meaning that the results can be represented as a vector of arranged pairs, each pair represents the radical of one of the equations B18, and we called this vector r1.

$$r_1 =$$

$$\begin{bmatrix} r_1(1)(0.9019 \quad 0.0981), r_1(2)(0.9001 \quad 0.0999), r_1(3)(0.9064 \quad 0.0936), r_1(4)(0.9127 \quad 0.0873), r_1(5)(0.9 \\ r_1(6)(0.9019 \quad 0.0981), r_1(7)(0.9091 \quad 0.0909), r_1(8)(0.91 \quad 0.09), r_1(9)(0.9991 \quad 0.0009) \end{bmatrix}$$

Note that we obtained two numerical values for each letter from letters of the text. So how do we know the correct value in order to complete the decryption*?*

We have a flag (B7/ 2) that we used during the encryption stage above and B7 is the largest numerical value of the vector of the numerical values of the text after converting it within the period (0,1)

$$\text{flag} = \frac{B_7}{2} = 0.4995$$

We also have a index vector is B15 where: -

$$B_{15} = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$$

We will explain the relationship between r1, flag and B15 through the detailed explanatory table below: -

Table 1: The relationship between (r1, flag, and B15).

No.	r_1	B_{15}	$B_{19} >$ or $\leq$ flag(0.4995)	B_{19}
1	$r_1(1)(0.9019 \quad 0.0981)$	0	$B_{19} \leq$ flag(0.4995)	0.0981
2	$r_1(2)(0.9001 \quad 0.0999)$	0	$B_{19} \leq$ flag(0.4995)	0.0999

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3	$r_1^{(3)}(0.9064 \quad 0.0936)$	0	$B_{19} \leq \text{flag}(0.4995)$	0.0936
4	$r_1^{(4)}(0.9127 \quad 0.0873)$	0	$B_{19} \leq \text{flag}(0.4995)$	0.0873
5	$r_1^{(5)}(0.9019 \quad 0.0981)$	0	$B_{19} \leq \text{flag}(0.4995)$	0.0981
6	$r_1^{(6)}(0.9019 \quad 0.0981)$	0	$B_{19} \leq \text{flag}(0.4995)$	0.0981
7	$r_1^{(7)}(0.9091 \quad 0.0909)$	0	$B_{19} \leq \text{flag}(0.4995)$	0.0909
8	$r_1^{(8)}(0.91 \quad 0.09)$	0	$B_{19} \leq \text{flag}(0.4995)$	0.09
9	$r_1^{(9)}(0.9991 \quad 0.0009)$	1	$B_{19} > \text{flag}(0.4995)$	0.9991

$$\therefore B_{19} = [0.0981 \quad 0.0999 \quad 0.0936 \quad 0.0873 \quad 0.0981 \quad 0.0981 \quad 0.0909 \quad 0.0900 \quad 0.9991]$$

B19 the vector represents the numerical values of the text components after the decryption, but it is transformed linearly, making it within the period (0,1).

Now we retrieve the original numerical values after decryption, taking advantage of the largest value in it B4: -

$$B_{20} = B_{19} * (B_4 + 1)$$
$$= [1.0e + 03 * (0.1090 \quad 0.1110 \quad 0.1040 \quad 0.0970 \quad 0.1090 \quad 0.1090 \quad 0.1010 \quad 0.1000 \quad 1.1100)]$$

The vector B20 should contain the numerical values of the text components after decryption, which must match the original text's numerical values in B3. However, through extensive testing, we found that the values in B20 do not exactly match those in B3, resulting in errors. Fortunately, the absolute error value is less than 0.5, allowing us to recover the original numerical values by rounding the numbers in B20 to the nearest integer using a complementary subprogram in MATLAB.

if $B_{20} - \text{floor}(B_{20}) \geq 0.5$

$B_{22} = \text{floor}(B_{20}) + 1$

else

$B_{22} = \text{floor}(B_{20})$

end

Then we clear the last value of B_{22} which we added in step (*) within the encryption phase.

