Steganography Arabic Text Based on Natural Language Process Documents

*Assist.prof. Dr. Hanaa M. Ahmed
University of Technology

**PhD Students: Maisa’a A. A. Khohder
University of Technology

Abstract: Obscurity is a main reason whereas computers can not know natural language. It have made great transaction steps trend developing instrument to morphological and syntactic analyzers for Arabic. One of the manners used in security areas is steganography. The rapid development of steganography scripts, it is a large security and confidentiality problem, it becomes necessary to find appropriate protection because of the significance, accuracy and sensitivity of the data during transmitted.

In this research is offer in a new method and to use one level to hide, this level is hiding by embedding and addition. The one level is embed a secret message twice, one bit in the LSB in the FFT and the addition of one kashida and add Single-Double Quotation in the same secret message. Using Random Singular Value Decomposition (RSVD) is NRG to find positions that are hiding within the text.

Linguistic steganography is covering all the techniques that deal with using written natural language to hide secret message. in this research presents a linguistic steganography for scripts written in Arabic language, using kashida, Single-Double Quotation and Fast Fourier Transform on the bases of using new technique entitled Random Singular Value Decomposition (RSVD) as allocation to hide secret message. The proposed approach is an attempt to present a transform linguistic steganography using one level for hiding to improve implementation of kashida and Single-Double Quotation, and improve the security of the secret message by using Random Singular Value Decomposition (RSVD). Are testing this method in terms of security and capacity, transparency, and robustness and this is way better than previous methods. The proposed algorithm ideal steganography properties.

Keyword: Arabic text, Linguistic Steganography, random Singular Value Decomposition, Kashida, Single-Double Quotation, Transform Based
المختصر:

الغموض هو السبب الرئيسي في عدم التعرف إلى اللغات الطبيعية، وأنها جعلت أكبر العمليات كخطوة باتجاه التطور وتحليل قواعد علوم الصوف في اللغة العربية. واحدة من الطرق المستخدمة في المجالات إخفاء المعلومات الأمنة. إن التطور السريع لإخفاء معلومات النصوص، بامكانه السرية والأمنية تواجه مشكلة كبيرة، وصبح من الضروري إيجاد حماية مناسبة بسبب أهمية ودقة وحساسية البيانات أثناء إرسالها.

لإخفاء المعلومات، يمكن استخدام طريقة جديدة باستخدام مستوى واحد لإخفاء المعلومات، وهذا المستوى والكشف يتم عن طريق دمج وإضافة. تم تضمين مستوى واحد في رسالة سرية مرة، وت واحد في تحويل LSB وإضافة كاسحة واحدة وضيف اقتباس الفردية والزوجي لنفس رسالة سرية، باستخدام عشوائية FFT لايجاد المواقع التي يخفى داخل النص.

إخفاء المعلومات اللغوي يغطي جميع التقنيات التي تتعامل مع استخدام كتابة اللغة الطبيعية لإخفاء رسالة سرية. في هذا البحث يتم إخفاء المعلومات اللغوية للنصوص مكتوبة باللغة العربية، وذلك باستخدام كاسحة واقتباس الفردية والزوجي، وتحويل FFT على أسس باستخدام تقنية جديدة (RSVD). يتم إيجاد مواقع لإخفاء رسالة سرية، بطريقة المقترحة وهي محاولة إخفاء المعلومات اللغوي باستخدام مستوى واحد لإخفاء لتحسين تنفيذ كاسحة واقتباس الفردية والزوجية، وتحسين أمن الرسالة السرية باستخدام عشوائية تفكك القيم الفردية للصورة (RSVD). يتم اختيار هذه الطريقة من الناحية الأمنية والقدرات، والشفافية، ومتانة، وهذا هو وسيلة أفضل من الطرق السابقة. الخوارزمية المقتراحة لها خصائص في إخفاء المعلومات مثالية.
**Introduction:**

Linguistic steganography is focused on apply changes to a cover text so as to embed secret message, in a way that the changes do not caused any unnatural or ungrammatical text. The component of natural language steganalysis framework can be categorized into three groups [1], as depicted in Figure (1), [2]:

1- **Natural Language Processing Resources and Techniques**

Natural Language Processing (NLP) has accumulated a great deal of fundamental knowledge in steganalysis environment. Thus, there are four components in NLP that can be considered during the construction of natural language steganalysis framework which are identified as:[3]

- **Corpora Resources:** "Several numbers of electronic corpora are available on Internet that has been created for NLP research on such as WordNet and VerbNet. These corpora resource are considered because of their availability and accessibility".

