American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
© Global Society of Scientific Research and Researchers
http://asrjetsjournal.org/

Information Security System Based on English and Myanmar Text Steganography

May Htet[a]*, Khin Myo Thant[b]

[a]Ph.D. Student, Department of Computer Engineering and Information Technology, Mandalay Technological University, Mandalay (100/107), Myanmar
[b]Lecturer, Department of Computer Engineering and Information Technology, Mandalay Technological University, Mandalay (100/107), Myanmar
[a]Email: mayhtet@iuj.ac.jp
[b]Email: khinmyothant97@gmail.com

Abstract

Due to the growth in frequent exchange of digital data over public channel, information security plays an important role in daily part of communication. Hence, various techniques like steganography are applied in information security area for more efficient information security system. This paper proposes two new text steganographic approaches using two different languages which are based on Unicode standard for secure data transfer over the public channel. The main aim is to overcome the limited embedding capacity, suspiciousness, and data damaging effect due to modification of existing steganographic approaches. The first approach conceals a message, without degrading the cover, by using four specific characters of words of the English cover text. The second approach performs message hiding by using the three specific groups of characters of combined words in Myanmar cover text while maintaining the content of the cover. The structure and operation of the proposed approaches based on the idea of existing text steganographic technique: Hiding Data in Paragraph (HDPara) algorithm. In this work, an empirical comparison of the proposed approaches with HDPara approach is presented. According to the comparison results, the proposed approaches outperform the existing HDPara approach in terms of embedding capacity.

Keywords: Embedding Capacity; Information Hiding; Steganography; Text Steganography; Myanmar Text Steganography.

----

* Corresponding author.

1. Introduction

In digital world, exchanging the multimedia information and private data over the common communication media such as Mobile or the Internet will draw the attention of the unwanted third parties. This may cause illegitimate attempts including snooping, data alteration, and stealing to crack and reveal the secret data. Thus, information security becomes awareness in today society and it demands for techniques that can ensure the security.

Steganography is one of the key techniques that can guarantee the security of secret information. It has been the more attractive alternative than others for information security as it can even protect the presence of secret information on communication channel by hiding it inside other multimedia objects [1]. Once someone recognizes the presence of data inside, the goal of steganography is defeated. In steganography, the secret data is the text that is secreted in the cover object, which can be an innocent carrier, for example, text, image, audio, and video files. After concealing the secret data inside the cover object, the stego object is generated. The stego key or position file is a file generated together with stego object after hiding process. The requirement of secret key in steganography process is optional.

For the proposed work, only text file is considered as the cover object. In this work, the proposed text steganographic techniques are used to hide the secret message into the innocuous cover text file.

The rest of the paper is organized as follows: Section 2 presents earlier works and recent researches on text steganography. Section 3 depicts the general model of proposed system. Section 4 describes the proposed approaches. Section 5 shows the performance analysis of the proposed methods. Section 6 draws the conclusion.

2. Related Works

Several researches have been made in the text steganography area using different languages for achieving a robust information security system. Some popular and recent researches based on text steganography are listed below.

In recent research [2], K.F. Rafat presented the Feature Coding Method. In this method, the characteristics of the text are modified to hide the secret message. For example, the length of characters such as h, b, d, are lengthened or shortened, letter points in characters such as i and j can be displaced etc. thereby concealing the information in the text. The major drawback of this method is that if a character recognition program is used or if retyping is done, it will automatically correct the text and the embedded content would get destroyed.

In word shifting method, the words are shifted horizontally or their distances are changed from left to right depending on the bit sequence thereby hiding the information in the text. The main weakness of this approach is if the distance is observed by someone, it is possible to obtain the hidden text from the stego text. In addition, the use of word processing program causes the harm to the hidden text [3].

Then, Shirali-Shahreza [4] presented a new text steganograpic method for hiding data in English text. This method makes use of different US and UK spellings of words in sentence for hiding the bit 0 or 1. Before data hiding, a list of words which have different UK and US spelling is prepared. Then, the embedding algorithm substitutes these words in the text depending on the secret bit. It has little capacity to hide data in the text.