$B_{22}(\text{end}) = [\ ]$

$B_{22} = [109 \quad 111 \quad 104 \quad 97 \quad 109 \quad 109 \quad 101 \quad 100]$

After restoring the numerical values of the components of the original text without error, we can retrieve the text after decryption.

$B_{23} = \text{char}(B_{22}) = \{\text{mohammed}\} = \text{TEXT}_{(\text{after decryption})}$

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2. Results.

We will explain the results through detailed tables.

Table 2: - Application of the algorithm to English text and Arabic text.

Language	text	Numerical values for the text	Encryption key	The encryption text	Numerical values for the text after decryption	Text after decryption	Encryption time Sec.	Decryption time Sec.
English	M o h a m m e d	77 111 104 97 109 109 101 100	23015	20.8327 30.9493 30.7261 30.4995 30.8859 30.8859 30.6294 30.5970 1.0395 0.9991	77 111 104 97 109 109 101 100	M o h a m m e d	0.0100	0.2520
Arabic	م ح م د ك و ف ي	1605 1581 1605 1583 32 1603 1608 1601 1610	10024	30.9315 30.8791 30.9315 30.8835 0.0869 30.9272 30.9380 30.9228 30.9424 1.0027 0.9999	1605 1581 1605 1583 32 1603 1608 1601 1610	م ح م د ك و ف ي	Very smell	0.0500

3. Readings and results

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

In line with our previous work, we will apply this algorithm to texts of various lengths and languages. The results and readings will be presented in Table No. (2), which includes the calculation of Throughput according to its standard formula.

$$\text{Throughput} = \frac{\text{text size}}{\text{Encryption time}}$$

Table 3: - Readings and results by applying the algorithm to several lengths of various texts.

Text length	Free Size $\times 10^{-3}$ k.B	Size on disk $\times 10^{-3}$ k.B	Encription. time m.sec	Thr. (size on disk-enc. time)	Thr. (free size-enc.time)	Decryption. time m.sec	Thr. (size on disk-dec. time)	Thr. (free size-dec. time
50	12,954	16,384	0.0041	4.0320	3.1879	0.0045	3.6267	2.8674
100	12,941	16,384	0.0013	2.9344	10.2163	0.0091	1.7983	1.4204
150	12.955	16.384	0.0095	1.7316	1.3692	0.0015	11.2261	8.8766
200	12.959	16.384	0.0056	2.9488	2.3324	0.0109	1.5017	1.1878
250	12.979	16.384	0.0239	0.6862	0.5436	0.0241	0.6809	0.5394
300	12.991	16.384	0.0047	3.4737	2.7543	0.0290	0.5641	0.4473
350	12.995	16.384	0.0334	0.4903	0.3889	0.0169	0.9668	0.7669
400	13.006	16.384	0.0319	0.5131	0.4073	0.0057	2.8940	2.2974
450	13.015	16.384	0.0189	0.8654	0.6875	0.0411	0.3986	0.3166
500	13.020	16.384	0.0395	0.4147	0.3295	0.0479	0.3424	0.2721
550	13.024	16.384	0.0360	0.4554	0.3620	0.0020	8.3624	6.6475
600	13.035	16.384	0.0508	0.3224	0.2565	0.0559	0.2931	0.2332
650	13.038	16.384	0.0440	0.3723	0.2963	0.0491	0.3335	0.2654
700	13.048	16.384	0.0519	0.3157	0.2515	0.0274	0.5982	0.4764
750	13.057	16.384	0.0490	0.3341	0.2663	0.0128	1.2793	1.0195
800	13.072	16.384	0.0563	0.2908	0.2320	0.0025	6.4497	5.1459
850	13.078	16.384	0.0235	0.6978	0.5570	0.0039	4.1852	3.3407
900	13.091	16.384	0.0087	1.8789	1.5013	0.0739	0.2216	0.1771
950	13.085	16.384	0.0658	0.2488	0.1987	0.0300	0.5452	0.4355
1000	13.097	16.384	0.0948	0.1729	0.1382	0.0034	4.7683	3.8117