- **Text Paraphrasing:** "One of the challenges in natural language steganalysis is to paraphrase a text in order to detect the hidden message".

- **Natural Language Parsing:** "This task can be described as reprocessing the text sentences and reproducing new structure for the sentences".

- **Linguistic Transformations:** "There are three types of lingulistic transformation in natural language environment which are synonym substitution, syntactic transformations, and semantic transformations" [3].
2- Principles of Steganography:

"Steganography involves hiding data in an overt message and doing it in such a way that it is difficult for an adversary to detect and difficult for an adversary to remove. Based on this goal, three core principles can be used to measure the effectiveness of a given steganography technique: amount of data, difficulty of detection, and difficulty of removal"[4].

- **Amount of data**: "suggests that the more data you can hide, the better the technique".

- **Difficulty of detection**: "relates to how easy it is for somebody to detect that a message has been hidden. There is usually a direct relationship between how much data can be hidden and how easy it is for someone to detect it. As you increase the amount of information that is hidden in a file, you increase the chance that someone will be able to detect that there is information hidden in the file".

- **Difficulty of removal**: "involves the principle that someone intercepting your file should not be able to remove the hidden data easily"[4].

I. **LITERATURE REVIEW AND FUNDAMENTAL USED IN PROPOSED SYSTEM**
Kashida is an Arabic redundant character which is used to justify the text, without affect the meaning of words. The Researchers suggested using one kashida as bit zero, and two kashida as bit one, or vice versa.

In 2007, A. Gutub, and M.Fattani, introduced a novel Arabic text steganography technique for Arabic script using letter points and kashida. The technique hides secret information as bits in Arabic letters (cover) by using kashida and points of letters. The technique consider un-point Arabic letters followed by a kashida if the secret bit is (0), and pint Arabic letters followed by kashida if secret bit is (1).

Their technique enhanced robustness and security but might have some limitation with capacity of the cover media if the number of secret bits of the secret information is large. This steganography technique is found to be suitable for other languages having similar scrip to Arabic for example Persian and Urdu [5].

In 2009, M. Hassan Shirali-Shahreza and Mohammad Shirali-Shahrzea, introduced a novel manner to conceal data in Persian (Farsi) and Arabic languages. In Unicode scale, there are to conceal two characters for 'Ya' (ی) and 'Kaf' (ک). The two characters of (ی) and (ک) has itself shape however various codes which they are utilized at the start or in the center of words. The major aim in this manner is perception translucence. It have a stellar perception translucence cause the stego-text who the employee sight is aright like for the main text [6].

In 2010, Adnan Abdul-Aziz Gutub, Wael Al-Alwani, and Abduleelah BinMahfoodh, introduced an improved Arabic text steganography technique for Arabic script using kashida. The approach hides secret information as bits within Arabic letters (cover) by using extension character (kashida). The technique considers one kashida if the secret bit is (0) and two kashida if secret bit is (1) after any letter can hold it. The finishing character is embedding just after the last bit of the secret information, then the kashida as is embed randomly to the rest script in
order to enhance the security of the technique. Also their technique enhanced security, capacity and robustness for Arabic scripts based secure communication [7]. A. Ali and F. Moayad, Introduced Arabic text steganography technique for Arabic script using kashida with Huffman code. The approach hides secret information as bits within Arabic letters (cover) by using extension character (kashida), and compressed the stego file using Huffman code. The technique considers absence of kashida if the secret bit is (0) and one kashida if secret bit is (1) after any connected letters. Also their technique applied to other than Arabic script based secure communication, with different document formats [8].

In 2013, Ammar Oden, Khaled Elleithy, Miad Faezipour, introduced an improved Arabic text steganography technique for Arabic script using variation kashida. The approach selected one of four scenarios randomly to hide secret information as bits within Arabic letters (cover) by using kashida. The technique considers un-point Arabic letters followed by a kashida if the secret bit is (0), and point Arabic letters followed by kashida if secret bit is (1) as first scenario, and vice versa as second senior. The third senior is adding kashida after Arabic letters if the secret bit is (1) and (0) otherwise, vice versa as forth senior. Also their technique enhanced security, complexity for Arabic script based secure communication [9].

