A Fast Text-to-Image Encryption-Decryption Algorithm for Secure Network Communication

Noor Sattar Noor 1, Dalal Abdulmohsin Hammood 1, Ali Al-Naji 1,2,* and Javaan Chahl 2

1 Electrical Engineering Technical College, Middle Technical University, Baghdad 1022, Iraq; bcc0055@mtu.edu.iq (N.S.N.); dalal.hammood@mtu.edu.iq (D.A.H.)
2 School of Engineering, University of South Australia, Adelaide, SA 5095, Australia; javaan.chahl@unisa.edu.au
* Correspondence: ali_al_naji@mtu.edu.iq; Tel.: +964-7710304768

Abstract: Data security is the science of protecting data in information technology, including authentication, data encryption, data decryption, data recovery, and user protection. To protect data from unauthorized disclosure and modification, a secure algorithm should be used. Many techniques have been proposed to encrypt text to an image. Most past studies used RGB layers to encrypt text to an image. In this paper, a Text-to-Image Encryption-Decryption (TTIED) algorithm based on Cyan, Magenta, Yellow, Key/Black (CMYK) mode is proposed to improve security, capacity, and processing time. The results show that the capacity increased from one to four times compared to RGB mode. Security was also improved due to a decrease in the probability of an adversary discovering keys. The processing time ranged between 0.001 ms (668 characters) and 31 s (25 million characters), depending on the length of the text. The compression rate for the encrypted file was decreased compared to WinRAR. In this study, Arabic and English texts were encrypted and decrypted.

Keywords: TTIED-CMYK mode; security; compression; capacity; encryption; decryption

1. Introduction

Sensitive information in the form of text, pictures, audio, and video are exchanged over communication systems. Cryptography is the application of techniques for encrypting data between the sender and receiver so that the receiver can read the data and an adversary cannot. Making sensitive data safe is critical with the overwhelming dependence on electronic communication systems in the modern world. Data may be transferred without the danger of being intercepted by employing several cryptographic procedures [1–4]. Encryption is the most common way to protect large amounts of text data [5]. Cryptography is a process to convert plain text into cipher using two types of key; symmetric and asymmetric key. Symmetric means that the same key is used between sender and receiver in encryption and decryption processes. The examples of symmetric keys are substitution, transposition, hill cipher, Play fair cipher, Vigener, and others. While asymmetric key means that there are two keys in encryption and decryption processing—a private and public key [6–8]. Figures 1 and 2 illustrate the block diagram of the cryptography asymmetric and symmetric keys, respectively. Many approaches, such as Advanced Encryption Standard (AES), Data Encryption Standard (DES), Rivest–Shamir–Adleman (RSA), Triple Data Encryption Standard (3-DES) and other established procedures, are used to encrypt and decrypt data [9–11]. However, these approaches are still ineffective at encrypting large-scale text with high redundancy and large storage capacities. To alleviate these issues, researchers have developed various text-to-image encryption algorithms to increase text security and efficiency, thus improving robust cryptographic algorithms.

In this work, there are two concepts of image encryption, color mode and color code, that need to be considered in data encryption. Color mode encoding is a group of different colors that are mixed to obtain the color required to display the image on the screen or
to print it from a printer [12]. How colors are combined using different color schemes is usually determined by the number of channels in a given color model, so various color schemes result in different levels of color detail and file size. Color code encoding is a system used to show data in the form of different colors where there is a code for each color that is drawn from a table defined by the particular application used [12,13]. For example, the letter ‘A’ represents black and so on. This system is used in several applications, including the color to value encoding of electronic resistors [13], as well as in programming languages, especially in the HTML markup language. The color space system, including the RGB color system (three color channels red, green, and blue, 24 bits of data per pixel, and $2^{24}$ unique colors available), the grayscale system (one intensity channel, 8 bits of data per pixel, and $2^8$ colors available), the CMYK color system (four color channels cyan, magenta, yellow, and black, 32 bits of data per pixel, and $2^{32}$ colors available), and the binary system as shown in Table 1 and Figure 3.

Figure 1. Asymmetric key cryptography.