In this paper [5], Megha Pathak suggested a new Steganographic scheme, which is useful for Hindi Language electronic writing. This method proposes a numerical code to Hindi letters which is built on the basis of 4-bit binary value. Later we replaced each 4-bit code with a different word which starts with the respective letters that are assigned in the scheme. This method requires a strong mental power and takes a lot of time. It also requires a special word and not all type of words can be used in this method.

Mujtaba S. Memon and Asadullah Shah suggested a new technique for information hiding in Arabic language. This approach has been developed by reversing the display manner of short vowel marks or Fatha in the Arabic language to hide secret messages within the text. The major weakness of this method is that the reverse Fatha used on the selected character gives the hints to detect the secret message [6].

In [7], Sravani Alameti and his fellows proposed a new approach for steganography in Telugu text. Secret message hiding in Telugu alphabets is performed by shifting inherent vowel signs either left or right. The main problem of this approach is that the stego text cannot withstand the modification causing the loss of hidden data.

Moreover, Monika Agarwal presents three novel approaches of text steganography and concludes that the proposed approaches achieve better embedding capacity than some other existing approaches [8].

Based on the idea from earlier researches, this paper presents two novel approaches of text steganography. The first approach uses the pre-defined English text and the second approach uses the pre-determined any meaningful piece of Myanmar text as cover file to hide the secret bits. Both proposed approaches aim to fulfill all the three main performance parameters of steganography: capacity, security, and robustness.

3. The Proposed System Model

Figure 1 shows the proposed system model. The model consists of two portions: sender side and receiver side.

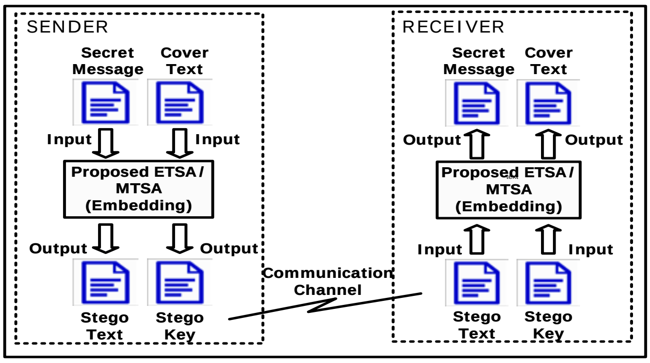

Figure 1: Proposed system model

At the sender side, the secret message is embedded in the cover text using the proposed embedding algorithm and generates the stego text and stego key. The stego text and stego key are sent to the receiver over the communication channel.

By the side of receiver, the receiver receives stego text together with the stego key from the sender. By using the proposed extracting algorithm, the receiver extracts the hidden secret message and original cover text from the stego text with the help of stego key.

4. The Proposed System Model

In this effort, the proposed approaches are used to hide both English and Myanmar messages in cover texts written in English and Myanmar languages. Each proposed approach has different embedding and extracting algorithms. The proposed approaches work on the proposed system model shown in Figure 1. Both approaches work on the binary value of embedded characters.

4.1. The Proposed English Text Steganographic Algorithm (ETSA)

The approach makes use of a pre-determined English cover text which can be taken from any source (e.g. a paragraph from a newspaper/magazine/book). It uses first (f), second (s), second last (s′), and last (l) letters of the words of a cover text for hiding the secret bits.

Firstly, the secret message must be converted into binary. Then each bit is read in order from the binary file. After that, a word is picked up sequentially from the cover text and it is written down in the stego text. Then each word will be checked for two times.

The first check is to make sure that the first and last letter of the word of the cover is different. The word with same first and last letter is not considered for hiding the secret bit.

If the first and last letter of the word is different, that word can be used for hiding message. The second check is to confirm that the second and second last letter of the word of the cover text is different.

Like the first check, a word having same second and the second last letter will be skipped for concealing the consecutive bit. The message hiding process is performed till the end of the binary file.