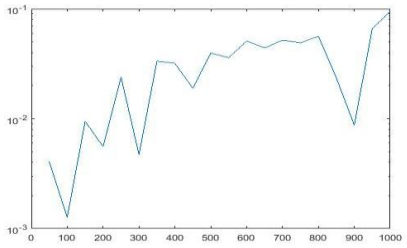
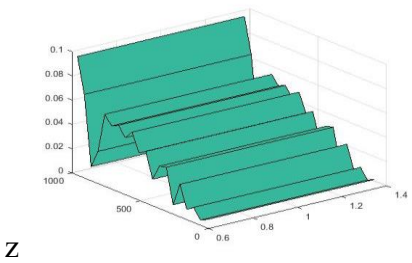
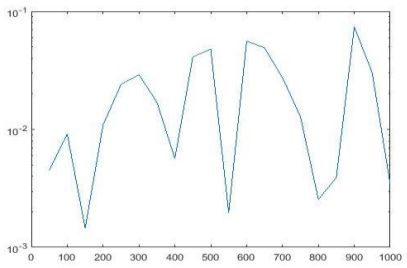
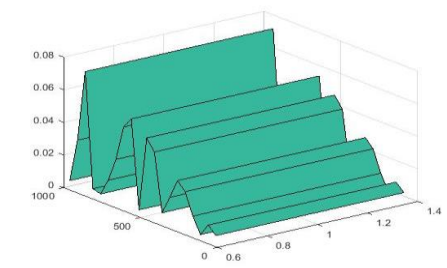
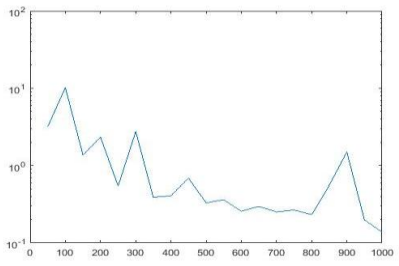
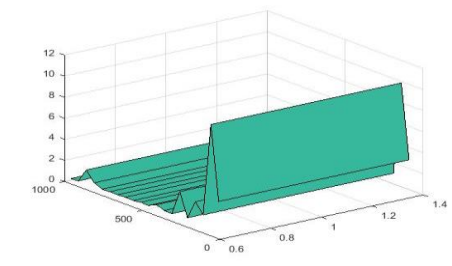
Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

To clarify the relationship between the concepts mentioned in the above table through iterative polygons and iterative strips we designed the figure below that gives a complete picture of all the relationships between the above concepts- :

<table>
<tr><td>  </td><td> 
z</td></tr>
<tr><td>Figure 4 : Repeating polygon Shows the relationship between Number of text characters & Encryption. time</td><td>Figure 5: Histogram Shows the relationship between Number of text characters & Encryption. time</td></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td>Figure 6: Repeating polygon Shows the relationship between Number of text characters & Decryption. time</td><td>Figure 7: Histogram Shows the relationship between Number of text characters & Decryption. time</td></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td>Figure 8: Repeating polygon Shows the relationship between Number of text characters & throughput between free size of text and encryption time</td><td>Figure 9: Histogram Shows the relationship between Number of text characters & throughput between free size of text and encryption time</td></tr>
</table>

Received: July 28, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

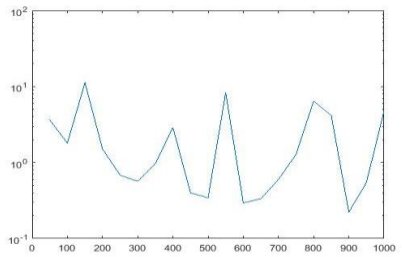


Figure 10: Repeating polygon Shows the relationship between Number of text characters & throughput between free size of text and Decryption. time

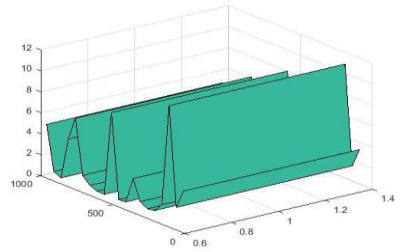


Figure 11: Histogram Shows the relationship between Number of text characters & throughput between free size of text and Decryption. time

4. Conclusions

1. This algorithm is part of the new generation that converts information into numerical values for encryption, suitable for all languages and components.
2. The results from testing on numerous samples are accurate and error-free, applicable to all languages.
3. Both this algorithm and its predecessor (MKTEXT-5) have minimal encryption time, though decryption takes longer due to solving quadratic equations for each text character. This balances its competitiveness with current algorithms.