Figure 2. Symmetric key cryptography.
Table 1. The specifications of the color space systems.

| CMYK Color System | RGB Color System | Grayscale Color System | Binary Color System |
|-------------------|------------------|------------------------|---------------------|
| This system consists of 4 channels cyan, magenta, yellow, and black | This system consists of 3 channels red, blue, and green | This system consists of only one channel gray | Each pixel in this system is black or white |
| The size of the pixel is 32 bits | The size of the pixel is 24 bits | The size of the pixel is 8 bits | The size of the pixel is 1 bit |
| The number of colors available is $2^{24}$ | The number of colors available is $2^{24}$ | The number of colors available is $2^{8}$ | The number of colors available is $2^{0}$ |
| preferable to use in printers | preferable to use on display screens | preferable to use in image processing | preferable to use in image processing |

Figure 3. An example of the color space systems.

This paper presents the CMYK color system to increase the security and capacity of the number of encrypted characters and decrease the size of the encrypted file.

The rest of this paper is arranged as follows: Section 2 presents the related works and mentions the limitations of former works. Section 3 describes the system framework. Section 4 shows the experimental results and analyzes the results obtained. Section 5 shows a discussion of the results and study advantages and limitations. Finally, the conclusion and future research directions are provided in Section 6.

2. Related Work

Researchers in the digital data field are continuously seeking to develop new robust encryption methods to secure private material, either by combining current techniques or by developing new techniques to protect text, image, audio, and video. Some attempts have recently followed the path of encrypting some types of data so as to transform them into another sort of data, such as text-to-image, text-to-audio, audio-to-image, and so on. For example, the text to image encryption algorithm (TTIE) was proposed in [14] which converted each letter in a text file into the RGB of a single pixel in an image to create a matrix. Then, it switched operations on the columns to make things more difficult for the attacker. Thus, the attacker must attempt $(256^3)^{25}$ for text analysis in this approach, which takes a long time. In another example, Abusukhon et al. [15] proposed an encryption technique,
called text-to-image encryption, based on the Diffie–Hellman (DH) key-exchange protocol to create an identical key (represented as a collection of RGB pixels) on both sides (client and server). The limitation of these studies [14–16] is that the hacker can know the number of letters in the main text, as shown in Figure 4.

![Figure 4. The limitation of the text-image encryption algorithm of the previous studies.](image)

Another study by Talib et al. [17] proposed a white-page image encryption algorithm (WPI) to improve security and make it difficult for hackers to guess the key, misleading them to also believe that the text message had no information. Their study employed the range 246 to 255 to construct varying densities of white pixels; however, the disadvantage of this work was that the color space was very small, comprising only nine colors from 246 into 255.

Another encryption method was proposed by Smita et al. [18]. This method was based on the cell division that happens in living organisms to encode a text message into a picture. Both the transmitter and receiver keep an image database to utilize in the encryption process. A picture was chosen to be encrypted with the message depending on the time of day. The R, G, and B values were extracted to get a total of three “chromosomes”, and then the three acquired vectors were split into an equal number of chromosomes (‘n’ from each R, G, and B). Three vectors, one from each of R, G, and B, were chosen and XOR’ed together, reducing 3n to n, and converting the vectors above to DNA sequences. The coding schemes 00-A, 01-T, 10-G, and 11-C were used.

Another study [19] presented a pixel-based text-to-image encryption technique using three stages, including associative memory, text-to-image encryption, and image analysis based on memory and data size. In the first stage, a group of nine characters was created. If the number was not a multiple of nine, additional spaces were added. Then, for each character, an ASCII code was generated. To display the text, the viewer must first split the picture into blocks. A secret key would be used to change each color component. This technique might be applied to a vast number of databases.

Another method for text-to-image encryption and decryption based on piece-wise linear chaotic maps was suggested by [20,21]. This work presented a piece-wise linear chaotic map approach for encrypting text into images (PWLCM). The suggested technique may take a text file as input and transform it into a cipher picture, which could then be decrypted. This method had the benefit of being highly secure and difficult to breach. PWLCM Maps were the simplest of the chaotic maps. The text was first transformed from 7-bit ASCII to its double counterpart, and then padding was performed to provide the appropriate matrix for the image to which the text would be converted. The matrix bits were then permuted, and diffusion occurring in two steps, first for the rows and subsequently for the columns. Finally, the PWLCM map was used for both permutation
and diffusion. However, the length of encrypted characters in file text was small, and the time required for encrypting and decrypting were not considered.

This study proposes a fast TTIED algorithm depending on the CMYK mode where the number of layers increases, the capacity of the characters to be encrypted increases, and the compression rate for encrypted-decrypted files decreases. Moreover, the number of colors available is increased to 4,294,967,295 instead of 16 million characters in prior studies. The total number of keys utilized has now risen to four. The processing time was sped up, leading to an increase in encryption strength and capacity.