As there is no change or modification to the cover text in hiding process, the indistinguishable stego text in appearance as the cover text is produced at the end of the embedding process

In extracting process, the reverse procedure is performed by comparing the words in the stego text with the characters in the stego key. The extracting process is performed till the end of the stego key.

Algorithm for Message Embedding

1. Get a cover text.

2. Convert the letters in secret message file to its binary equivalent (bin).
3. Read a bit (x) from the bin.
4. Read a word from the cover text and write it in the stego text. (f = first letter, l = last letter, s = second letter, and s´= second last letter of the word.)
5. If first and last letter of the word is the same, then go to step 4.
6. If x = 0, write “f” in the stego key.
7. Else if x = 1, write “l” in the stego key.
8. If second and second last letter of the word is the same, then go to step 3.
9. Read a bit (x) from the bin.
10. If x = 0, write “s” in the stego key.
11. Else if x = 1, write “s´” in the stego key.
12. Repeat steps 3 to 11 till the end of the bin file.
13. Send the stego text and the stego key to the receiver.

Algorithm for Message Extraction

1. Get the stego text and stego key.
2. Read a character (c) from the stego key.
3. Read a word from the stego text and write it in the cover text. (f = first letter, l = last letter, s = second letter, and s´= second last letter of the word.)
4. If first and last letter of the word is the same, then skip that word and go to step 3.
5. If c = f, then bit b = 0.
6. Else if c = l, then bit b = 1.
7. Write b in a file.
8. If second and second last letter of the word is the same, then go to step 2.
9. Read a character (c) from the stego key.
10. If c = s, then bit b = 0.
11. Else if c = s´, then bit b = 1.
12. Write b in a file.
13. Repeat steps 2 to 12 till the end of the stego key.
14. Convert the file into its character equivalent.
15. Obtain the original message.

Figure 2 and Figure 3 show the examples of data embedding and extracting processes of the proposed ETSA approach.

Figure 4 shows the selected cover text in which the secret binary file is to be hidden. Figure 5 shows the generated stego text after hiding the binary file in cover text. It can be seen that the appearances of the two text files are indistinguishable.

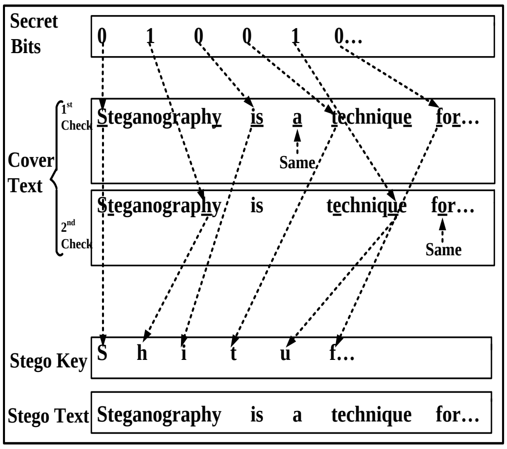
Secret Bits
0 1 0 0 1 0…
1st Check
Cover Text
Steganography is a technique for…
Same
2nd Check
Steganography is technique for…
Same
Stego Key
S h i t u f…
Stego Text
Steganography is a technique for…

Figure 2: Example of message embedding process

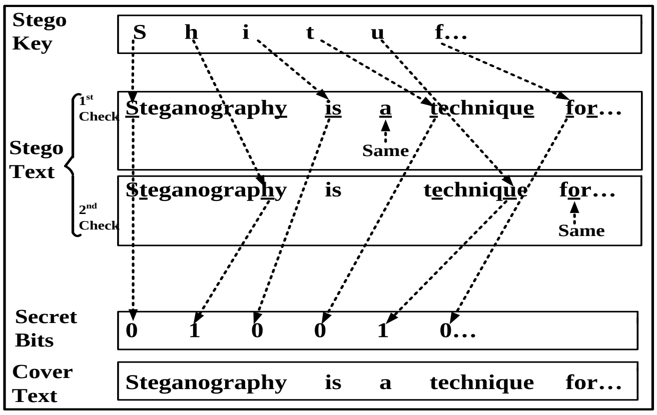
Stego Key
S h i t u f…
1st Check
Stego Text
Steganography is a technique for…
Same
2nd Check
Steganography is technique for…
Same
Secret Bits
0 1 0 0 1 0…
Cover Text
Steganography is a technique for…

Figure 3: Example of message embedding process

Steganography is a technique for hiding data such as messages into another form of data such as images files. Basically, one communicating party can use steganography to conceal secret message called covered data into an image file called carrier file. The carrier file is then sent to the other communicating party who will decipher it and the secret data hidden inside the different cover file called carrier.

