applied sciences

MDPI

Article

A Dynamic Multi-Layer Steganography Approach Based on Arabic Letters' Diacritics and Image Layers

Saad Said Alqahtany [1], Ahmad B. Alkhodre [1], Abdulwahid Al Abdulwahid [2] and Manar Alohaly [3,*]

[1] Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 42351, Saudi Arabia; aalkhodre@iu.edu.sa (A.B.A.)
[2] Computer and Information Technology Department, Jubail Industrial College, Royal Commission for Jubail and Yanbu, P.O. Box 10099, Jubail Industrial City 31961, Saudi Arabia; abdulwahida@rcjy.edu.sa
[3] Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
* Correspondence: mfalohaly@pnu.edu.sa

Abstract: Steganography is a widely used technique for concealing confidential data within images, videos, and audio. However, using text for steganography has not been sufficiently explored. Text-based steganography has the advantage of a low bandwidth overhead, making it a promising alternative for protecting sensitive information. Among languages, Arabic is known for its linguistic richness, making it ideal for text-based steganography. This paper proposes a robust, dynamic, and multi-layered steganography approach that uses text, encryption algorithms, and images. This approach utilizes Arabic diacritic features to hide limited-size and highly classified information. The algorithm uses several scenarios and is extensively tested to ensure the required level of security and user performance. The experimental results on actual data demonstrate the robustness of the proposed algorithm, with no noticeable impact on the carrier message (original text). Furthermore, no known potential attack can break the proposed algorithm, making it a promising solution for text-based steganography.

Keywords: steganography; security and cryptography; Arabic text; text hiding; privacy

Citation: Alqahtany, S.S.; Alkhodre, A.B.; Al Abdulwahid, A.; Alohaly, M. A Dynamic Multi-Layer Steganography Approach Based on Arabic Letters' Diacritics and Image Layers. *Appl. Sci.* **2023**, *13*, 7294. https://doi.org/10.3390/app13127294

Academic Editor: Habib Hamam

Received: 5 May 2023
Revised: 6 June 2023
Accepted: 14 June 2023
Published: 19 June 2023

1. Introduction

With the rapid development of the global information and big data industry, significant challenges have emerged in data protection and concealment. Digital data are much more easily exposed to theft and forgery than before. Steganography presents itself as a promising solution for data confidentiality and protection. Figure 1 shows the four essential components of steganography systems: the cover (aka envelope), the confidential message (information to be hidden), the steganography algorithm, and the insecure channel [1].

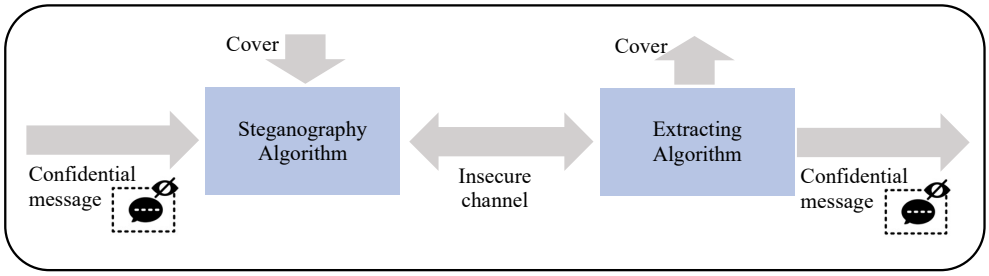

Figure 1. General scheme of steganography.

The robustness of the steganography algorithm is one of its most important characteristics in ensuring the confidentiality of the secret message against potential breaches [2].

On the other hand, the algorithm must ensure a coherent encoding of the secret message to protect the concealed data from damage [3]. The cover can be photo, video, audio, or text. Several scientific studies have been published recently on steganography and can be classified according to the type of cover used (Figure 2) [4].

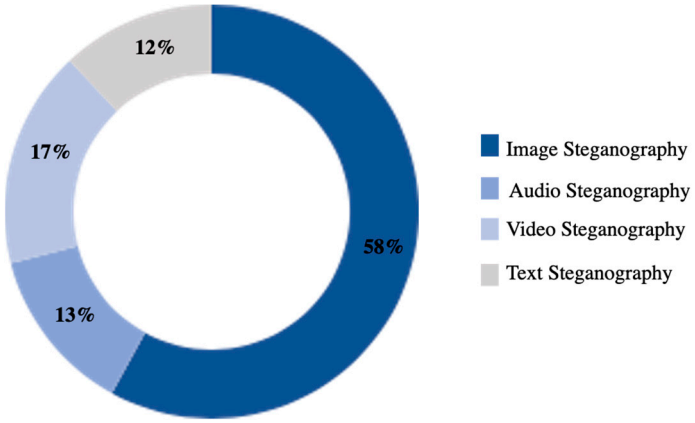

Figure 2. The used cover in steganography processes.

Steganography is usually used for two primary purposes—for the protection of intellectual property and copyright (watermarking) and the concealment of confidential information. Steganography has been intensively applied to images, as it is easy to modify an image to conceal the data in a way that is imperceptible to the naked eye. As for images, most steganography algorithms change the least important bits in the image without affecting the appearance of the original image. Despite recent research efforts in text-based steganography, the majority of these studies examined applications in the English language.

There are some research efforts in steganography applications in other languages, such as Persian and Urdu, but Arabic has been less well researched compared to other languages. Arabic, Persian, and Urdu are all part of the same language family and share many similarities. However, there are some differences in the way diacritics are used in these languages. In Arabic, diacritics are used to indicate short vowels and other linguistic features. There are three basic diacritics in Arabic: Fatha, Kasrah and Dammah. These diacritics are used to indicate the pronunciation of Arabic words and are an important part of learning to read and write the language. On the other hand, Persian diacritics are used less frequently than in Arabic. Persian also does not have a system of short vowels like Arabic. Similarly, Urdu, like Persian, uses diacritics less frequently than Arabic. Overall, while all three languages use diacritics to some extent, their heavy usage and varied purposes in Arabic provide more variations and options to embed secrets. Hence, this paper focuses on steganography using Arabic, as it is one of the richest living languages [5].

The Arabic language differs from other natural languages; the pronunciation of words and letters is based not only on the position of the letters in the word but also on the word's position in the context of the sentence. For this reason, in the seventh century, Al-Khalil bin Ahmed Al-Farahidi, an Arab linguist, introduced diacritical marks on letters to distinguish them from one another [6]. Table 1 shows an example of some Arabic diacritics and how the letters are pronounced based on the assigned diacritics. The Arabic language has 28 different letters, and with the presence of diacritics as well as letter placements, the number of possibilities for writing each letter with seven diacritics becomes 196, in addition to a set of special case characters such as ,ـه,ـي,ـى and ۘ. This variety makes Arabic script more suitable for embedding confidential information [7]. The different methods that have been used in the literature are dots, diacritics, Kashida, Unicode, sharp edges of letters, poetry, and hybrid styles. This paper proposes a robust, multi-layered steganography approach and algorithm for limited-size and highly classified information utilizing Arabic diacritic features.

Table 1. Diacritics in the Arabic Language.

النطق **Pronunciation**	الرمز **Symbol**	الحركة **Diacritics**	الحرف **Letter**
با	◌َ	فتحة	الباء
بو	◌ُ	ضمة	
بي	◌ِ	كسرة	
ب	◌ْ	سكون	
بن	◌ً	تنوين الفتح	
بون	◌ٌ	تنوين الضم	
بين	◌ٍ	تنوين الكسر	
ابب	◌ّ	الشدة	

This research work's contributions can be summarized as follows:

1. The dynamic multi-layered concept: Previous research studies primarily focused on specific techniques or a combination of techniques to conceal information, such as using Arabic movements or kashida within texts. However, our work introduces a dynamic multi-layered concept that enables the concealment of secrets of various sizes. This approach provides flexibility and adaptability in hiding information within texts.
2. A novel steganography algorithm: We propose a novel steganography algorithm that specifically addresses the method of hiding text with a text cover in different ways and scenarios and then embeds it in a cover image as another layer. By incorporating Arabic diacritics and Hamzas, our algorithm enhances the concealment process and makes it highly challenging for potential attackers to detect hidden information. This algorithm offers improved security and resilience against detection.

These contributions expand the body of knowledge on steganography by introducing a dynamic multi-layered concept and a new steganography algorithm that optimizes the concealment of information within texts. The proposed techniques enhance the security and effectiveness of text-based steganography, providing researchers and practitioners with valuable tools to protect sensitive information in various contexts.

This paper is organized as follows: Section 2 provides an overview of the previous research focused on the process of steganography using the Arabic language. Section 3 details the proposed approach, and Section 4 discusses the results, including a comparison with similar previous studies. Section 5 concludes the study.

2. Related Work

In this section, we briefly review image-based steganography methods. Then, we provide a detailed review of Arabic text-based steganography methods.

2.1. Image Steganography Methods

Based on literature studies in steganography, two basic ways to embed sensitive data in images exist. The first is spatial domain methods, and the second is the transform domain technique [8].

The first method relies on a direct change in some of the bits in the image without causing noticeable modifications to the image's appearance. For this, the binary-encoded secret is embedded into the least significant bit [9,10]. The researchers in [11] used the genetic algorithm to find the most suitable places for hiding. Tahir et al. used all RGB color image bits for embedding [12]. Furthermore, the authors of [10,13] used edge technology in the images to embed secrets. The second method of image-based steganography relies on changing the features of the secret before hiding it in the picture to increase security and complicate the detection process. For example, with encryption, the original information cannot be fully recovered if the secret hidden in the image is not found. On the other

hand, to raise the level of security, other methods adopted inserting bits in a different order from the natural sequence of pixels, such as inserting a pixel that achieves a significant difference from the adjacent pixel so that it does not exceed a certain threshold [14,15]. Some approaches also used special encodings and did not rely on standard encodings to increase the level of confidentiality [16], whereas others compressed the data before encoding it to increase the size of the message that can be hidden [17]. Furthermore, when working with images, there are many options to increase the level of complexity, such as embedding in the gray level or within the color layers in a certain agreed order [18,19], or in the histogram instead of the basic image [20]. Moreover, [21] utilized double-cover steganography by applying two covers in the concealment process to increase the security of secret information.