3. System Framework

In this work, CMYK mode was proposed to use in the text to image encryption algorithm. CMYK is widely used in printers because it has several advantages to display images clearly and in colors that are close to what is perceived as true.

This mode consists of four layers—CMYK. This can increase the number of characters that are loaded to each pixel in each layer, and thus the capacity becomes four times that of previous studies. Only four keys are used. This implies that depending on the programmer, eight keys per layer can use two keys, or 16 keys per layer can use four keys. Some of the definitions used for explaining the proposed TTIED algorithm are listed below:

- **Plaintext (PT):** Is the original text to be encrypted; this text is input by the sender (user).
- **Length of plaintext or secret text (ST length):** It is necessary to know the number of characters in the original text because the resulting image in this work is directly proportional to the size of the text (the number of letters). One of the advantages of this work is that the smaller the text size, the smaller the image size. In this way, the efficiency of the algorithm increases in terms of encryption strength and saving of storage space.
- **Encryption image or cipher text (EI):** The encrypted image, which is the image resulting from this algorithm after the letters were converted to the color in the image, is CMYK mode, not RGB mode, as each pixel in the image carries four letters.
- **Size of encrypted image (EI size):** The decryption process depends mainly on the size of the encrypted image received by the recipient through the size of the image, and after several calculations are made on the image, the encrypted text is extracted based on the following equations:

\[
\text{EI size} = \text{width} \times \text{length} \times \text{No. of channels}
\]

\[
\text{Ceil} \left( \sqrt{\frac{\text{ST}}{4}} \right)
\]

where Equation (1) is to find the size of image encryption and Equation (2) is for the text size that is proportional to the dimensions of the image encryption.

- **Key:** The key is used in the en/decryption algorithm to obtain the original text.
- **Size of key (KS):** The larger the size of the key, the greater the difficulty in the process of cracking the cipher and thus the greater the strength of the encryption. The size of the key is critical in the process of encryption and decryption.

By default, four keys were used in this paper, each layer has its own key, and as previously mentioned, it is possible to use several keys according to the programmer’s desire. Each letter takes a color, in CMYK this is a 32 bit number of four possible colors with 256 levels each. The first key is for Layer 1 (cyan), the second key is for Layer 2 (magenta), the third key is for Layer 3 (yellow), and the fourth key is for Layer 4 (black). All keys are entered by the sender according to the language to be encrypted: English, Arabic, Hindi or any other language. This is one of the advantages of this work. It is also possible to encrypt a text consisting of letters in various languages. In this work, four keys were used; each key was 37 characters long.

The schematic diagram of the proposed method and the flowchart that shows the main steps of the TTIED algorithm based on CMYK are shown in Figures 5 and 6, respectively.
The schematic diagram of the proposed algorithm.

Key 1, 2, 3, 4 = [0-1, a-z, /, @, #, ....... all ASCII symbol]
= [0 - 4,294,967,295]
Each key is different from the other in terms of letter and color values

Figure 5.
Figure 6. The flowchart of the proposed TTIED–CMYK algorithm.

4. The Results of Experiments and Security Analysis

4.1. Our Experiments

All experiments were carried out using Python on a computer with an Intel Core i7 2.6G Hz CPU, 8 GB of RAM, and 64-bit Windows 10 operating system. Table 2 illustrates the proposed work with different lengths of text, dimensions, number of keys, types and sizes of the images. The encrypted images of the proposed algorithm for different lengths of text from 668 into 25 million characters are shown in Figure 7.

Table 2. The experiments carried out for the proposed algorithm.

| No. of Exp | No. of Character | Dimension | No. of Key | Type of Image | Size of Image |
|------------|------------------|-----------|------------|---------------|---------------|
| 1          | 668              | 13 × 13   | 4          | .png          | 699 byte      |
| 2          | 25,010           | 80 × 80   | 4          | .png          | 1.36 Kb       |
| 3          | 3,001,981        | 867 × 867 | 4          | .png          | 68 Kb         |
| 4          | 12,007,934       | 1733 × 1733 | 4     | .png          | 253 Kb        |
| 5          | 25,315,980       | 2516 × 2516 | 4      | .png          | 509 Kb        |
Figure 7. The encrypted images of the proposed algorithm for different lengths of text.

4.2. Statistical Analysis

4.2.1. Histogram Analysis

The histogram of the ciphered image should be significantly different from the histogram of the plain image, and the histogram of the ciphered image should be as uniformly distributed as possible, which means that the probability of the existence of any color scale value is the same, and it is totally random [22–24]. Figure 8 shows the histogram of plain text, and Figure 9 shows the ciphertext for different text lengths.