Figure 4: Cover text

Steganography is a technique for hiding data such as messages into another form of data such as images files. Basically, one communicating party can use steganography to conceal secret message called covered data into an image file called carrier file. The carrier file is then sent to the other communicating party who will decipher it and the secret data hidden inside the different cover file called carrier.

Figure 5: Stego text

4.2. The Proposed Myanmar Text Steganographic Algorithm (MTSA)

This approach uses the meaningful Myanmar cover text file from sources (e.g. Myanmar magazine, text book, newspaper).

Myanmar language has a total of 75 characters, which are rounded in shape as shown in Table 1 and Myanmar script is written from left to right [8].

As Myanmar characters are connected together (where in English all words consist of separated letters), in order to perform word-based information hiding procedure using Myanmar text, Myanmar words are needed to be separated out before embedding process.

Therefore, a word itself may be a single word (standard-alone word) or combined word which can be any combination of these characters.

A combined word can consist of one consonant, zero or more medial, vowel, and dependent various sign. The order of alphabets in combined word is sorted by the "Thinbongyi" approach which is the current national standard [9]. Figure 6 shows the example of Myanmar word segmentation.

Table 1: An example of a table

Category Name	Name	Glyph	Unicode Code Point
C	Consonants	က ခ ဂ ဃ င စ ဆ ဇ ဈ ဉ ည ဋ ဌ ဍ ဎ ဏ တ ထ ဒ ဓ န ပ ဖ ဗ ဘ မ ယ ရ လ ဝ သ ဟ ဠ အ	U+1000…U+1021
M	Medials	◌ျ ◌ြ ◌ွ ◌ှ	U+103B…U+103E
D	Dependent Vowels	◌ါ ◌ာ ◌ိ ◌ီ ◌ု ◌ူ ေ◌ ◌ဲ	U+102B…U+1032
V	Myanmar Sign Virama	◌္	U+1039
A	Myanmar Sign Asat	◌်	U+103A
F	Dependent Various Signs	◌ံ ◌့ ◌း	U+1036…U+1038

I	Independent Vowels, Independent Various Signs	ဤ ဧ ဪ
		၌ ၍ ၏
			U+1024; U+1027;
			U+102A; U+104C;
			U+104D; U+104F;

E	Independent Vowels, Independent Various Sign Aforementioned	ဣ ဥ ဦ ဩ
		၎
			U+1023; U+1025;
			U+1026; U+1029;
			U+104E;

G	Myanmar Letter Great Sa	ဿ	U+103F
S	Myanmar Digits	၀ ၁ ၂ ၃ ၄ ၅ ၆ ၇ ၈ ၉	U+1040…U+1049
P	Punctuations	၊ ။	U+104A…U+104B

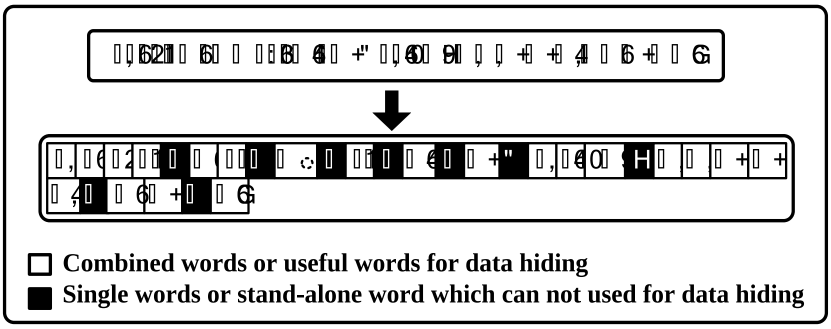
□ Combined words or useful words for data hiding
■ Single words or stand-alone word which can not used for data hiding

Figure 6: Example of word segmentation

This approach works on the binary value of the characters of secret message. It performs message hiding using consonant (C), medial (M), or dependent vowels (D) of the combined words of the Myanmar cover text.