In 2016, Divya and Sasirekha introduced a steganography technique based on the discrete wavelet transform (DWT) to minimize distortion in the cover image. However, this method involves a trade-off between security and imperceptibility, prioritizing higher security over imperceptibility [22]. In 2018, Kumar et al. proposed a DWT-based steganography approach that employed three different coefficients—horizontal, vertical, and diagonal—to conceal confidential images. They also presented a secret key technique to mitigate visual quality distortion in the stego-image [23]. El-Khamy et al. (2017) discussed an intriguing method for audio steganography to improve payload capacity, security, and robustness by utilizing two levels of integer wavelet transform and modifying wavelet and chaotic map coefficients. However, this scheme achieved a hiding capacity of 25% of the cover image and an SNR of 44.6 dB [24]. In 2018, Shafi et al. proposed an image steganography technique based on fuzzy wavelet and DWT, which they tested by embedding text within an image [25]. Finally, in 2019, Sharifzadeh et al. proposed a CNN-MLP-based classifier for image classification, combining the positive features of a convolutional neural network (CNN) and multilayer perceptron (MLP) to enhance image detection capabilities [26]. In 2022, Durafe et al. proposed an algorithm that introduces a robust and blind color image steganography method using fractal cover images, SVD, IWT, and DWT. This algorithm achieves high security, imperceptibility, and a large hiding capacity. Performance analysis shows minimal image degradation and efficient resource utilization [27].

In conclusion, the proposed image steganography algorithm presents a novel approach using fractal images as cover images. These fractal images, generated through unique mathematical equations and characterized by specific parameters, provide a complex framework for hiding information. Furthermore, by incorporating a combination of techniques such as SVD and DWT or SVD and IWT, the algorithm addresses the limitations of using SVD alone, ensuring enhanced security and robustness. Furthermore, this hybrid approach offers various benefits, including the ability to resist noise interference, maintain image quality even with minor changes to singular values, achieve robust concealment using the DWT-SVD combination, enable simultaneous compression from loss to lossless with IWT-SVD, facilitate performance comparison for optimal technique selection, and preserve color features in image steganography. Overall, this algorithm introduces an effective and versatile solution for secure and efficient information hiding within fractal cover images.

2.2. Arabic Text Steganography Methods

Several scholars have proposed techniques to embed sensitive data in Arabic text and other languages [28,29]. Their studies focused on the characteristics of the language, such as punctuation marks, diacritics, and others. This subsection reviews existing work on Arabic text steganography and presents its advantages and disadvantages.

Using Dot positions: The Arabic language contains 15 dotted letters out of its 28 letters, which are shown in Figure 3.

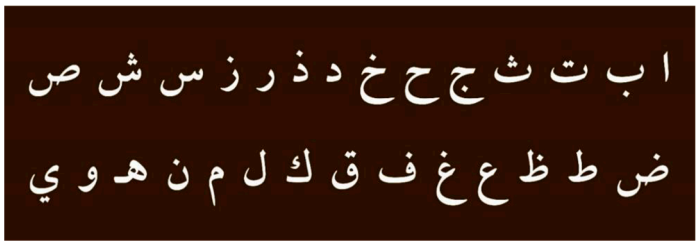

Figure 3. Letters of the Arabic language.

The research in this area focuses on changing the positions of the dots of the letters. The position of the dot in the letter itself is changed by shifting it, as presented in Figure 4 [30].

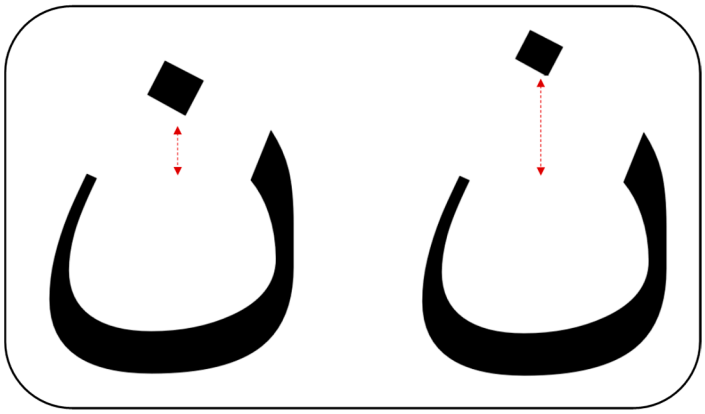

Figure 4. Example for shifting the position of the dot.

This method has advantages in that there are 16 Arabic characters (15 as mentioned earlier plus the letter Kaf (ك)) that can be used in the embedding process, in addition to the different types of calligraphy, from Al-Diwani, Al-Raq'ah, Al-Farsi, Al-Thulus, and Al-Naskh [31]. However, the common shortcomings of this approach are:

- The process of extracting the text requires an optical character recognition (OCR) process to extract it to establish the locations of the variable points in their places, in addition to the fact that a change in font type would lead to complete loss of the embedded data;
- The position of the hidden bits is not random; thus, the predictability can be exploited;
- Low sustainability.

Using Kashida: Kashida is implemented in Arabic and Persian by stretching a letter, as shown in Figure 5. This method allows for extending the original text with characters to hide the secret message. One of this method's most significant challenges is alleviating text distortion. The outcome can be compressed to enhance security. Several research studies have focused on this field.

Normal text	Using Kashida
مرحبا	مرحبـــــا
سؤال	ســـؤال
محمد	محمـــــد

Figure 5. Example of applying Kashida.

In [28], the authors proposed a static steganography algorithm. However, the proposed algorithm is breakable, and the mode of action can be discovered easily. In [32], the authors

combined more than one method with the Kashida, including using special symbols to hide the secret. The researchers in [30] hid the secret in the Kashida, using different and more complex scenarios.

The researchers in [31,33] presented two ways to hide the text: a false spaces-based approach and a hybrid approach combining false spaces with the Kashida. This method can be used in all languages using Arabic alphabets, including Persian and Urdu. However, this method uses a pattern-based technique for the embedding process, which might lead to secret predictability, while [34] used only the dots. The previous algorithms were modified by adding bits through the connection of the characters and their extension to hide one bit, as well as using a zero-width character to embed two bits per cursive [35].

Using letter shapes and edges: This method relies on the shape of the Arabic letter and hides the text in the edges (Figure 6 shows an example of the edges of the Arabic letter)

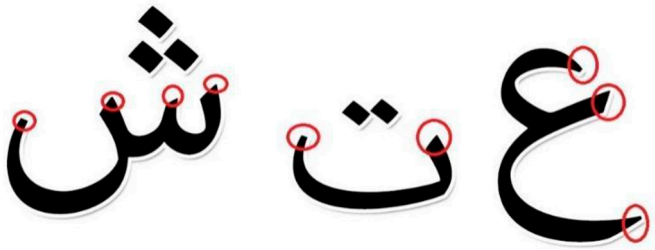

Figure 6. Example for letter shapes and edges method.

This method has a high secret embedding capacity [36]. It is also a performance-friendly steganography approach as it does not require re-writing the text. Instead, it only modifies the edges of the letter to hide the secret. In [37], the authors extended this approach to hide secrets in the dots of 15 dotted Arabic letters. However, despite the potential of this approach, it is vulnerable to predictability exploitation, as the concealment is not performed with random locations. In addition, it requires OCR to reveal the hidden message.

Using character encoding: This method exploits the fact that some letters, namely the letter Kaf (ك) and the letter Yaa (ي), have multiple encodings according to their placement in a word (see Table 2) [33]. This approach is mostly used to ensure imperceptibility, as the hidden text that the user sees is the same as the original text. However, the problem remains with the limited capacity for hiding the secret message.

Table 2. Example of the character encoding method based on a letter's position.

معزولاً **Isolated**	في أخرها **At its End**	في أوسطها **In its Middle**	في أول الكلمة **At the Beginning of the word**
ي	ـي	ـيـ	يـ
ك	ـك	ـكـ	كـ

The authors of [38] hid information in Arabic texts using invisible letters, namely ZWNJ (zero width non-joiner) and ZWJ (zero width joiner). ZWNJ and ZWJ are used as connectors to concatenate other letters together. This steganography method has limited capacity, as it relies only on these two symbols. However, it is characterized by imperceptibility [39]. In [40], the authors proposed a steganography approach that causes no noticeable change in the original text using separate letters in Arabic. The authors of [41,42] extended [40] by encrypting the secret before hiding it. While encryption provides robust concealment, relying solely on two symbols in the encryption process limits the system's overall capacity.

Using Diacritics: Several studies proposed diacritics-based steganography approaches. The authors of [43] used only the Fatah diacritic for concealment, as it is one of the most

frequently used diacritics in Arabic text. A disadvantage of this approach is the small secret capacity, as it only uses one diacritic.

Mohamed et al. improved the previous approach by using more than one diacritic [38]; however, this approach lacks coherence and robustness. The researchers in [44,45] took advantage of the fact that the diacritics in Arabic texts are not compulsory. Accordingly, they proposed hiding the diacritics or keeping them based on the secret. One of the advantages of this method is its high capacity, as the hidden diacritics themselves may contain some information. One of the weaknesses of this method is that the changes in diacritics' positions must be secure.

Another approach was proposed based on stacking diacritics in the written text without obvious visual indicators [46]. Similarly, [47] proposed a steganography approach based on reversing the direction of diacritics, as in Figure 7. However, one of the limitations of the methods proposed in [46,47] is their small capacity due to the number of diacritics and the ease of detection. In addition, [48] utilized the Fatah (ً) and Kasrah (ٍ) diacritics in the concealment process. This drawback is that the hiding places in the cover text are not random.

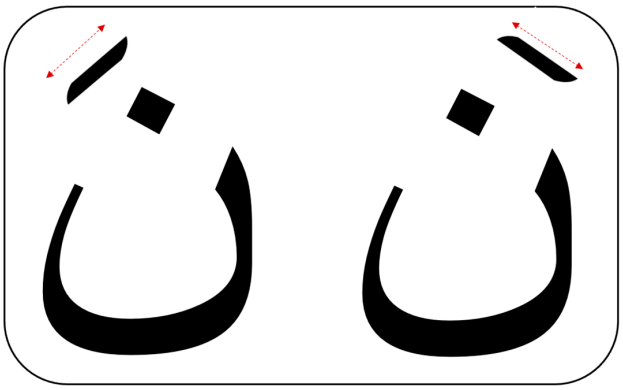

Figure 7. Example for diacritics direction method.

After analyzing previous studies, we found that several methods were proposed to hide secrets in Arabic texts. Some of these methods involve changing the form of the diacritics or stretching the letters, while others involve replacing one diacritic with another. Some studies also used a combination of these methods. In addition, researchers have focused on specific criteria to test the effectiveness of their methods, such as the capacity of the steganography method, the imperceptibility of the hidden secrets, and the robustness of the methods.