Figure 8. Histogram of plain text.
4.2.2. Information Entropy (IE) Analysis

Information Entropy (IE) measures the average amount of information per bit in an image. It provides all of the information that is accessible in the image. Every pixel has a unique value. As a result, the entropy of an encrypted image denotes that each pixel has a uniform probability distribution [25,26]. It is calculated as follows:

$$E(X) = - \sum_{i=1}^{2^L-1} p(x_i) \log_2 (p(x_i))$$

(3)

where E(X) denotes the message source’s entropy (X). The probability of occurrence of Xi is denoted by p(xi). \{0–8\} is the value of IE. For an 8-bit image, it should be close to 8.

The higher the value of information entropy, the more randomness that may be achieved at the pixel level. All experiments showed entropy 1–5. In this study, the entropy depends on the size of the file; when the length of text increases, the entropy is increased, as shown in Table 3.

Table 3. Information entropy for encrypted images.

| Algorithm | Types of Cipher     | Entropy   |
|-----------|---------------------|-----------|
| Exp.1     | Text to image       | 6.92      |
| Exp.2     | Text to image       | 7.792     |
| Exp.3     | Text to image       | 7.792     |
| Exp.4     | Text to image       | 7.791     |
| Exp.5     | Text to image       | 7.799     |

4.2.3. Key Space (KS) Analysis

Security keys are an important element of every encryption technique since the algorithm’s strength is dependent on them. The keys should be able to withstand any form of attack. Large keyspace and high sensitivity are desired qualities of robust secret keys [30–32].

In this work, the attacker requires a lot of time to find the key, and the reason for this is that each pixel is produced by a mixture of four layers, so it carries four letters. The number of colors is $2^L$, then the number of possibilities to find the key can be found as $x^y$, where x is the number of characters used, y is the number of layers, and z is the number of colors available.
In the process of encryption and decryption, we need the English letters (a–z), numbers (0–9) and space, a total of 37 characters, as shown in Table 4. The number of key possibilities can then be calculated by \(37^4\) and equal to \((37^2 \times 10^{2585827972})\).

Table 4. CYMK with four layers and a number of keys.

| No. | Color Image 32 bit, Color Value in Each Pixel | Each Pixel can be Loaded with 4 Letters; Each Letter is 8 bit. |
|-----|-------------------------------------------|----------------------------------------------------------|
| 1.  | 0                                         | 1–37                                                    |
| 2.  | 1                                         | 1–37                                                    |
| 3.  | 2                                         | 1–37                                                    |
| 4.  | 3                                         | 1–37                                                    |
| ... |                                           |                                                         |
| 32. | 4,294,967,295                             | 1–37                                                    |

The number of key possibilities of the proposed TTIED-CMYK algorithm compared to the most similar related studies is shown in Table 5.

Table 5. A comparison of the number of key possibilities of the proposed algorithm compared to the most related studies.

| Methods | The Number of Possible Key Possibilities |
|---------|-----------------------------------------|
| Ref. [2] | \(2^{128}\)                              |
| Ref. [15] | \(2^{65} \times 255^3 \times 255^{32}\) |
| Ref. [33] | \(2^{360}\)                              |
| The proposed work | \(37^2 \times 10^{2585827972}\) |
|             | \(256^2 \times 10^{2585827972}\) |

It is clear from Table 5 that the number of key possibilities when using 256-character ASCII key is calculated by \(256^{42^{32}}\) and equal to \(256^2 \times 10^{2585827972}\).

4.3. Key Sensitive Analysis

Efficient and robust encryption should have a high sensitivity to its secret key, meaning that even slight changes in the secret key should result in totally different cipher images. Moreover, with a slightly changed key, the method must be impossible to decrypt a cipher image or obtain some plain text patterns [34–39]. This is particularly useful for defending against brute force attacks. In this paper, there are four keys, each key consists of 37 or 256 bytes. In this test, applied 37 bytes, the total key size is 148 bytes. The encrypted image is decrypted using four slightly different test keys to examine the key sensitivity of our proposed technique. The plaintext is “This text to test TTIED-CMYK Mode Algorithm”. The results presented in Table 6 show the influence of minor changes to key.