A. Fast Fourier Transform and its Inverse

The mathematical formula to Fourier Transform of a time domain function \( f(x) \), for real numbers \( x \) and \( y \) is [12]:

\[
F(y) = \int_{-\infty}^{+\infty} f(x) \exp[-i2\pi xy] \, dx \quad \ldots \ldots (1)
\]

And the mathematical formula to its inverse is [10]:

\[
f(x) = \int_{-\infty}^{+\infty} F(y) \exp[i2\pi xy] \, dy \quad \ldots \ldots (2)
\]
Where:

\( f(x) \): Time domain function
\( F(y) \): Frequency domain function
\( x \): Argument with units of time
\( y \): Argument with units of frequency
\( e \): Base of natural logarithms
\( i \): Imaginary unit \((i^2 = -1)\).

**B. Singular Value Decomposition (SVD)**

Singular Value Decomposition technique splits given matrix into a product of orthonormal matrices and a diagonal matrix. The mathematical formula to Singular Value Decomposition is [11].

\[
A = USV^T
\]  \( .................................................. (3) \)

\[
A = \begin{bmatrix} u_1 & u_2 & \ldots & u_m \end{bmatrix} \begin{bmatrix} s_1 & 0 & \ldots & 0 \\ 0 & s_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & s_n \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{bmatrix}, \quad \ldots (4)
\]

Let \( A \) be an \( m \times n \) matrix. Performing SVD to \( A \) factorizes it into a product of orthogonal matrix, diagonal matrix and another orthogonal matrix as:

\[
A = USV^T \quad .................................................. (5)
\]

Where,

\( A \): original image matrix
\( U \): \( m \times m \) product of orthogonal matrix
\( S \): \( m \times n \) diagonal matrix
\( V \): \( n \times n \) orthogonal matrix
C. Random Singular Value Decomposition (RSVD)

Is a new technique to generate a set of random position \((x_i, y_j)\) to apply the embedding algorithm, from decomposing original image \((A)\) using SVD, the result is \(B\). Detect the non-zeros elements, and converges into nearest integer the results is RSVD. Figure (2) is example for the original image.

![Original image example](image)

Figure (2): Original image example.

The original image \(A\) is decomposed into three matrixes: \(U\) of size \(m_i \times m_j\) matrix, \(V\) of size \(m_i \times n_j\) matrix, and \(D\) of size \(n_i \times n_j\) matrix. The new array \(B = S*V*D\) as depicted in Figure (3), indicates random location from original image \(A\).
Figure (3): Random location generated using RSVD.

Algorithm RSVD:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input: original image A</td>
</tr>
<tr>
<td>2</td>
<td>Output: random number</td>
</tr>
<tr>
<td>3</td>
<td>Process:</td>
</tr>
<tr>
<td>3.1</td>
<td>Step1: input A</td>
</tr>
<tr>
<td>3.2</td>
<td>Step2: Apply SVD algorithm to A</td>
</tr>
<tr>
<td>3.3</td>
<td>Step1: B = U<em>S</em>V^T</td>
</tr>
<tr>
<td>3.4</td>
<td>Step2: For i= 1 to Length of B</td>
</tr>
<tr>
<td>3.5</td>
<td>Step3: No= B[i];</td>
</tr>
<tr>
<td>3.6</td>
<td>Step4: No= absolute (No);</td>
</tr>
<tr>
<td>3.7</td>
<td>Step5: While (integer (No)==0)</td>
</tr>
<tr>
<td>3.8</td>
<td>Step6: No =No*10;</td>
</tr>
<tr>
<td>3.9</td>
<td>Step7: RSVD[i]=(integer (No));</td>
</tr>
<tr>
<td>3.10</td>
<td>Step8: Next</td>
</tr>
<tr>
<td></td>
<td>End of algorithm</td>
</tr>
</tbody>
</table>

III. PROPOSED SYSTEM

IV. IDEA FOR PROPOSED SYSTEM

The proposed approach main idea as depicted in Figure (4) the embedding, and Figure (5) the extraction, is to use RSVD as generated random offset location, to added random kashida characters and single-double quotation to the rest Arabic word texts as a first layer, where the first
layer is inject the secret message bits in the inverse FFT (LSB of (real (FFT) of selected Arabic text word))), and then apply one kashida character. The first addition of kashida is for the hiding process of the secret information, while the second addition of the first layer is inject again the secret message bit using single-double quotation is for confusion purpose to insure security of the secret message.