However, based on our literature review, embedding secrets in texts is considered one of the weakest steganography methods compared to embedding secrets in pictures and video files. This is because this method has limited secret capacity. To overcome this limitation, we propose a multi-layered system that uses a new method of concealing secrets in Arabic texts. This method is difficult to detect with the naked eye and uses a multi-layered approach to conceal medium-sized and large secrets.

3. The Proposed Approach

This paper proposes a robust and multi-layered algorithm to hide information of limited size which is highly confidential and important, such as in military applications. Through our research, we aim to provide a solution that is more effective than the previous solutions in the steganography process by using the properties of the Arabic language. We focus on hiding confidential information in the diacritics of Arabic letters and Hamzas in more than one scenario. This will have a significant impact on the amount of text to be hidden and its robustness on the one hand, and on the other hand, we focus on the variety of hiding scenarios. In addition, for the text to be highly protected, we rely on the method of multi-layering. We first encrypt the text, hide it within a text cover, encrypt the result

and then embed it within an image cover. The proposed layers provide added protection to maintain the confidentiality of the hidden text. To recover the hidden message, the receiver reverses the previous steps.

The rest of this section presents the stages of the embedding process in some detail, starting by presenting the prototype of our proposal, which consists of four layers, as shown in Figure 8.

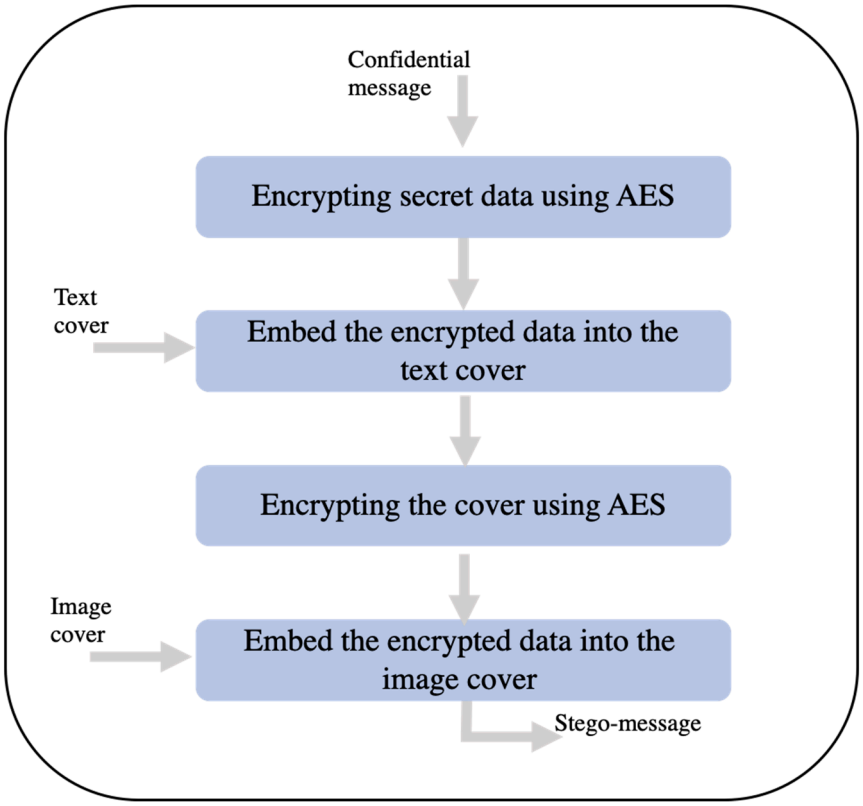

Figure 8. Layers of the proposed approach.

Phase 1: Encrypting secret data using AES

At this phase, to increase the complexity of steganography analysis, the secret will be encrypted using the AES algorithm [49].

Phase 2: Embed the encrypted data into the Cover

In the process of hiding confidential information, we will follow two different scenarios, one of which can be chosen according to the nature of the application or data, as follows:

First scenario: This scenario depends on the diacritics used in the Arabic language in a way that is unique compared to other languages. Therefore, storing the encoded data resulting from the first step in the form of zeros and ones will depend on changing some parsing characters. For example, the diacritic will be changed to a Fatha (ً) in the case of 0 and to a Dammah (ٌ) in the case of 1. However, the switch within the diacritics will be in an agreed sequence (Like Table 3) between the two parties (the sender and the receiver), in the sense that there is a map agreed upon between the two parties in the exchange places for information.

Table 3. Example for sequences of positions for hiding.

The bit number of the secret message to be hidden	1	2	3	4	5	6	7	8	9	10	11	12	...
The number of the diacritic used in concealment	1	5	50	70	71	72	80	100	102	104	105	106	...

Thus, it is very difficult for any opponent to know where the data are embedded into the text, especially since some diacritics will not change despite the data being embedded. Thus, even an expert in the Arabic language, in the event of suspicion that the text contains hidden information, will be able to discover only a small number of the zeros and ones, through which they cannot decipher the secret information embedded within the text. It would also be impossible for automated analysis systems. However, one of the disadvantages of this method is the need to agree on a map of substitutions between the two parties, which may not be available in some circumstances. Therefore, another scenario was proposed to solve such cases.

Second scenario: The letter Alif (ا) in Arabic is written in more than one way, as in the following figure. In some places, this letter is written with a Hamza, i.e., decorated with a Hamza above or below. These Hamzas (in the Arabic language) have two types, so can either be called a disjunctive Hamza (هَمْزة القَطْع) or a conjunctive Hamza (هَمْزة الوَصْلِ). The conjunctive Hamza draws the letter Alif without a Hamza, while the disjunctive Hamza is drawn at the top (Figure 9).

Figure 9. Examples of Hamzas in the Arabic language.

The second scenario relies on the disjunctive Hamza (هَمْزة القَطْعِ) and conjunctive Hamza (هَمْزة الوَصْلِ). As many native speakers of the Arabic language make mistakes in writing some words that contain Hamza, the idea is similar to the first scenario. We will replace the disjunctive Hamza (ء) with (إ) with a Alif with Kasrah (إ) in the case of 1, or it will remain the same in the case of 0. The same method will be used on the upper Hamza also, changing (أ) into (ا), which is difficult even to notice by eye when reading the text, which is the most important goal of the science of concealment.

We note that in this model, the pattern is relied upon, and therefore there is no need for a table or map to be agreed upon between the sender and the receiver, but this leads to an increase in the number of characters in the new text compared to the old text, and as well as a lower level of security compared to the first scenario.

It is worth noting that a map can be employed in the second scenario and a pattern in the first scenario.

It is worth mentioning that in this scenario, a specific model was relied upon in the concealment process, and therefore there is no need for a table or map to be agreed upon between the sender and the receiver. However, using this scenario, the size of the new text will be larger than the old text because of the introduction of new characters. Additionally, the level of safety will be lower than the previous scenario. The map can be reused in the second scenario and the model can be reused in the first scenario.

Algorithms 1 and 2 depict the proposed scenarios (first and second Scenarios). For a deeper understanding of the above two scenarios, Figures 10 and 11 show an example of a practical implementation of the application we developed to ensure the effectiveness and achievability of the proposed approach.

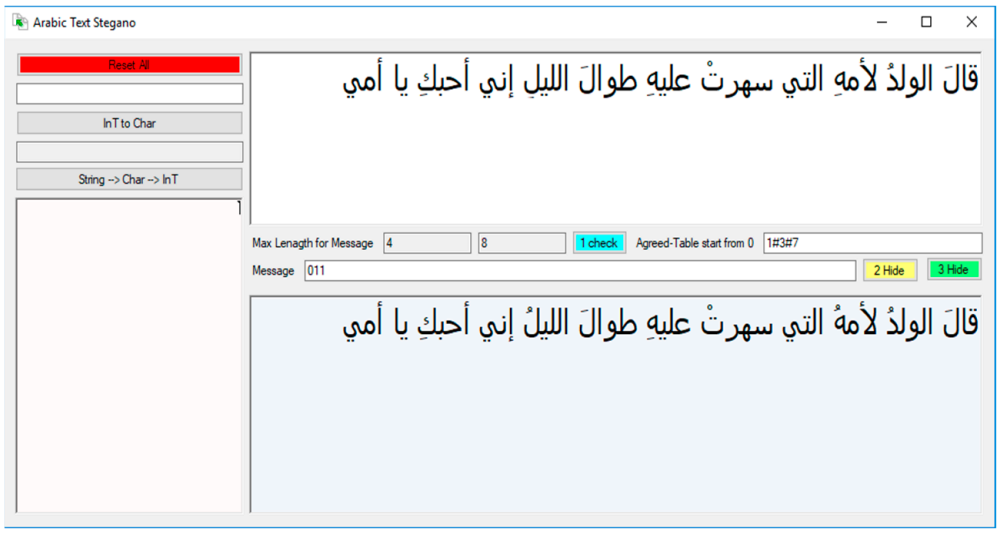

Figure 10. Sample implementation for the first scenario.

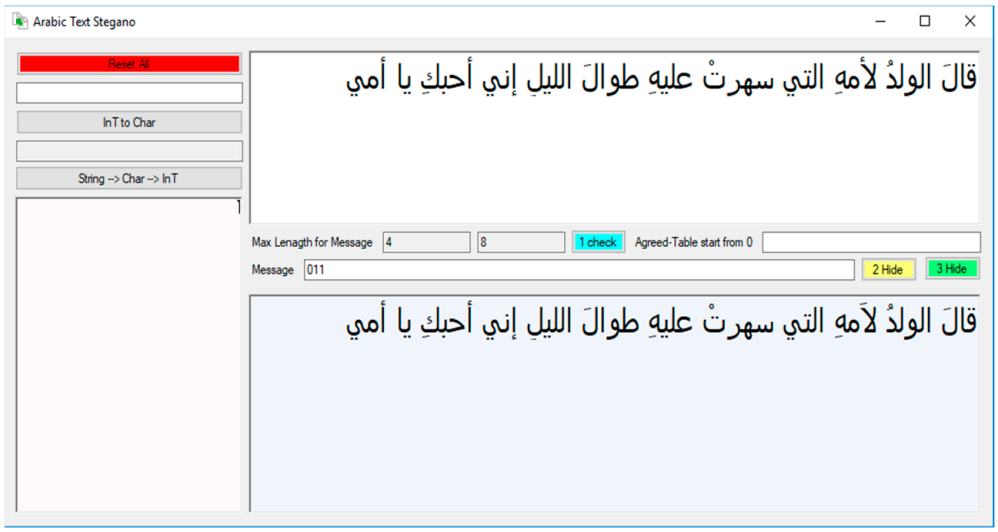

Figure 11. Sample implementation of the second scenario.