4.4. Time and Capacity Analysis

Many parameters affect the speed of an algorithm, including the CPU’s specifications and structure, memory capacity, text size, software used, and so on. The time required to convert the text to image algorithm is known as execution time. It is the sum of the compile and run times. The execution time should be minimum for the practical implementation of text-to-image encryption. It is generally measured in seconds, milliseconds, or minutes [40]. The image was processed on a layer-by-layer basis, which improved the speed. The capacity was four times compared with others. Table 7 shows that the suggested algorithm
is exceptionally fast, and the processing time was from 1 micro to 31.01 sec depending on
the length of text, as shown in Table 7 and Figure 10.

Table 6. The key sensitivity of the proposed algorithm with slightly changed key.

| Key | Cipher Text | Plaintext |
|-----|-------------|-----------|
| Decryption with the correct key | ![Cipher Text](image1) | This text to test TTIED-CMYK Mode Algorithm |
| Decryption with four slightly different test keys | ![Cipher Text](image2) | vruptogr2phu.tupfahhl40cmj9amz4vz84.b/m5e |

Table 7. Encryption time in experiments 1–5.

| Ref. No. | Capacity per Characters | Time Encryption per Second |
|----------|-------------------------|----------------------------|
| [14]     | 618                     | 18 [41]                    |
| [42]     | 5.77 GB                 | 41.004 h                   |
| [3]      | 200                     | 1                          |
|          | 400                     | 2                          |
|          | 600                     | 4                          |
|          | 800                     | 5                          |
|          | 1000                    | 6                          |
| [2]      | 6                       | 40 [41]                    |
| [41]     | 627                     | 0.7 [41]                   |
| [21]     | 22261                   | NA                         |
| [43]     | NA                      | 60 to produce nine keys    |
|          | 668                     | 0.000001                   |
|          | 25,010                  | 0.03                       |
|          | 3,001,981               | 4.069                      |
|          | 12,007,934              | 17.001                     |
|          | 25,315,980              | 31.01                      |

4.5. Application Program of TTIE_CMYK Mode

To make it easier for the user to use, the code has been converted into an application
program called TTIE_CMYK mode with the extension .exe in the Python language, as
shown in Figure 11.

4.6. Comparison Our Work with Previous Studies

TTIED-CMYK is proposed in this work to improve security and increase the capacity
of the number of characters in TTIED-CMYK image. Figure 12 clears the proposed work
and previous studies in terms of number of layers, number of keys, type of mode—CMYK
and RGB. It is clear that our proposal is better than others. The number of keys is increased
with mode type CMYK. Table 8 shows the comparison between CMYK and RGB mode.
The proposed TTIE-CMYK algorithm in comparison with previous works. 

Figure 10. The relationship between text size and encryption time of the proposed algorithm.

Figure 11. The GUI operator panel of the proposed algorithm.

Figure 12. The proposed TTIE-CMYK algorithm in comparison with previous works.
Table 8. The proposed work compared with others studies.

| Ref. No. | No. of Layers for Each Character | Mode Type | No. Key |
|----------|---------------------------------|-----------|---------|
| Refs.1–3 [3,14,42] | 1:3 | RGB | 2 |
| Ref.4 [16] | 1:3 | RGB | 4 |
| Ref.5 [41] | 1:1 | RGB | 4 |
| Ref.6 [21] | 1:3 | RGB | 3 |
| The proposed method | 1:1 | CMYK 4 Layers | 4 |

4.7. Capacity and File Size Comparison

Figure 13 shows the comparison between the proposed algorithm and QR related to the capacity. It is clear that the capacity is increased in TTIED-CMYK according to the following equation.

\[
\text{Capacity} = \text{length} \times \text{width} \times \text{No.of channels} \tag{4}
\]

No. of channels to CMYK equals four.

Dimension of QR code version 40 equal 177 × 177. The maximum capacity is ~30,000 bytes [44,45]. However, CMYK mode takes the same size as the QR code. Thus, the capacity 177 × 177 × 4 equals ~125,000 bytes.

![Max. capacity](image)

Figure 13. The capacity comparison between the proposed work and WINRAR in terms of data.