Figure (4): The proposed hiding process.

Figure (5): The proposed extraction process.
Embedding process

- The flowchart of embedding process, as shown in Figure (6).

Figure (6): The flowchart of embedding process.

Embedding Algorithm:
Input: secret message, image A, set of Arabic texts.
Output: stego-cover.
Process:

Step1. Secret message binarization: The secret message is hidden in form of (0) s, and (1) s, which represent (64) bit Unicode of each character using the hexadecimal representation. \( N \), is the total number of secret message bits. Figure (8) presents the binarization process to secret
Generate Random positions: The process of generated random positions, using RVSD, starts by using applying SVD algorithm to the input image (A) to generate a sequence of random values C that represents offset of Arabic text words to start the embedding process. The total number of Generate Random positions is \( N \), where \( N \) is the total number of secret message bits.

Step2. Cover selection: select Arabic text (cover) that can hold input secret message bits.
Step3. Do while not end of Arabic text words
Step4. Embedding layer one: For each secret message bit and Generate Random positions do
Step5. Use c value as offset to next word to embed the secret message bit, into
   inverseFFT (LSB (real (FFT (select Arabic text word)))), then apply one
   kashida if the secret message bit is one or if the secret message bit is zero.
Step6. End of For.
Step7. Else
Step8. Embedding layer one: add again secret message bit single-double quotation randomly to the rest Arabic text words
Step9. End of Do.
Step10. End.

Figure (7): Secret message Binarization.
469

Figure (8): Secret message binarization Example.

a. Extraction Process

- The flowchart of extraction process, as shown in Figure (9).

```
Extraction Algorithm:

Input: secret message, image A, stego cover.
Output: secret message.
Process:
```

![Diagram](image-url)
Step 1. Generate Random positions: The process of generated random positions, using RSVD, start by using applying SVD algorithm to the input image (A) to generate a sequence of random values C that represents offset of Arabic text words to start the extraction process.

Step 2. Loading: Load stego-cover, and Generate Random positions.

Step 3. For each Generate Random Positions do

Step 4. Use C value as offset to next word to extract the secret message bit, from LSB of select Arabic text word (stego-cover).

Step 5. End of For.

Step 6. Converts each seven bits into one letter the result is the secret message.

End.

V. RESULTS AND DISCUSSION

In this section discuses to cases to ensure the proposed technique security:

Case one: Embedding Kashida

An example of result of applying the proposed technique using embedding layer

one, as depicted in Figure (10).
It can be concluded from case one that is visually easy to find the locations of secret message that is embed in stego-cover.

**Case two: Embedding Single-Double Quotation**

An example for applying the proposed technique using embedding layer one kashida and applying the proposed technique (embedding layer one put again single or double quotation ) as depict in Figure (11) using the same secret message is hide twice.
It can be conclude from case two that is visually not easy to find the locations of secret message that is embed in stego-cover.

**Case three: Robustness**

An example of result of applying the proposed technique using embedding layer one the steganography have no change, this state indicates to robustness in kashida or single double quotation. As depicted in figure (12).
Figure (12): Proposed technique example robustness in layer one.

Case four: Security

In this proposed technique, the secret message is hidden in FFT in LSB and the FFT is transformed to IFFT in layer one, the secret message is not known by the attacker. Thus where all kashidas and single double quotation in layer one are deleted, data can be retained in the hide of secret message in LSB. This technique gives high security. The Jaro-Winkler is applied method, as depicted in table (1), table (2), table (3), and table (4).

The Jaro distance is:  
\[ dj = \frac{1}{3} \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m - t}{m} \right) \]
If the word is ليلقيها without stego, \( dj = \frac{1}{3}(7/7+7/7+7-1/7) = 0.9523 \)

where \( t = 1 \)

If the word is ليلقيها stego in layer one, \( dj = \frac{1}{3}(8/8+8/8+8-1/8) = 0.9583 \)

else

If the word is ملامحة stego in layer one, \( dj = \frac{1}{3}(6/6+6/6+6-1/6) = 0.9444 \)

where \( t=2 \)

If the word is ملامحة' stego in layer one, \( dj = \frac{1}{3}(8/8+8/8+8-2/8) = 0.9166 \)

else

If the word is الشاعر stego in layer one, \( dj = \frac{1}{3}(6/6+6/6+6-1/6) = 0.9444 \)

where \( t=2 \)

If the word is الشاعر" stego in layer one, \( dj = \frac{1}{3}(8/8+8/8+8-2/8) = 0.9166 \)

\[ 
\begin{array}{c|cccc|c|c|c}
& ل & ي & ق & م & د & س & ل & ي & ق & م & د & س \\
\hline
\text{Layer one} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Layer one} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Layer one} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Layer one} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Layer one} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Layer one} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\text{Layer one} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\hline
\end{array} 
\]

Table (1): Similarity between cover and stego cover in layer one.
Table (2): Similarity between cover and stego cover in layer one.