After converting the secret message into binary form, each secret bit is hidden by reading a word in sequence from the Myanmar cover text and writing it in the stego text. If the word is a single word or acts as stand-alone words, skip it and read another word from the cover text. Then, depending on the bit to be secreted, write the corresponding characters either consonant, medial, or dependent vowel of the word in stego key. The data hiding process is repeated till the end of the binary file.

Similarly, this approach also produces exactly the same stego text in appearance as the cover text. It shows that the cover text has not been modified due to embedding.

Like the ETSA approach, in extracting process, the original message is extracted from the stego text by comparing the characters in stego key with the words in stego text. The extracting process is repeated until all the embedded characters are extracted from the stego text with the help of stego key.

Algorithm for Message Embedding:

1. Get the Myanmar cover text.
2. Convert the letters in input secret message to its binary equivalent (bin) using Unicode.
3. Read a bit (x) from the bin.
4. Read a word from the cover text and write it in the stego text. (C = consonant, M = medial, and D = dependent vowel of the combined word)

Assume that one word is completed if the following patterns are found:

i. Single words or stand-alone words which can be either consonants (C) (e.g. က, ခ), independent vowels and Independent various signs in group I (e.g. ဣ, ဧ, ၌ ၍) and group E (e.g. ဥ, ဩ, ၎), Myanmar letter great sa (G), Myanmar digits or punctuations which are neither followed by medials (M), dependent vowels (D), dependent various signs (F), Myanmar sign asat (A) or invisible Myanmar sign virama (V) but after which next consonants, independent vowels and Independent various signs in group I and group E, Myanmar letter great sa, Myanmar digits or punctuations are found.

ii. Combined words which are started with either consonants, independent vowel in group E or Myanmar letter great sa and ended by either medials, dependent vowels, dependent various signs, Myanmar sign asat or invisible Myanmar sign virama after which next consonants, independent vowels and Independent various signs in group I and group E, Myanmar letter great sa, Myanmar digits or punctuations are found.

5. If the word is a single word or acts as stand-alone word, skip this word and go to step 4.
6. Check the character (b) in the combined word.
7. If x = 0, and b = M or D, write “b” in the stego key.
8. Else if x = 1, and b = C, write “b” in the stego key.
9. Else skip this character (b) and go to step 6.
10. Repeat steps 3 to 9 till the end of the bin file.
11. Send the stego text and the stego key to the receiver.

Algorithm for Message Extraction

1. Get the stego text and stego key.
2. Read a character (c) from the stego key.
3. Read a word from the stego text and write it in the cover text. (C = consonant, M = medial, and D = dependent vowel of the combined word.)
4. Assume that one word is completed if the same patterns are found as like in embedding process.
5. If the word is a single word or acts as stand-alone word, skip this word and go to step 3.
6. Check the character (b) in the combined word.
7. If c = b, then

   If b = M or D, then write 0 in the file.

   Else write 1 in the file.

   End if

8. Else skip this character (b) and go to step 5.
9. End if.
10. Repeat steps 2 to 8 till the end of the stego key.
11. Convert the file into its character equivalent using Unicode.
12. Obtain the original message.

Figure 7 and Figure 8 show the examples of data embedding and extracting processes of the proposed MTSA approach. The pre-selected cover text in which the secret binary file is to be concealed is shown in Figure 9. The stego text generated after hiding the binary file in cover text is depicted in Figure 10. The two files are exactly the same.