Figure 14 shows the file size comparison between the proposed algorithm and WINRAR. Two experiments were carried out. The first test was to reduce the size of the image from 18 KB with WinRAR to 7 KB with TTIE-CMYK. Secondly, the size of the image was decreased from 26 KB with WinRAR to 10 KB with TTIE-CMYK.
5. Discussion

In this paper, the proposed TTIED-CMYK algorithm was shown to be useful in several applications, such as wireless sensor networks and the internet of things (IoT). The proposed algorithm has a high encryption strength in terms of the number of keys, the size of each key, and high capacity at different aspects, such as the size of text, the number of keys, and the execution time. It opens a platform for academics to investigate ways to improve the speed of the text-to-image algorithm, as well as increasing storage capacity, as the algorithm does not require much storage space, and security, as the algorithm provided a high level of protection. The proposed algorithm also has random behavior and extremely fast computations in comparison with previous studies. The key size is 148 or 1024 bytes, and it has a large secret key (4 keys; each key size is 37 or 256 bytes). However, TCP transmission is required; therefore, it is preferably used in e-mail, and key exchange is required.

6. Conclusions

This paper introduces the TTIED-CMYK algorithm to improve security, capacity, and processing time. CMYK mode consists of four layers; Cyan, Magenta, Yellow, and Black. The results were produced using the Python language and showed that capacity increased from 1 to 4 times compared with RGB mode, defining improvement as being based on the probability of finding keys. The processing time range was between 0.001 ms for 668 characters and 31 s for 25 million characters depending on the length of text. Both Arabic and English languages were encrypted and decrypted in this study, and there was no constraint on the language of the input text. In TTIED-CMYK mode, the capacity was 125,000 bytes (characters) compared to QR codes which are ~30,000 bytes (characters), and the file size was reduced by 7 KB compared to the WinRAR archive, which was 18 KB. The key combinatorial space was increased, as shown in Tables 4 and 5. This proposed method gives good results compared to previous work. In future work, we recommend using plain text sensitivity, image to image encryption to validate CYMK with NPCR and UACI tests, and the Chi-squared test.