<table>
<thead>
<tr>
<th>Layer one</th>
<th>Single quotation</th>
<th>cover without stego</th>
<th>Stego cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>‐</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>م</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ل</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ا</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>م</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ه</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table (3): Similarity between cover and stego cover in layer two.

<table>
<thead>
<tr>
<th>Layer one</th>
<th>Double quotation</th>
<th>cover without stego</th>
<th>Stego cover</th>
</tr>
</thead>
<tbody>
<tr>
<td>‐</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>م</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ل</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ا</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>ع</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ه</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table (4): Explaining hide capacity ratio in proposal algorithm.

<table>
<thead>
<tr>
<th>No of cover</th>
<th>Secret message size (Byte)</th>
<th>Secret message size (KB)</th>
<th>Carrier file size (Byte)</th>
<th>Carrier file size (KB)</th>
<th>Average of hide capacity ratio %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22528</td>
<td>22</td>
<td>54272</td>
<td>53</td>
<td>0.4150 B or KB</td>
</tr>
<tr>
<td>2</td>
<td>22528</td>
<td>22</td>
<td>81920</td>
<td>80</td>
<td>0.275 B or KB</td>
</tr>
</tbody>
</table>
Case five: Transparency

In this proposed technique is very high transparency, because not seen in human vision and not clear for attack. Special when the text without kashida and single-double quotation. As depicted in figure (13) and figure (14).

<table>
<thead>
<tr>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
</tr>
<tr>
<td>Case five: Transparency</td>
</tr>
</tbody>
</table>

**Figure (13): Proposed technique example Transparency in layer one.**

**Figure (14): Proposed technique example Transparency in layer one.**
Case six: Capacity

In this proposed technique the capacity is change during hiding a secret message, because in the first state is convert Arabic text to FFT and two state is addition the kashida in layer one and injection single Double quotation in layer two. The amount of hiding data is increase in cover, because addition and injection in file carrier imply relative increase in stego cover. The equation below is:

\[
\text{Hidden Ratio} = \frac{\text{amount of hidden data}}{\text{carrier file size}}
\]

For example:
- Hide ratio = 22KB/53 KB = 0.4150
- Hide ratio = 22 KB/ 80 KB = 0.257

VI. CONCLUSION:

In this paper a new layers Arabic language steganography is implemented using the FFT implementation and Kashida as an embed process, and RSVD as random location generator to embed the Arabic secret message in the Arabic script. We present some conclusions bellow:

1. Applying Steganography methods to document (text) files as a cover which is written by Arabic language is difficult, due to the visually sensitivity of Arabic letters to any miner change as in case one.
2. The RSVD is fast search algorithm, which is improved to use as means to allocate randomly positions in the cover media (Arabic scripts) to perform the embedding operation.
3. As embedding methods, usually frequency method is harder against attack than time domain method, so using FFT and Kashida as embedding method, which improve its security agents attack.
4. Algorithm robustness: The proposed algorithm prohibits any change to carrier (Arabic script) during the transmission process since the hidden secret message does not change the cover (Arabic script) file properties such as, file size, content, and format during the transmission.
5. Algorithm transparency: the proposed algorithm improves the transparency property by hiding secret message inside the Arabic script using FFT. In addition another layer of hiding is applied using Kashida.

6. Algorithm security: the proposed algorithm improves the security property by hiding secret message inside the Arabic script using FFT and apply kashida as first layer then apply single or double quotation as second layer to the rest Arabic script.

7. Algorithm Capacity: This algorithm is more capacity after hide a secret message in cover Arabic text as the equation is:

\[
\text{Hidden Ratio} = \frac{\text{amount of hidden data}}{\text{carrier file size}}
\]
References


[7] Adnan Abdul-Aziz Gutub, Wael Al-Alwani, and Abdulelah Bin Mahfoodh,