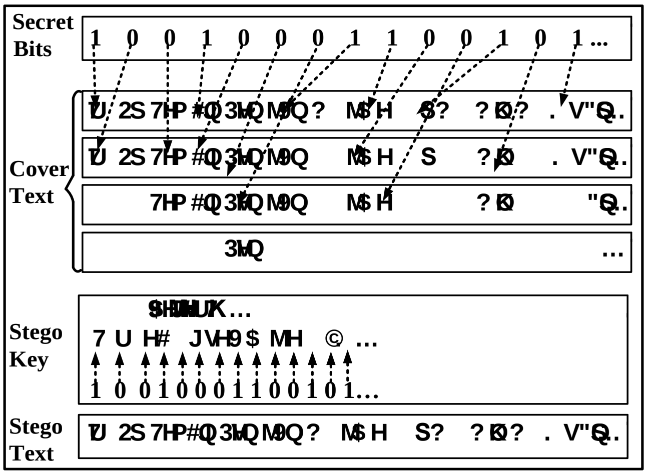

Figure 7: Example of message embedding process

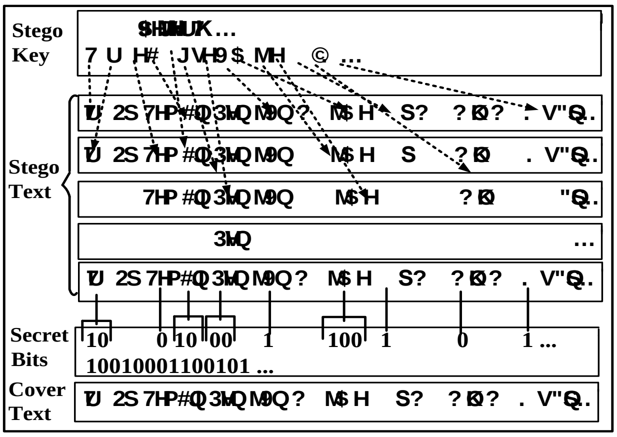

Figure 8: Example of message embedding process

Figure 9: Cover text

Figure 10: Stego text

5. Performance Analysis

To measure the effectiveness of the proposed embedding algorithms, it is considered with four aspects: capacity consideration, similarity measure of cover text and stego text, security consideration of the proposed approaches.

5.1. Capacity Consideration

Capacity is the amount of information that is to be hidden in a cover carrier. It is usually represented in percentage form [10]. Capacity ratio is calculated as:

Capacity ratio = (amount of hidden bytes)/ (size of the cover text in bytes) *(1)*

*Percentage Capacity = Capacity ratio *100* *(2)*

As characters are encoded as Unicode characters, each character occupies 2 bytes in memory.

Table 2: Example secret messages

Secret Message	1	2	3	4	5	6	7	8	9	10
Size in Bytes	6	12	24	64	146	278	502	942	1560	1840

Figure 11 compares the percentage capacity of the proposed approaches with that of existing text steganographic approach, HDPara. The proposed approaches provide better embedding capacity than HDPara algorithm. Figure 12 shows the average percentage capacity of the approaches.

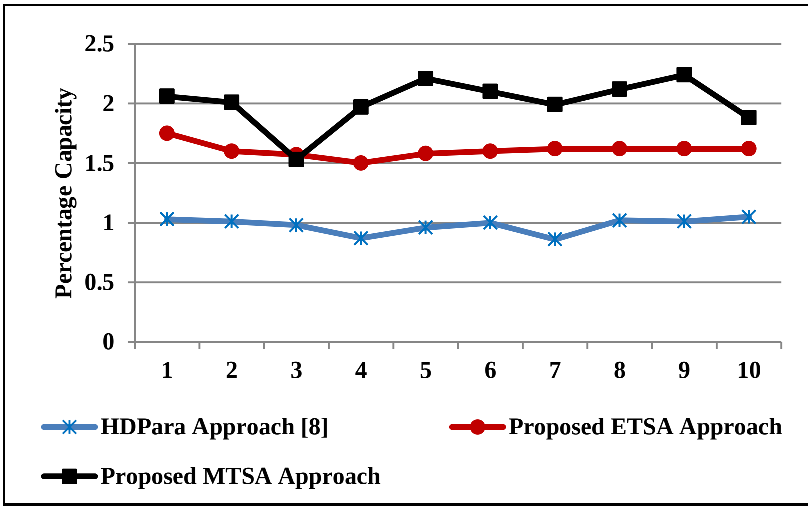

Figure 11: Percentage capacity of the proposed approaches over ten experimental sample data