Author Contributions: Conceptualization, A.A.-N.; methodology, N.S.N. and D.A.H.; software, N.S.N.; validation, N.S.N. and D.A.H.; formal analysis, N.S.N. and D.A.H.; investigation, N.S.N., D.A.H. and A.A.-N.; resources, N.S.N. and D.A.H.; data curation, N.S.N. and D.A.H.; writing—original draft preparation, N.S.N. and D.A.H.; writing—review and editing, A.A.-N. and J.C.; visualization, A.A.-N. and J.C.; supervision, A.A.-N. and D.A.H.; project administration, A.A.-N., D.A.H. and J.C.; funding acquisition, A.A.-N. and J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.
Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Jebur, R.S.; Der, C.S.; Hammood, D.A. A Review and Taxonomy of Image Denoising Techniques. In Proceedings of the 6th International Conference on Interactive Digital Media (ICIDM), Bandung, Indonesia, 14–15 December 2020; pp. 1–6. [CrossRef]
2. Singh, A.; Gilhorta, R. Data security using private key encryption system based on arithmetic coding. Int. J. Netw. Secur. Its Appl. 2014, 2, 58–67. [CrossRef]
3. Singh, S.; Jain, A. Combination of RGB Substitution for Text to Image Encryption Technique using AES. Sputniks Int. J. Eng. Sci. Technol. 2015, 2, 1–7.
4. Naji, M.A.; Hammood, D.A.; Atee, H.A.; Jebur, R.S.; Rahim, H.A.; Ahmad, R.B. Cryptanalysis cipher text using new modeling: Text encryption using elliptic curve cryptography. AIP Conf. Proc. 2020, 2203, 020003. [CrossRef]
5. Abusukhon, A. Block cipher encryption for Text-to-Image Encryption algorithm. Int. J. Comput. Eng. Technol. (IJCET) 2013, 4, 50–59. Available online: https://www.researchgate.net/publication/282656379 (accessed on 6 February 2022).
6. Padhiar, S.; Mori, K.H. A Comparative Study on Symmetric and Asymmetric Key Encryption Techniques. In Implementing Data Analytics and Architectures for Next Generation Wireless Communications; IGI Global: Hershey, PA, USA, 2022; pp. 132–144. [CrossRef]
7. Kapoor, J.; Thakur, D. Analysis of Symmetric and Asymmetric Key Algorithms. In ICT Analysis and Applications; Springer: Singapore, 2022; pp. 133–143. [CrossRef]
8. Alaya, B.; Sellami, L. Clustering method and symmetric/asymmetric cryptography scheme adapted to securing urban VANET networks. J. Inf. Secur. Appl. 2021, 58, 102779. [CrossRef]
9. Naji, M.; Hammood, D.A.; Atee, H.A.; Yahia, S. Implantation of Caesar and Hill Cipher on Database for Better Security. Eur. J. Technol. Eng. 2019, 18, 24.
10. Ibada, A.J.; Ehkan, P.; Ngadiran, R.; Hammood, D.A.; Alkhayyat, A. RGB Image Encryption using Hill Algorithm and Chaos System. J. Physics: Conf. Ser. 2021, 1962, 012061. [CrossRef]
11. Fatil, P.; Narayankar, P.; Narayan, D.G.; Meena, S.M. A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish. Procedia Comput. Sci. 2016, 78, 617–624. [CrossRef]
12. Khanam, R.; Johri, P.; Divían, M.J. Human Skin Color Detection Technique Using Different Color Models. In Trends and Advancements of Image Processing and Its Applications; Springer: Cham, Switzerland, 2022; pp. 261–279.
13. Alemán, S.M.; Alemán, S.M.; Gaytán, S.; Gaytán, S. It doesn’t speak to me’: Understanding student of color resistance to critical race pedagogy. Int. J. Qual. Stud. Educ. 2016, 30, 128–146. [CrossRef]
14. Abusukhon, A.; Talib, M. A novel network security algorithm based on private key encryption. In Proceedings of the 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), Kuala Lumpur, Malaysia, 26–28 June 2012; pp. 33–37. [CrossRef]
15. Abusukhon, A.; Anwar, M.N.; Mohammad, Z.; Alghannam, B. A hybrid network security algorithm based on Diffie Hellman and Text-to-Image Encryption algorithm. J. Discret. Math. Sci. Cryptogr. 2019, 22, 65–81. [CrossRef]
16. Arun, M.; Mohamedazarudeen, S.; Nivek, T.N. AES based Text to Pixel Encryption using Color Code Conversion by Modulo Arithmetic. Int. J. Recent Res. Sci. Eng. Technol. 2015, 1, 37–42.
17. Abusukhon, A.; Mohammad, Z.; Talib, M. A novel network security algorithm based on encrypting text into a White-Page Image. Lect. Notes Eng. Comput. Sci. 2016, 2225, 410–414. [CrossRef]
18. Vaishnavikannan; Jhajharia, S. Using Cell Division for Text to Image Encryption. In Proceedings of the 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 7–8 January 2016; pp. 1–7.
19. Malathi, K.; Kavitha, R.; Liza, M. Pixel based method for Text to Image Encryption. In Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA 2020), Coimbatore, India, 5–7 November 2020; pp. 1001–1004. [CrossRef]
20. Bououdjen, R.; Abdelouahab, M.S.; Jarad, F.; Hammouch, Z. A novel fractional piecewise linear map: Regular and chaotic dynamics. Int. J. Gen. Syst. 2021, 50, 501–526. [CrossRef]
21. Kolhe, M.L.; Tiwari, S.; Trivedi, M.C.; Mishra, K.K. Text-to-Image Encryption and Decryption Using Piece Wise Linear Chaotic Maps. In Advances in Data and Information Sciences; Springer: Singapore, 2019; pp. 23–33.