5. Discussion

Strengths:

1. Inputs match outputs with no errors.
2. Fast encryption and decryption times.
3. Applicable to all texts, components, and languages.
4. Innovative in converting text to a digital vector for encryption and decryption.
5. Uses a general real number as the encryption key.
6. Potential for adapting to encrypt images, videos, and sounds.
7. Maintains the size of English letters.

Weaknesses:

1. The key's absolute value must be within (22, 44) ∪ (44, ∞), requiring a mathematical adjustment.
2. A new value (10 * max(B)) must be added to the vector for accurate results.
3. Does not distinguish between bold, italic, and underscore.

6. Acknowledgments

We extend our thanks to (Maream Mohammed Alkufi) for her linguistic contributions to this and previous research. We wish her continued success in her endeavors.

Received: July 28, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

References

[1] Symmetric-key encryption software Archived 02 August 2017 on the Wayback Machine.

[2] "New cloud attack takes full control of virtual machines with little effort". Ars Technica. Archived from the original on June 07, 2017. Accessed on December 25, 2016.

[3] "Data Encryption in Transit Guideline". Archived from the original on December 10, 2013

[4] Maher Jalal Burjus AL-Bashkani- Image Cryptography using General Singular Values Decomposition - Thesis of master degree- Faculty of Computer Science & Mathematics \ the University of Kufa- January / 2019.

[5] Tufillaro, Nicholas; Abbott, Tyler; Reilly, Jeremiah (1992). An Experimental Approach to Nonlinear Dynamics and Chaos (https://archive.org/ details/unset0000unse_q2b7). Addison-Wesley New York. ISBN 978-0-201-55441-0.

[6] Witold Dzwinel , " Spatially extended populations reproducing logistic map " , AGH Institute of Computer Science, Al. Mickiewicza 30, 30-059 Krakow, Poland . Submitted to Central European Journal of Physics (CEJP), January 2009

[7] Grassberger, P.; Procaccia, I. (1983). "Measuring the strangeness of strange attractors". Physica D. 9 (1–2): 189–208. Bibcode:1983PhyD....9..189G (https://ui.adsabs.harvard.edu/abs/1983PhyD....9..189G). doi:10.1016/0167-2789(83)90298-1 (https://doi.or g/10.1016%2F0167-2789%2883%2990298-1).

[8] Ahmed G. Radwan ," genuine article some generalized discrete logistic maps " , Engineering Mathematics Department, Faculty of Engineering, Cairo University, 12613, Egypt (https://ui.adsabs.harvard.edu/abs/1981JSP....26..173G). doi:10.1007/BF01106792 (https://doi.org/10.1007% 2FBF01106792).

[9] Sprott, Julien Clinton (2003). Chaos and Time-Series Analysis. Oxford University Press. ISBN 978-0-19-850840-3.

[10] Strogatz, Steven (2000). Nonlinear Dynamics and Chaos (https://archive.org/details/nonlineardynamic00stro). Perseus Publishing. ISBN 978-0-7382-0453-6.

[11] Grassberger, P. (1981). "On the Hausdorff dimension of fractal attractors". Journal of Statistical Physics. 26 (1): 173–179. Bibcode:1981JSP....26..173G

[12] MOHAMMED ABDUL HAMEED JASSIM AL-KUFI- Image Encryption with (General Singular Values Decomposition) via logistic function by Encryption key only real number (method-2)- Not published yet.

Received: July 28, 2025