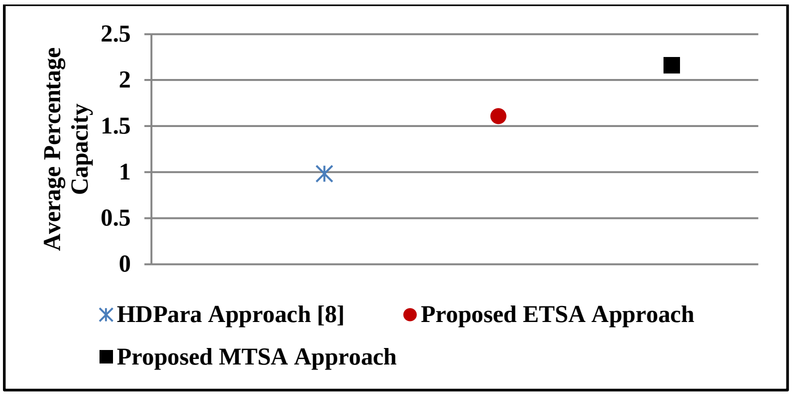

Figure 12: Average percentage capacity of the approaches

From the cover size point of view, due to the different structures and operations of embedding approaches, each approach requires different sizes of cover to hide the same 200 bytes of secret message as illustrated in figure 13. In this figure also, the proposed approaches are more efficient as they require smaller cover size than the HDPara does.

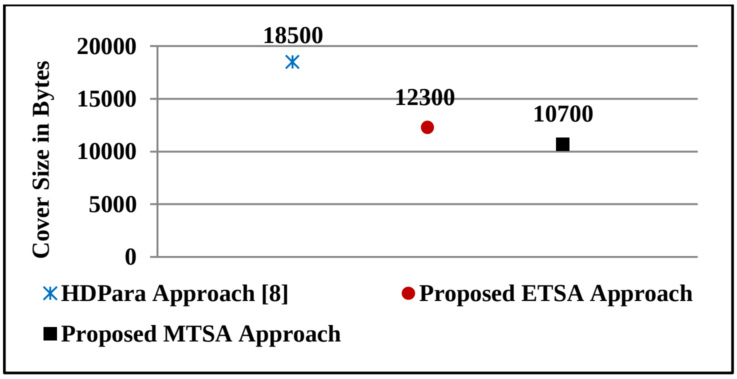

Figure 13: Maximum Cover Size (Bytes) Required Hiding 200 Bytes of Secret Message

In figure14, the same cover size is used for each approach to hide message size of 400 bytes. It shows that the amount of secret bytes hidden by each approach is different and the proposed approaches can hide more amounts of secret message even though all approaches use the same cover size.

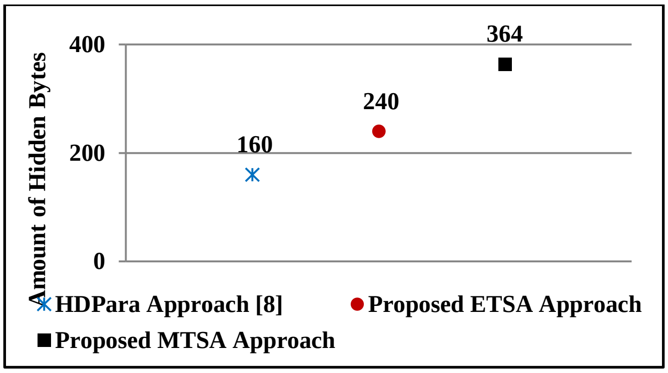

Figure 14: Number of Bytes Hidden Using Same Size of Covers when Message Size is 400 Bytes

As it can be seen in figure 15, the percentages capacities of the approaches are different for each case, where different sizes of covers with different content of useful characters are used for hiding the same 500 bytes of message. In figure 16 also, the percentages capacities of the approaches are different for each case, where different sizes of covers with same content of useful characters are used for hiding the same 500 bytes of message. According to the results from figure 15 and figure 16, it can be said that the percentage capacities of the approaches depend on the number of useful characters, the sequence of the useful characters in the cover text and the binary sequence of the secret message.