Assuming we have the following Arabic text as a cover for what we want to hide:

قالَ الولدُ لأمهِ التي سهرتْ عليهِ طوالَ الليلِ إني أحبكِ يا أمي

In the beginning, the application calculates the statistics of the cover text in order to know the threshold of embedding (in our case, the text contains eight diacritics and four Hamzas), meaning that the maximum number of bits that we can hide is 12 bits in this phrase, which is illustrated in Figures 10 and 11.

Assuming that the message to be hidden is 011 (this is the output of Phase 1), in the first proposed scenario, which involves concealing through diacritics, first we entered the table agreed upon between the two parties (concealing positions), which is provided in the example shown in Figure 10 (1#3#7). This means that the concealing will be performed in the first, the third, and the seventh diacritics.

Note that the diacritic that corresponds to the number 0 is replaced by the Fatha, and the diacritic that corresponds to the number 1 is replaced by the Dammah. Therefore, we did not change the vowel in the word "قالَ", but we changed it to a Dammah in the word "لأمهُ" and the word "الليلُ".

There is a condition that is tested before implementation, which is that the number of positions in the map agreed upon must be identical to the length of the message, and the last position in the map is smaller than the total number of diacritics in the text.

Similarly, in the second proposed scenario (where no hiding map is needed), the Hamza corresponding to 0 is swapped from the beginning, which is an upper Hamza "أ", with "ا" (Alif with Fatha), while the lower Hamza "إ" is replaced by "ا" (Alif with Kasrah), which is difficult to spot with the naked eye. We note in Figure 11 that only the Hamza in the word "لأمه" was replaced by an Alif with Fatha and became "لأمه". In addition, in order to increase the amount of data that can be concealed and the level of security, the two proposed scenarios can be combined.

Finally, Figure 12 shows the images in which the entire text is concealed (before and after), as there is no visible difference between them.

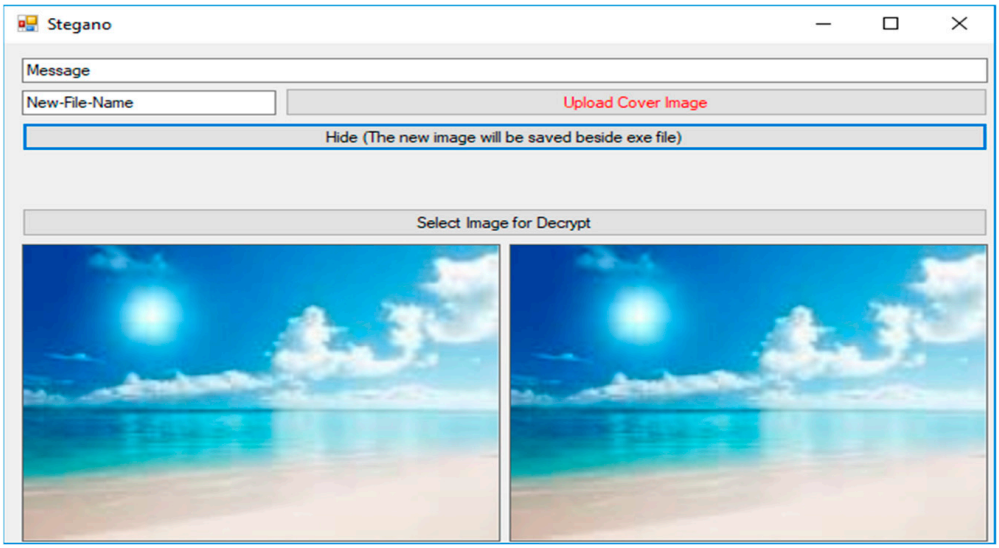

Figure 12. Sample implementation of Phase 4.

Phase 3: Encrypting the Cover

Here, we will encrypt the text resulting from the first and second stage, the same AES encryption algorithm can be used here, or any other encryption algorithm can be used to increase the level of protection.

Phase 4: Embed the encrypted data into the Cover

At this stage, we hide the encrypted text from the previous stage in a new cover (image) to ensure increased data confidentiality. In the case of a pre-agreed expanded map between the sender and the receiver (because the volume of data here is larger at this point), LSB would be an excellent solution, being easy to achieve and difficult to hack. In the absence of a map, we propose the idea of storage with a complex scenario that cannot be discovered while maintaining the quality of the cover. Based on the RGB image, the proposed system will store the first value in pixel number 8 in matrix R, the second in pixel number 9 in matrix G, the third in pixel number 10 in matrix B, and so on. Thus, it will become difficult to detect the encrypted message. Algorithm 1 presents the pseudocode of phase 4.

Algorithm 1 First Scenario Algorithm

String Encrypt_First_Scenario(string message, int [] Positions, String Key, String CoverText)

```
Start
Map [] = Positions [];
String Sec_msg = AES.Encrypt(message, Key);
Binary [] msg = Convert_To_Binary(Sec_Msg);
Int L1 = Length (msg);
Int L2 = Count (Map);
Int L3 = Find_Count ("ًٌ ّ ٍ", CoverText); // Number of all HARAKAT
Int Index = 0;
If (Map[End] <= L3)
    If (L1 == L2)
        While (Index < L1)
            Bit b = msg[Index];
            If (b)
                CoverText [ Map[Index] ] = "ٍ";
            Else
                CoverText [ Map[Index] ] = " ٌ ";
            End
            Index ++;
        End While
    Else
        return "Error Mapping";
    End IF
Else
    Return "Error Capacity";
End IF
Return CoverText;
```

End Function

Algorithm 2 Second Scenario Algorithm

String Encrypt_Second_Scenario(String message, String Key, String CoverText)

```
Start
String Sec_msg = AES.Encrypt(message, Key);
Binary [] msg = Convert_To_Binary(Sec_Msg);
Int L1 = Length (msg);
Int L2 = Find_Count (" ۡ ٔ", CoverText); // Number of all HAMAZAT
Int Index = 0;
If ( L1 < L2)
    While (Index < L1)
        Bit b = msg[Index];
        If (b)
            CoverText [Index] = " ٔ";
        Else
            CoverText [Index] = " ۡ";
        End
        Index ++;
    End While
Else
    return "Error Mapping";
End IF
Return CoverText;
End Function
```

4. Result and Discussion

4.1. Implementation

The following section presents the results of our experiments. All the proposed algorithms and scenarios were implemented using the Visual Studio Net 2019 environment with the C# programming language. Figures 10–12 show the execution of the proposed approach (the first scenario, second, and final phase of hiding in images).

In Table 4, we present the effectiveness and applicability of the proposed approach. We carried out the steps of the proposed approach based on a real example in all four stages. First, a key was used for the AES encryption algorithm, and since the key must be 16 characters in length, the MD5 algorithm was applied, which gives a fixed-length output of 16 bytes, whatever the length of the key.

Table 4. Example for implementing the four phases of the proposed approach.

Password	Key	We Padded the "Key" to 16 Characters by Zeros, or MD5 Can Be Used
Secret Message	"Message"	
Encrypted Message	U3YeVbnZ6vDUksOOZSoWng==	Note: Many algorithms can be used here RSA, AES, DES, etc.
Binary Message	00011000 01010101 00110011 01011001 01100101 01010110 01100010 01101110 01011010 00110110 01110110 01000100 01010101 01101011 01110011 01001111 01001111 01011010 01010011 01101111 01010111 01101110 01100111 00111101 00111101	Note: The first 8 bits represent the length of the message, which is 24 = 00011000 The next 8 bits represent the "U" character
Cover Text **Second Scenario we rely on the Hamza**	تتميّز اللّغة العربيّة عن كافّة اللّغات العالميّة الأُخرى بمجموعةٍ من الخصائص وهي: الأصوات: من المُميّزات الأساسيّة للّغة العربيّة؛ إذ يُعتَبر نظام النّطق فيها من أهمّ أنظمة الكلام اللغويّ، فيُستخدَم اللّسان، والحلق، والحنجرة من أجل نطق الحروف والكلمات بناءً على أصواتها، وتُقسَم الأصوات في اللّغة العربيّة إلى مجموعة من الأقسام، مثل أصوات الإطباق، وأصوات الحنجرة، وغيرها. المُفردات: هي الكلمات التي تتكوّن منها اللّغة العربيّة، ويُصنَّف المعجم اللغويّ الخاص فيها بأنّه من أكثر المعاجم اللغويّة الغنيّة بالمفردات والتّراكيب؛ فيحتوي على أكثر من مليون كلمة. وتُعتبر المفردات الأصليّة في اللّغة العربيّة عبارةً عن جذور ثلاثيّة للكلمات الأُخرى، فينتج الجذر اللغويّ الواحد العديد من الكلمات والمُفردات. كما أن اللفظ بالإعتماد على الحركات والصرف المرتبط بالمفردات وجزورها والنحو الذي يمثل أساس الجملة فيها ... كل ذلك مما يميزها عن باقي اللغات. لذلك تُصنّف اللّغة العربيّة كواحدةٍ من اللّغات التي تَحتفظ بنظامٍ نحويٍّ خاصّ بها، ويُساعد في إعراب جُملها وبيان طُرق كتابتها..	
Text after Hiding first 16 bits. **We changed 6 characters. (Green)**	تتميّز اللّغة العربيّة عن كافّة اللّغات العالميّة الأُخرى بمجموعةٍ من الخصائص وهي: الأصوات: من المُميّزات الأساسيّة للّغة العربيّة؛ إذ يُعتَبر نظام النّطق فيها من أهمّ أنظمة الكلام اللغويّ، فيُستخدَم اللّسان، والحلق، والحنجرة من أجل نطق الحروف والكلمات بناءً على أصواتها، وتُقسَم الأصوات في اللّغة العربيّة إلى مجموعة من الأقسام، مثل أصوات الإطباق، وأصوات الحنجرة، وغيرها. المُفردات: هي الكلمات التي تتكوّن منها اللّغة العربيّة، ويُصنَّف المعجم اللغويّ الخاص فيها بأنّه من أكثر المعاجم اللغويّة الغنيّة بالمفردات والتّراكيب؛ فيحتوي على أكثر من مليون كلمة. وتُعتبر المفردات الأصليّة في اللّغة العربيّة عبارةً عن جذور ثلاثيّة للكلمات الأُخرى، فينتج الجذر اللغويّ الواحد العديد من الكلمات والمُفردات. كما أن اللفظ بالإعتماد على الحركات والصرف المرتبط بالمفردات وجزورها والنحو الذي يمثل أساس الجملة فيها ... كل ذلك مما يميزها عن باقي اللغات. لذلك تُصنّف اللّغة العربيّة كواحدةٍ من اللّغات التي تَحتفظ بنظامٍ نحويٍّ خاصّ بها، ويُساعد في إعراب جُملها وبيان طُرق كتابتها.	