22. Daoui, A.; Karmouni, H.; Sayyouri, M.; Qidaa, H.; Maaroufi, M.; Alami, B. New robust method for image copyright protection using histogram features and Sine Cosine Algorithm. Expert Syst. Appl. 2021, 177, 114978. [CrossRef]
23. Dhal, K.G.; Das, A.; Ray, S.; Gálvez, J.; Das, S. Histogram Equalization Variants as Optimization Problems: A Review. Arch. Comput. Methods Eng. 2020, 28, 1471–1496. [CrossRef]
24. Luo, W.; Duan, S.; Zheng, J. Underwater Image Restoration and Enhancement Based on a Fusion Algorithm with Color Balance, Contrast Optimization, and Histogram Stretching. IEEE Access 2021, 9, 31792–31804. [CrossRef]
25. Li, G.; Zhang, X.; Yang, H. Complexity Analysis of Three-Dimensional Fractional-Order Chaotic System Based on Entropy Theory. *IEEE Access* 2021, 9, 73012–73028. [CrossRef]
26. Sparavigna, A.C. Entropy in Image Analysis III. *Entropy* 2021, 23, 1648. [CrossRef]
27. Khan, J.S.; Ahmad, J. Chaos based efficient selective image encryption. *Multidimens. Syst. Signal Process.* 2019, 30, 943–961. [CrossRef]
28. Ahmad, J.; Khan, M.A.; Ahmed, F.; Khan, J.S. A novel image encryption scheme based on orthogonal matrix, skew tent map, and XOR operation. *Neural Comput. Appl.* 2018, 30, 3847–3857. [CrossRef]
29. Nestor, T.; De Dieu, N.J.; Jacques, K.; Yves, E.J.; Iliyasu, A.M.; El-Latif, A.A.A. A Multidimensional Hyperjerk Oscillator: Dynamics Analysis, Analogue and Embedded Systems Implementation, and Its Application as a Cryptosystem. *Sensors* 2019, 20, 83. [CrossRef]
30. Huang, Z.-J.; Cheng, S.; Gong, L.-H.; Zhou, N.-R. Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform. *Opt. Lasers Eng.* 2020, 124, 105821. [CrossRef]
31. Tutueva, A.V.; Karimov, A.I.; Moysis, L.; Voles, C.; Butusov, D.N. Construction of one-way hash functions with increased key space using adaptive chaotic maps. *Chaos Solitons Fractals* 2020, 141, 110344. [CrossRef]
32. Lakshmi, C.; Thenmozhi, K.; Rayappan, J.B.B.; Amirtharajan, R. Hopfield attractor-trusted neural network: An attack-resistant image encryption. *Neural Comput. Appl.* 2019, 32, 11477–11489. [CrossRef]
33. Panwar, K.; Purwar, R.K.; Jain, A. Cryptanalysis of an Image Encryption Scheme using Reverse 2-dimensional Chaotic Map and Dependent Diffusion. In Proceedings of the 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 22–23 February 2018; pp. 236–239. [CrossRef]
34. Degadwala, S.D.; Kulkarni, M.; Vyas, D.; Mahajan, A. Novel Image Watermarking Approach against Noise and RST Attacks. *Procedia Comput. Sci.* 2020, 167, 213–223. [CrossRef]
35. Liu, J.; Wang, Y.; Liu, Z.; Zhu, H. A chaotic image encryption algorithm based on coupled piecewise sine map and sensitive diffusion structure. *Nonlinear Dyn.* 2021, 104, 4615–4633. [CrossRef]
36. Shah, A.A.; Parah, S.A.; Rashid, M.; Elhoseny, M. Efficient image encryption scheme based on generalized logistic map for real time image processing. *J. Real-Time Image Process.* 2020, 17, 139–2151. [CrossRef]
37. Wang, X.; Yang, J.; Guan, N. High-sensitivity image encryption algorithm with random cross diffusion based on dynamically random coupled map lattice model. *Chaos Solitons Fractals* 2020, 143, 110582. [CrossRef]
38. Sumathi, M.; Sangeetha, S. A group-key-based sensitive attribute protection in cloud storage using modified random Fibonacci cryptography. *Complex Intell. Syst.* 2020, 7, 1733–1747. [CrossRef]
39. Midoun, M.A.; Wang, X.; Talhaoui, M.Z. A sensitive dynamic mutual encryption system based on a new 1D chaotic map. *Opt. Lasers Eng.* 2020, 139, 106485. [CrossRef]
40. Arai, T.; Okabe, Y.; Matsumoto, Y. Precursory Analysis of Attack-Log Time Series by Machine Learning for Detecting Bots in CAPTCHA. In Proceedings of the 2021 International Conference on Information Networking (ICOIN), Jeju Island, Korea, 13–16 January 2021; pp. 295–300. [CrossRef]
41. Al-Husainy, M.A.F. A New Implementation of Text-To-Image Encryption Using Multilevel Substitution and Transposition. *Int. J. Comput. Sci. Inf. Secur.* 2017, 14, 851–857.
42. Abusukhon, A.; Talib, M.; Nabulsi, M.A. Analyzing the Efficiency of Text-to-Image Encryption Algorithm. *Int. J. Adv. Comput. Sci. Appl.* (IJACSA) 2012, 3. [CrossRef]
43. Abusukhon, A.; Mohammad, Z.; Al-Thaier, A. An authenticated, secure, and mutable multiple-session-keys protocol based on elliptic curve cryptography and text-to-image encryption algorithm. *Concurrency Comput. Pr. Exp.* 2021, 34, e6649. [CrossRef]
44. Al-Zahrani, M.S.; Wahsheh, H.A.M.; Alsaaede, F.W. Secure Real-Time Artificial Intelligence System against Malicious QR Code Links. *Secur. Commun. Networks* 2021, 2021, 5540670. [CrossRef]
45. Fu, Z.; Liu, S.; Sun, R.; Fang, L.; Yu, B. Distributed color QR code with high-capability and fast decoding. *Multimed. Tools Appl.* 2020, 79, 32471–32485. [CrossRef]