Table 4. *Cont.*

Password	Key	We Padded the "Key" to 16 Characters by Zeros, or MD5 Can Be Used
Encrypted text	0gO0UBikrTtP4eH0gY6QKjak5P5C5QnqTNEI8pwR8hWO/fB88MabKXayAEUNr2wuCRKVSTw XkqkBesLfB99AvJ1N059XLOKKcVo8sB66V8G2LZh9ou53RrRHqz1hNDSd8DSIMHFDbNWcUr VE9iGXsuwsTLltQUdymYLAKvCbGB+84MbFrVgkCN0B7/ouKqHthBH1+3lMzx0nJzDtt/5pPM5 jhT3MFBzE30XGbRclLBjBfW8eYXYdZHBJg9pqdD8vSTY2axH9ksNkY6zygslGHM/rlTPbuo0s1b aTg5ws3SUkoqC1RO4xihHSNyUvahX5iNsPtCANrZnCVRC5Rjx4py8LC67IqhDxmT+yFDBOtW MkF6O31LnfmrNv1DqTvuwiiTweBymyyLzz1SdkfbWsY9CQgoX/+83kCR1os0YaedToVvtffrbKg ej9OSVPVunpthqDVXf0DMvhDCHAnVfiQNAYXkhoODTP/pWBzBDta1eEtysfg4D4uGA9uwcq Gqohrfc2utuogNHIrTj/UfiWSFo9IM1vC3TJ1VVS/8Dt4gW7n9zHhlLsa10BRAXpwdFByjnPjua5n vI1kDLH/FS9S7O94p6CMAJuHWt9bYKMxFaW41zecnmTAwgbtGnIWe33TbrZ/iavv30Q9+7ZTS tI5KZlLqw8lugy9hk7jyOT2TL5S8bLF/0TKgoIi0AOztjBHnruglBeMK/ziovHkNR6xDak5P5C5Qn qTNEI8pwR8hWkTzcQZn1gEWPzypkjDM3iZi57j7TxNa8QJMjHJJdMB1FNx5MqN1pU5EDHu9 QpFwFel5UKY67bD4OQdEd+mQgBTahTRzrMWFJTi/zx7VQr2uNGBzXBP140KMEQCp3JUfTN 6vRRsD7C1El3O5+ZrvwJQa4EWqpZ71T/91FnVOZfUQBLLMbErO/QiQoY0ahQR5+coKkgyqIV9 LtS3qIS3rAPlKxGSIAXLwVs93oU+18Yhw+okLVuGCFFLTenT5m9Q8Uf2Dl+M1In+7aSlmFnJRp eKXf12KrnwBusoGzckKnufKnsnY2tAZ0J5CV33B9ZK/NBiurNvSShmetLuR0nJ4hRB5fXvdiqUg9 +ehuTE5VKmx+N4UugJ9Q0IeChmWttonBVTIkcmeLS93OhNJ1M7uRdeldGHdt8w/icIgj/JR8ko5 Y209Vl51YMN/RBVbKxesey7LWLb1/YdiaxfBvdY4rNQzD2QcWieUAvxiLDS3cdHzofRQv6m1C nSX/X1pCVIXJjcwlghf7EFLx+ks7Pdn00B3xGZVACMKI/LesnwUF3KWvbuHMucbJkJ2uzGQHG QTyCCYpqNNkuXVr4jjwt2k/DHvn9vES5zXGJQC+yfS4f6OR5/UtmlreutWfXMKmPg7iwU658M duhpxwkAS2lDH4WSgQRLBf5N/C0h78WgecjdWVYbRajw267qvGKsZzJqtPOFCK2pHusqzLjc+ /hHAuZiht7hUzTDEWxjasvXNQpbsfEsyP4YeXpneEB0pqRom6khskO1aikwblh9tOr+lIrEcNaf5IJ hxaxPqxD3HRF/PzubE3CG/ExpyOrrJyLvhetSr1f9+jLG5LGbAovkdqCmWOy0f7DUcT8wNvIny 3vtixACqlsr4xYuW/37qInkojPcbKLSoCjzhKQTDzklnkxy1TOSlwsx2nbnJELu+ZrO6CHdxOYCR CmDD8hQm5ibyFKXjKbhpkYYWRrdNDIkWXaI9q2sWmZzgTT9tLgCjcLZlgYN1Ir907Ea89t/1P eIKvCAqsRnbIQ242Yfh6imnXCcbHM9Hj4ilwyF9a0BhuWgUMvytPeA1jqNnfs+hywOPMiYxVIc uxG7fWTiIYGoIi9mlnqSThMaAE1RXXq2mr4nSnlA2N6aakJuvUoAfXoxBRM31OIKMF2o9P55X hS2EoFfkeybfFGPFxPZTZL0x4TroUzVUuAiun0lFQJU6nIC3gRdurSJ3DkbJi/P9gwX57+RDeWA VioYNxVdQp7MHiE5004eaNG7aP4AmzH1c+cJKF4uLZVAawnf+9jrXbrN/a8IayhNb/IOtzbvWh u69DOwaUwr4qFx+HzHlEwczoUoxMQb+oc1FIQ4xssaArape7dF99LkXLomxFibSudS1ToJpXHv UalvEW3ZI1LkRhFV7JcTdo+9GjG/lWR4/kwtS1MhBkZ2Mi214YSpafiE8ovnCmy37COTeeX+QcY RneNg0NPSh4pYqFmy5YZm6Kneq3eZDUJiktC2qMbSImQiors5E97lL+wvbMsrbG8vr6e9NCMy U1aXIkO9XpiUBqKjMvnVCYmxx/jDi37V1rMrB/XA+UWVYNPNhjzbEMk9GDCX1engVT3jWFs To4Hed5/Acw5PTlsGrDoiZzFhHxw5t6QzQ==
Size of required cover	2380 chars × 8 = 19,040 pixels will be changed Image 150 × 130 = 19,500 Pixels with three layers for RGB	
Cover Image		**Image after Hiding** 

We assumed that the message to be hidden is the word "Message". The message was initially encrypted using the AES algorithm, with the result being "U3YeVbnZ6vDUksOOZ

SoWng==". We note that the new message length is 24 characters instead of 7 before the encryption. In the second stage, we will hide the encrypted message within the Arabic text. In the first place, the length of the message must be embedded, and then the letters of the message. Therefore, we inserted the number 24 after representing it in binary "00011000", and then we inserted the ASCII code for the letter "U" (01010101) and then the letter "3" and so on until the end of the message.

The steganography is shown only for the length of the message and the first letter of the message (the length = 24, the first letter is U) within the Arabic script. We note that the Arabic text must contain several Hamzas greater than the number of bits to be inserted. After completing the concealed process, we colored the characters that are changed, which corresponded to a bit value of 1 only in the text, in green, and the other characters remained in red because they did not change, as they corresponded to a bit value of 0. After completing the complete concealing of the message, we applied the third stage, which is the encryption of the new Arabic text after concealment within it, using the encryption algorithm (using the same agreed key). The output will be full ciphertext (as shown in Table 4).

Finally, we concealed the large ciphertext within the RGB image according to Algorithm 3 within the proposed approach.

For the purposes of checking the quality of any steganography approach, usually we use the correlation between the result image (after insertion) and the cover image (before insertion) by calculating the similarity (a good method is one that gives higher similarity and a lower error rate). Additionally, the mean squared error (MSE) and bit error rate (BER) are common metrics to find the rate of changes after hiding and rate of errors [11]. The equations of the selected metrics are presented in (1), (2), and (3). Moreover, the implementation of these metrics is depicted in Figures 13 and 14 and Table 5.

Table 5. Metrics results of the double method.

Correlation	MSE	BER
0.994	0.005	0.002

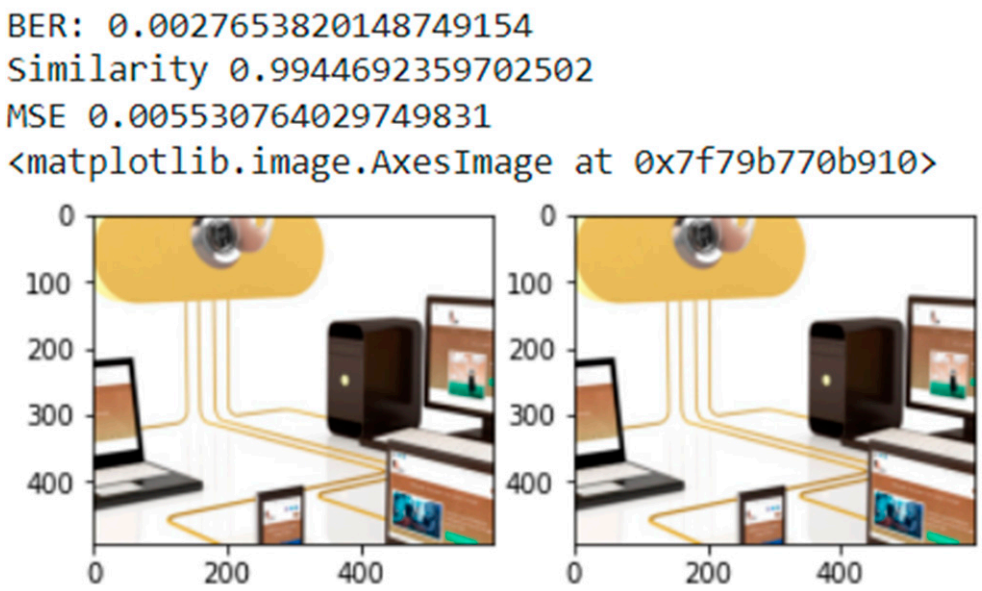

Figure 13. Result of a python code to calculate the similarity, MSE, and BER.

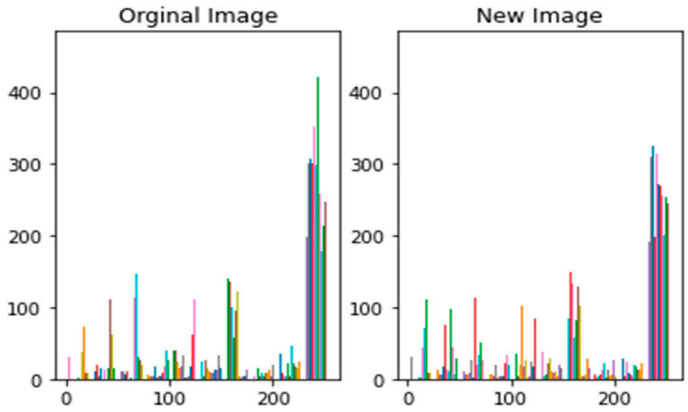

Figure 14. Comparison between the histogram of the original and new image after hiding.

$$Correlation = \frac{Number\ of\ marching\ pixels}{Total\ number\ of\ pixels} \tag{1}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}[C(ij) - N(ij)]^2 \tag{2}$$

where n is the number of rows, m is the number of columns, and C and N represent the cover and new images.

$$BER = \frac{Number\ of\ errors}{Total\ number\ of\ bits} \tag{3}$$

Figure 15 shows the *MSE* (Equation (2)) in the resulting image compared to the original image before embedding. When the data size changes (number of bits), it is noticed that with the gradual increase in the size of the data, the error rate increases, but it continues to have a small margin, which is an indicator that confirms the efficacy of the method used and explains the small value of the error due to the modification of the least important bit in the color matrix, only one for each pixel.

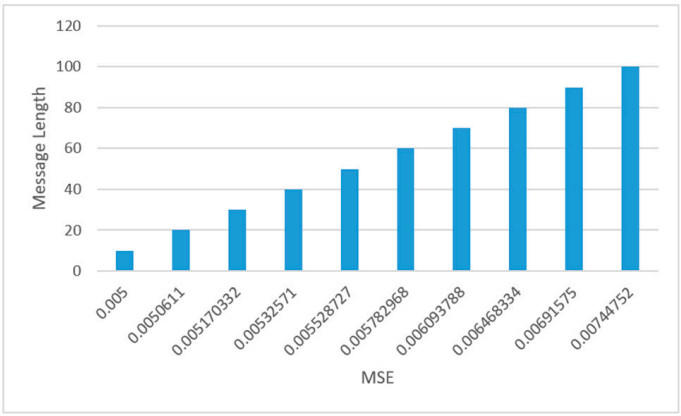

Figure 15. MSE value vs. Message length.

Figure 16 shows the percentage of correlation between the original image and the resulting image after pasting with the change in the size of the data that is pasted. The percentage of similarity or correlation is calculated through equation number (1), and the higher it is, the better indicator. However, embedding in more data means an increase in the number of variable pixels and thus a decrease in the similarity value from 1.

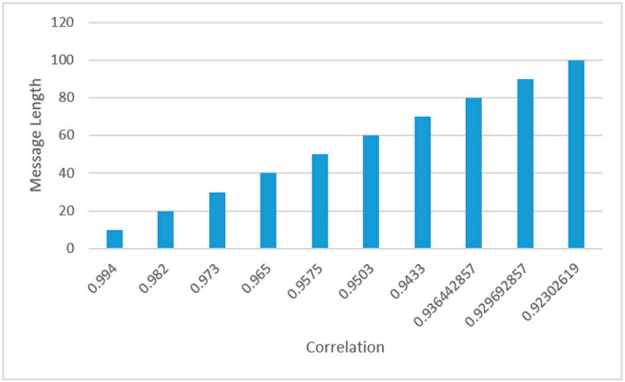

Figure 16. Correlation value vs. message length.

4.2. The Metrics of Evaluating Steganography

Steganography is a technique that fits into a broader category of hiding confidential information inside various forms of media. This method offers several benefits, including the ability to store large amounts of data, a high level of capacity, security, imperceptibility, and robustness [50].

4.2.1. Imperceptibility

Steganography is a method used to hide information in a way that is difficult to detect or predict. The success of steganography is usually measured by its imperceptibility, which refers to the ability of an attacker to locate or predict the hidden information [51]. In the case of concealing text, achieving imperceptibility requires the use of cover texts that are very large. However, using text alone does not always guarantee imperceptibility, especially when concealing large amounts of data. Most methods for handling Arabic texts rely on the principle of modifying elements within the text. If these modifications are made in certain locations, the text's reader may not detect them or assume them to be inadvertent errors. Nonetheless, if the errors are repeated in a specific or excessive manner within the text, the reader may become suspicious that something is being hidden. To ensure imperceptibility in this situation, an additional cover, such as an image, may be used to conceal the entire text.

In terms of imperceptibility, the scenario referred to as the "Hamza scenario", where the entire text comprises movements, may be more effective than the scenario of hiding and revealing diacritics. This is because in the Arabic language, it is more likely that an expert would notice an incorrect diacritic than detect a hidden diacritic, especially when it concerns a Hamza.

When considering situations where the cover text entirely consists of diacritics and the amount of secret data is limited, it is prudent to select the diacritic scenario as the preferred option. This preference arises from the difficulty in detecting any modifications made to diacritics embedded within fully formed text, particularly when the changes occur within the interior of words. Incorporating diacritics into such scenarios is therefore likely to enhance the robustness and effectiveness of the text steganography technique, leading to an improved quality of secret data transmission.

4.2.2. Capacity

The maximum amount of data that can be hidden within a cover represents another crucial aspect of steganography [47]. In the case of the Arabic language, there are two proposed scenarios: the Hamza and diacritic scenarios. The capacity for the Hamza scenario is limited as it relies on the natural distribution of Hamzas within Arabic text. While Alif is the most common letter, it only accounts for up to 10% of the text, resulting in a relatively small amount of information that can be stored, which is also dependent on the size of the cover text. On the other hand, the diacritic scenario offers greater capacity, as one is able

to hide data in all letters of text by decorating letters with diacritics, which represents the maximum capacity for text-based methods. In terms of images, techniques based on the LSB principle can typically conceal up to 12.5% of the image size (number of pixels), which is adequate considering the considerable size of digital images.

The following equations represent the relationship between the maximum number of bits that can be embedded and the corresponding factors in each scenario. We represent the maximum number of bits that can be hidden as "M".

Diacritic Scenario: The number of diacritics that can be used in the text is represented as "D". The equation for Scenario 1 can be written as:

$$M = D$$

Hamza Scenario: The number of Hamzas used in the text is represented as "H". The equation for Scenario 2 can be written as:

$$M = H$$

Hybrid Scenario: The total maximum number of bits that can be embedded in the hybrid scenario is equal to the sum of the maximum number of bits for Scenario 1 and Scenario 2. The equation for the hybrid scenario can be written as:

$$M = D + H$$

4.2.3. Robustness

In the context of steganography, the term "robustness" refers to the text's ability to maintain the confidentiality of hidden information even when subjected to modifications such as compression [52]. While this is a crucial aspect of traditional text-based steganography methods, it is of lesser significance in the context of image steganography. In situations where transmission or reception distortions occur—whether intentionally or not—the text may become distorted rather than detected, rendering robustness moot. However, the proposed concept addresses this issue by Unicode encoding for the text and its diacritics. Any changes in the text's diacritics occur at the Unicode level, which ensures that any compression or conversion processes applied to the text cover and the hidden text will not affect its robustness, resulting in a high level of robustness that is immune to distortion.

In the context of image evaluation, it is widely acknowledged that employing techniques such as image modification or compression can exert a substantial influence on the evaluation criterion, particularly when the LSB method is utilized. Consequently, to mitigate the potential negative effects of these techniques, it is recommended to utilize frequency domain-based methods as an alternative to spatial methods. Such methods offer the potential to enhance the accuracy and reliability of the evaluation process, thereby enabling more robust and effective image analysis.

4.2.4. Security

Security is the art of concealing secrets within a text, that it may remain hidden from any intruder's eyes, be it by mathematical, statistical, or visual means [53]. To enhance the level of security in our proposed approach, we implement several measures. Firstly, encryption is introduced as a fundamental step before the jamming process. An attacker who detects changes in the text and reveals some bits will not be able to reassemble all the jammed bits, and henceforth, will not be able to decrypt the collected text and benefit from it. To evaluate the effectiveness of the proposed approach, a set of experiments is conducted. The goal of these experiments is to demonstrate the efficiency of the new approach by measuring the characteristics outlined in the above paragraph.

4.3. Experimental and Investigations on the Proposed Method Based on the Steganography Metrics

4.3.1. Hypotheses

- First, Table 6 displays the details of the transcripts utilized in our experiments. Four different texts of lengths and properties were employed.
- Secondly, the secrets were randomly selected and comprise a combination of binary digits (zeros and ones).
- Thirdly, the proposed approach was evaluated by a team of five experts using the five-point Delphi method, where decisions were made based on the majority agreement among the team.

Table 6. Description of tests used in the experiments.

#	Description	Size (Characters)	Number of Hamzas	Number of Diacritics	Secret msg Size (Bits)	Scenario 1 No. Changes	Scenario 2 No. Changes
T1	Small Text without Diacritics	120	20	0	10	4	10
T2	Small Text with some Random Diacritics	719	17	21	10	5	11
T3	Medium Size text	2500	59	1200	50	19	30
T4	relatively long	15,000	310	7000	200	56	123

4.3.2. Capacity Feature

Based on Table 6, it is evident that the scenario of diacritics surpasses the scenario of Hamzas significantly in terms of capacity. This is because diacritics can be added to any letter, allowing for the entire text to be utilized (maximum capacity). However, it should be noted that the Hamza scenario has a limited capacity, since Hamzas only constitute approximately 2–3% of the text, making this scenario suitable only for short messages.

4.3.3. Security and Imperceptibility Features

Regarding the first text, in the Hamza scenario, only 6 Hamzas were altered by adding a diacritic, whereas 14 characters were modified in the scenario, equivalent to the length of the secret message. As a result, the secret was easily detected by experts in the case of diacritics, whereas only part of the message was discovered in the Hamza scenario, specifically the number of changes, which was four. However, it is possible to reveal the complete message in both cases with certain software tests. In terms of the imperceptibility criterion, both scenarios are weak, and thus, we recommend employing the third and fourth layers of protection, which involve encrypting the small piece of text and encapsulating it within an image to enhance the level of security and imperceptibility (Figure 17).

The second text in the diacritic scenario already contains random diacritics. However, after being hidden in specific locations according to the predefined MAP, 5 out of the 10 diacritics (the length of the secret message) were modified. This number represents only a portion of the message that experts were able to discover, rather than the entire message. Some experts believed that all movements comprised the message, resulting in the discovery of 21 diacritics, which is longer than the message itself. In both scenarios, experts were unable to decrypt the message due to incomplete message bits. Nevertheless, the encrypted secret message can be accessed by using a brute-force algorithm. Similarly, in the Hamza scenario, only modified Hamzas were detected, totaling five. As the number of Hamzas is less than the number of diacritics in the text, using the brute-force algorithm would be less costly for attackers. Regarding the imperceptibility criterion, it is inadequate in both scenarios. Therefore, we recommend using the third and fourth layer of protection by encrypting and encapsulating the entire small piece of text within an image. This approach increases the level of security and imperceptibility simultaneously.

Figure 17. (Main, (**a**,**b**)): the average time in minutes of secret detection using phase 2.

The third text was fully constructed, and subsequently, a longer message was concealed within it. The changes made involved altering 30 diacritics within the diacritic scenario and 19 in the Hamza scenario. Despite the secret being 50 characters long, the number of changes made was lower. The experts required greater focus and time to identify the specific locations of the changes and had to review the text multiple times. This process was particularly challenging due to the problematic nature of the entire text, making it difficult to precisely identify the necessary changes. The average time spent by the experts reviewing the text further corroborates this observation (Figure 17—Main). Thus, the imperceptibility criterion can be deemed moderately successful in this case, particularly with a short secret message length, as the text would not arouse suspicion, given the simplicity of the errors, which are often present in some Arabic texts. However, for longer secret messages and a higher number of diacritical errors, the Hamza scenario might be more effective than the diacritical scenario. In the present case, numerous diacritical errors are evident, which would be noticeable to an expert reader of the Arabic language. Consequently, when dealing with longer secret messages, we must either significantly increase the length of the cover text or rely on the embedding layer within the image for additional protection. In such cases, steganography analysis would be extremely challenging, since the detector would only have access to a small fraction of the message in both scenarios. Automated testing for all cases would be very expensive, particularly in the diacritical scenario, and the difficulty would only increase with the length of the text. Therefore, the proposed algorithm for texts could be deemed effective in such cases without the need for an image layer.

The fourth text is comprehensively formed and surpasses its predecessors in length. Nonetheless, the outcomes it yields completely validate the results of the previous text, albeit with additional time and effort devoted (Figure 17a,b). The larger size of the text

considerably elevates the level of security and intensifies the complexity of penetrating and cracking the encrypted message. Consequently, employing an image layer may be unnecessary in such scenarios. It is imperative to note that all previous experiments discussed the process of decrypting the encrypted secret message, not the underlying message. Therefore, the attacker must also decrypt the message, which substantiates the robustness of the proposed algorithm. Regarding the criterion of robustness, text-based algorithms typically exhibit higher efficiency and immunity than their image-based counterparts. This is because most image compression algorithms lose a portion of the data, necessitating the use of image-based steganography techniques based on the frequency domain, as opposed to those based on the spatial domain, such as the least significant bit (LSB) method.

Algorithm 3 for Steganography layers selection.

Input (cover + secret)
Output (hidden secret)
Step1: Determine the scenario of choice for hiding secret messages: Diacritics or Hamzas
Step2: Calculate the Capacity to determine which scenario has the highest capacity.
Step3: Apply Testing for Crypto Analysis and Imperceptibility Criterion to evaluate the effectiveness of the chosen scenario.

1. If the chosen scenario is inadequate in terms of imperceptibility:

Then, employ the third and fourth layers of protection by encrypting and encapsulating the entire small text within an image. Conceal the secret message within the chosen scenario by modifying the necessary characters according to the predefined MAP.

2. If the secret message is longer

Then, significantly increase the length of the cover text or rely on the embedding layer within the image for additional protection.

3. If necessary, employ an image layer for additional security in larger scenarios.
4. Use text-based algorithms for higher efficiency and immunity. If image-based algorithms are used, embedding techniques are based on the frequency domain.

4.4. Comparison between the Proposed Method and the Other Methods Based on the Steganography Metrics

The comparison of different steganography approaches provides valuable insights into their performance in terms of imperceptibility, capacity, robustness, and security. By examining the results presented in the Table 7, we can gain a deeper understanding of the strengths and weaknesses of each approach, leading us to identify the most effective technique for achieving optimal results in steganographic applications.

Upon careful analysis of the provided table, it becomes evident that the "Diacritics and Hamzas" approach, denoted as "our approach", excels across all evaluated criteria, namely imperceptibility, capacity, robustness, and security. Among the listed approaches, the hybrid technique demonstrates high imperceptibility and capacity, but specific details are lacking. The Kashida approach appears to achieve high imperceptibility and capacity in several instances, but robustness and security information remain undisclosed or unknown. Similarly, the diacritics approach shows high imperceptibility and capacity, but lacks explicit information about robustness and security. The combination of Kashida and Unicode and the use of shape both have insufficient information regarding their effectiveness. The Unicode approach exhibits high imperceptibility and capacity, while the robustness and security aspects remain unspecified. In contrast, the "Diacritics + Hamzas" approach, characterized as our approach, achieves high ratings across all four dimensions, making it the most comprehensive and reliable choice for ensuring superior imperceptibility, capacity, robustness, and security in the context of the given table.

Table 7. Comparison with other approaches.

Reference #	Approach Used	Imperceptibility	Capacity	Robustness	Security
[17]	Hybrid	high	high	high	high
[29]	Kashida	unknown	high	unknown	unknown
[32]	Diacritics	unknown	high	unknown	unknown
[54]	Kashida	unknown	high	unknown	unknown
[30]	Hybrid	high	unknown	unknown	unknown
[31]	Kashida	unknown	high	unknown	high
[33]	Kashida	unknown	high	unknown	unknown
[34]	Kashida	unknown	high	unknown	unknown
[35]	Kashida and Unicode	unknown	unknown	unknown	unknown
[36]	Kashida	unknown	unknown	unknown	unknown
[37]	shape	unknown	high	unknown	unknown
[55]	Kashida	unknown	high	unknown	unknown
[38]	Unicode	unknown	high	unknown	unknown
[39]	Unicode	unknown	high	unknown	unknown
[40]	Unicode	unknown	high	unknown	unknown
[41]	Unicode	unknown	high	unknown	unknown
[42]	Unicode	unknown	unknown	unknown	unknown
[43]	Unicode	high	high		unknown
[44]	Diacritics	unknown	high	unknown	unknown
[45]	Diacritics	unknown	high	unknown	unknown
[56]	Diacritics	unknown	high	unknown	unknown
[46]	Diacritics	unknown	high	unknown	unknown
[47]	Diacritics	unknown	high	unknown	unknown
[48]	Diacritics	Low	high	unknown	unknown
[4]	Diacritics	unknown	high	high	high
our approach	**Diacritics and Hamzas**	**high**	**high**	**high**	**high**

5. Conclusions

In conclusion, this paper presents a novel and robust steganography algorithm that utilizes the limited-sized diacritics of Arabic letters and Hamzas to embed confidential information effectively. The proposed approach outperforms previous steganography solutions by leveraging the inherent properties of the Arabic language, such as the variability of diacritics and the different ways to write the Arabic letter Alif with Hamza. Furthermore, the algorithm makes the hidden text almost impossible to detect by employing a multi-layered technique that includes encryption, cover concealment, and cover hiding.

This paper introduces two embedding scenarios for the encrypted data: one based on the diacritics used in the Arabic language and the other based on the different ways to write the Arabic letter Alif with Hamzas. The scenario based on diacritics outperforms the Hamza scenario in terms of capacity, as diacritics can be added to any letter in the text. However, both scenarios have weaknesses in imperceptibility, and additional layers of protection, such as encryption and image encapsulation, are recommended to enhance security.

The proposed algorithm significantly enhances the capacity and robustness of the hidden text while diversifying the hiding scenarios. However, a disadvantage of this method is the need to establish a substitution map between the two parties in the first scenario. Therefore, this research proposes a new solution that can potentially aid in military and other highly confidential applications. Furthermore, experiments show that the proposed algorithm has a relatively large secret capacity, but the complexity of cracking the message increases with the length of the message. Therefore, the proposed text-based algorithm can be deemed effective without needing an image layer, and it is also robust against attacks, as the attacker must also decrypt the message.

Overall, five experts have thoroughly investigated and evaluated the proposed steganography algorithm based on Arabic diacritics and Hamzas using the Delphi method. The results demonstrate its effectiveness and potential for use in military and other highly

confidential applications while highlighting areas for improvement to enhance its security and imperceptibility.

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

Funding: This project is funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R383), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was supported through the Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R383) provided through Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

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

References

1. Al-Shatnawi, A.M. A New Method in Image Steganography with Improved Image Quality. *Appl. Math. Sci.* **2013**, *6*, 3907–3915.
2. Swanson, M.D.; Kobayashi, M.; Tewfik, A.H. Multimedia Data-Embedding and Watermarking Technologies. *Proc. IEEE* **1998**, *86*, 1064–1087. [CrossRef]
3. Kaçar, S.; Konyar, M.Z.; Çavuşoğlu, Ü. 4D Chaotic System-Based Secure Data Hiding Method to Improve Robustness and Embedding Capacity of Videos. *J. Inf. Secur. Appl.* **2022**, *71*, 103369. [CrossRef]
4. Alifah Roslan, N.; Izura Udzir, N.; Mahmod, R.; Gutub, A. Systematic Literature Review and Analysis for Arabic Text Steganography Method Practically. *Egypt. Inform. J.* **2022**, *23*, 177–191. [CrossRef]
5. Duals. The 10 Most Word-Rich Languages in the World You Should Know about. Available online: https://blog.duals.app/the-10-most-word-rich-languages-in-the-world-you-should-know-about-cece2b91942e (accessed on 4 June 2023).
6. Hussein, A. The Efforts of Al-Khalil Bin Ahmad Al-Farahidi and His Views on the Science of Dots through the Book The Decisiveness of Punctuation in Quran by Abi Amr Al-Dani: Exploration and Analysis. *Islam. Sci. J.* **2023**, *14*, 152–174. [CrossRef]
7. Alanazi, N.; Khan, E.; Gutub, A. Efficient Security and Capacity Techniques for Arabic Text Steganography via Engaging Unicode Standard Encoding. *Multimed. Tools Appl.* **2021**, *80*, 1403–1431. [CrossRef]
8. Divya, A.; Thenmozhi, S. Steganography: Various Techniques In Spatial and Transform Domain. *Int. J. Adv. Sci. Res. Manag.* **2016**, *1*, 81–89.
9. Das, P.; Kushwaha, S.C.; Chakraborty, M. Data Hiding Using Randomization and Multiple Encrypted Secret Images. In Proceedings of the 2015 International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, India, 2–4 April 2015; pp. 298–302.
10. Chan, C.-K.; Cheng, L.M. Hiding Data in Images by Simple LSB Substitution. *Pattern Recognit.* **2004**, *37*, 469–474. [CrossRef]
11. Nosrati, M.; Hanani, A.; Karimi, R. Steganography in Image Segments Using Genetic Algorithm. In Proceedings of the 2015 Fifth International Conference on Advanced Computing & Communication Technologies, Haryana, India, 21–22 February 2015; pp. 102–107.
12. Tahir, A.; Amit, D. A Novel Approach of LSB Based Steganography Using Parity Checker. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2015**, *5*, 314–321.
13. Luo, W.; Huang, F.; Huang, J. Edge Adaptive Image Steganography Based on LSB Matching Revisited. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 201–214. [CrossRef]
14. Basahel, A.; Yamin, M.; Ahmed, A.; Abi Sen, A. Enhancing Security of Transmitted Data by Improved Steganography Method. *Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 239–244.
15. Wu, D.-C.; Tsai, W.-H. A Steganographic Method for Images by Pixel-Value Differencing. *Pattern Recognit. Lett.* **2003**, *24*, 1613–1626. [CrossRef]
16. Afrakhteh, M.; Ibrahim, S. Adaptive Steganography Scheme Using More Surrounding Pixels. In Proceedings of the 2010 International Conference on Computer Design and Applications, Qinhuangdao, China, 25–27 June 2010; Volume 1, pp. 225–229.

17. Kadhem, S. Improving Hiding Security of Arabic Text Steganography by Hybrid AES Cryptography and Text Steganography. *J. Eng. Res. Appl.* **2016**, *6*, 60–69.
18. Potdar, V.M.; Chang, E. Grey Level Modification Steganography for Secret Communication. In Proceedings of the 2nd IEEE International Conference on Industrial Informatics, 2004, INDIN '04, Berlin, Germany, 24–26 June 2004; pp. 223–228.
19. Al-Rahal, M.S.; Sen, A.A.; Basuhil, A.A. High Level Security Based Steganoraphy in Image and Audio Files. *J. Theor. Applies Inf. Technol.* **2016**, *87*, 29–37.
20. Li, X.; Zhang, W.; Gui, X.; Yang, B. A Novel Reversible Data Hiding Scheme Based on Two-Dimensional Difference-Histogram Modification. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1091–1100. [CrossRef]
21. Alsaawy, Y.; Abi Sen, A.A.; Alkhodre, A.; Bahbouh, N.M.; Baghanim, N.A.; Alharbi, H.B. Double Steganography—New Algorithm for More Security. In Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021; pp. 370–374.
22. Divya, V.; Sasirekha, N. High Capacity Steganography Technique Based on Wavelet Transform. In Proceedings of the 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, India, 19 November 2016; pp. 1–5.
23. Kumar, V.; Kumar, D. A Modified DWT-Based Image Steganography Technique. *Multimed. Tools Appl.* **2018**, *77*, 13279–13308. [CrossRef]
24. El-Khamy, S.E.; Korany, N.; El-Sherif, M.H. Robust Image Hiding in Audio Based on Integer Wavelet Transform and Chaotic Maps Hopping. In Proceedings of the 2017 34th National Radio Science Conference (NRSC), Alexandria, Egypt, 13–16 March 2017; pp. 205–212.
25. Shafi, I.; Noman, M.; Gohar, M.; Ahmad, A.; Khan, M.; Din, S.; Ahmad, S.H.; Ahmad, J. An Adaptive Hybrid Fuzzy-Wavelet Approach for Image Steganography Using Bit Reduction and Pixel Adjustment. *Soft Comput.* **2018**, *22*, 1555–1567. [CrossRef]
26. Sharifzadeh, F.; Akbarizadeh, G.; Seifi Kavian, Y. Ship Classification in SAR Images Using a New Hybrid CNN–MLP Classifier. *J. Indian Soc. Remote Sens.* **2019**, *47*, 551–562. [CrossRef]
27. Durafe, A.; Patidar, V. Development and Analysis of IWT-SVD and DWT-SVD Steganography Using Fractal Cover. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 4483–4498. [CrossRef]
28. Gutub, A.; Fattani, M. A Novel Arabic Text Steganography Method Using Letter Points and Extensions. In Proceedings of the WASET International Conference on Computer, Information and Systems Science and Engineering (ICCISSE), Vienna, Austria, 25–27 May 2007; pp. 28–31.
29. Abdul-Aziz Gutub, A.; Al-Nazer, A.A. High Capacity Steganography Tool for Arabic Text Using 'Kashida'. *ISeCure* **2010**, *2*, 107–118.
30. Gutub, A.; Al-Juaid, N.; Khan, E. Counting-Based Secret Sharing Technique for Multimedia Applications. *Multimed. Tools Appl.* **2019**, *78*, 5591–5619. [CrossRef]
31. Al-Nofaie, S.; Gutub, A.; Al-Ghamdi, M. Enhancing Arabic Text Steganography for Personal Usage Utilizing Pseudo-Spaces. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 963–974. [CrossRef]
32. Odeh, A.; Elleithy, K.; Faezipour, M. Steganography in Arabic Text Using Kashida Variation Algorithm (KVA). In Proceedings of the 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 3 May 2013; pp. 1–6.
33. Al-Nofaie, S.M.A.; Gutub, A.A.-A. Utilizing Pseudo-Spaces to Improve Arabic Text Steganography for Multimedia Data Communications. *Multimed. Tools Appl.* **2020**, *79*, 19–67. [CrossRef]
34. Al-Alwani, W.; Mahfooz, A.B.; Gutub, A.A.-A. A Novel Arabic Text Steganography Method Using Extensions. *World Acad. Sci. Eng. Technol.* **2007**, 83–86.
35. Odeh, A.; Elleithy, K. Steganography in Arabic Text Using Zero Width and Kashidha Letters. *Int. J. Comput. Sci. Inf. Technol. IJCSIT* **2012**, *4*, 1–11. [CrossRef]
36. Roslan, N.A.; Mahmod, R.; Udzir, N.I. Sharp-Edges Method in Arabic Text Steganography. *J. Theor. Appl. Inf. Technol.* **2011**, *33*, 32–141.
37. Roslan, N.A.; Mahmod, R.; Udzir, N.I.; Zurkarnain, Z.A. Primitive structural method for high capacity text steganography. *J. Theor. Appl. Inf. Technol.* **2014**, *67*, 373–383.
38. Ali, A.E. A New Text Steganography Method by Using Non-Printing Unicode Characters. *Eng. Technol. J.* **2010**, *28*, 72–83.
39. Mohamed, A.A. An Improved Algorithm for Information Hiding Based on Features of Arabic Text: A Unicode Approach. *Egypt. Inform. J.* **2014**, *15*, 79–87. [CrossRef]
40. Obeidat, A. Arabic Text Steganography Using Unicode of Non-Joined to Right Side. *J. Comput. Sci.* **2017**, *13*, 184–191. [CrossRef]
41. Khami, M.J. Unicode with Rules Arabic Text Data Hiding. *J. Educ. Pure Sci.* **2018**, *8*, 52–76.
42. Ditta, A.; Azeem, M.; Naseem, S.; Gulzar Rana, K.; Adnan Khan, M.; Iqbal, Z. A Secure and Size Efficient Algorithm to Enhance Data Hiding Capacity and Security of Cover Text by Using Unicode. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 2180–2191. [CrossRef]
43. Aabed, M.A.; Awaideh, S.M.; Elshafei, A.-R.M.; Gutub, A.A. Arabic Diacritics Based Steganography. In Proceedings of the 2007 IEEE International Conference on Signal Processing and Communications, Dubai, United Arab Emirates, 25–27 November 2007; pp. 756–759.
44. Gutub, A.; Elarian, Y.; Awaideh, S.; Alvi, A. *Arabic Text Steganography Using Multiple Diacritics*; University of Sharjah: Sharjah, United Arab Emirates, 2008.

45. Bensaad, M.L.; Yagoubi, M.B. Boosting the Capacity of Diacritics-Based Methods for Information Hiding in Arabic Text. *Arab. J. Sci. Eng.* **2013**, *38*, 2035–2041. [CrossRef]
46. Gutub, A.; Ghouti, L.; Elarian, Y.; Awaideh, S.; Alvi, A. Utilizing Diacritic Marks for Arabic Text Steganography. *Kuwait J. Sci. Eng. KJSE* **2010**, *37*, 89–109.
47. Memon, M.S.; Shah, A. A Novel Text Steganography Technique to Arabic Language Using Reverse Fat5th5ta. *Pak. J. Eng. Technol. Sci.* **2015**, *1*, 106–113. [CrossRef]
48. Ahmadoh, E.M.; Gutub, A.A.-A. Utilization of Two Diacritics for Arabic Text Steganography to Enhance Performance. *LNIT* **2015**, *3*, 42–47. [CrossRef]
49. Daemen, J. AES Proposal: Rijndael. Available online: https://www.cs.miami.edu/home/burt/learning/Csc688.012/rijndael/rijndael_doc_V2.pdf (accessed on 4 June 2023).
50. Majeed, M.A.; Sulaiman, R.; Shukur, Z.; Hasan, M.K. A Review on Text Steganography Techniques. *Mathematics* **2021**, *9*, 2829. [CrossRef]
51. Li, B.; He, J.; Huang, J.; Shi, Y.Q. A Survey on Image Steganography and Steganalysis. *J. Inf. Hiding Multimed. Signal Process.* **2011**, *2*, 142–172.
52. Shih, F.Y. *Digital Watermarking and Steganography: Fundamentals and Techniques*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2017; ISBN 978-1-4987-3877-4.
53. Jero, S.E.; Ramu, P.; Swaminathan, R. Imperceptibility—Robustness Tradeoff Studies for ECG Steganography Using Continuous Ant Colony Optimization. *Expert Syst. Appl.* **2016**, *49*, 123–135. [CrossRef]
54. An Introduction to Arabic Calligraphy: 1 Hardcover—Illustrated, 28 Octobers 2016. Online at Desertcart KSA. Available online: https://www.desertcart.com.sa/products/49607260-an-introduction-to-arabic-calligraphy-1?gclid=CjwKCAjwyeujBhA5EiwA5WD7_Uo-HlQaKj1oXO5pI0L6hYXs478Yw2Pr6RSn90IPJVq9TC87Agmi6BoCkt8QAvD_BwE (accessed on 3 June 2023).
55. Shirali-Shahreza, M.H.; Shirali-Shahreza, M. Arabic/Persian Text Steganography Utilizing Similar Letters with Different Codes. *Arab. J. Sci. Eng.* **2010**, *35*, 213–222.
56. Bensaad, M.L.; Yagoubi, M.B. High Capacity Diacritics-Based Method for Information Hiding in Arabic Text. In Proceedings of the 2011 International Conference on Innovations in Information Technology, Abu Dhabi, United Arab Emirates, 25–27 April 2011; pp. 433–436.