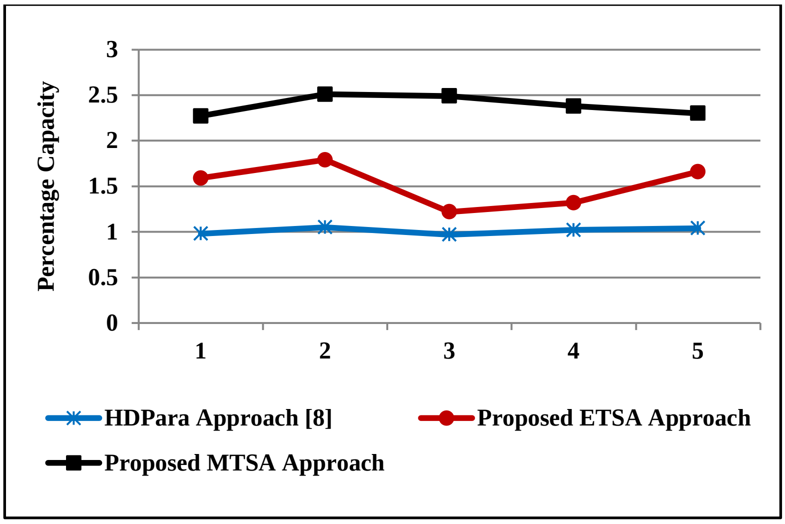

Figure 15: Percentage Capacity of the Approaches Using Different Types of Covers with Different Number of Useful Characters for Hiding the Same 500 Bytes of Message

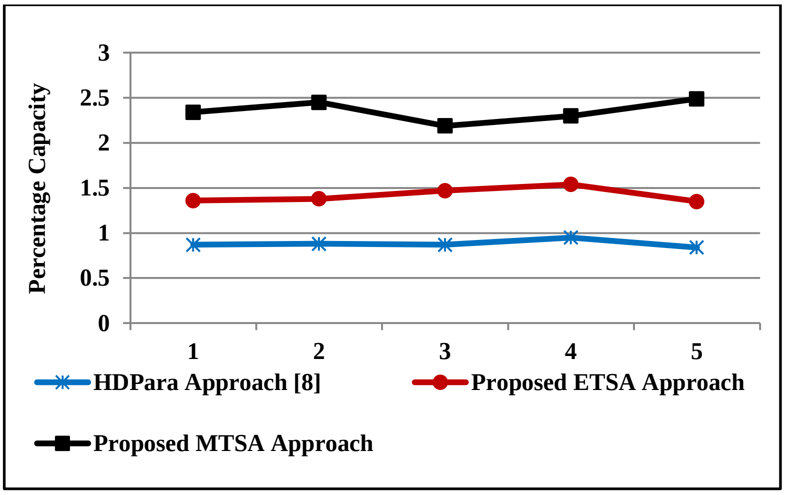

Figure 16: Percentage Capacity of the Approaches Using Different Types of Covers with Same Number of Useful Characters for Hiding the Same 500 Bytes of Message

5.2. Similarity Measure of Cover Text and Stego Text

Similarity metrics are used to measure distance between two values or sequence. As the aim of steganography is to hide the existence of secret data, similarity between cover text and stego text is important to show that the cover text has not been modified due to embedding. In this paper, Jaro-Winkler similarity metric is applied for comparing the similarity between cover text and the stego text.

Jaro-Winkler Similarity Metric

The Jaro-Winkler metric (Jaro score), which is a variant of the Jaro distance, is a measure of similarity between two strings. The Jaro–Winkler distance metric is designed and best suited for short strings such as person names. It is mainly used in the area of record linkage (duplicate detection). The higher the Jaro score for two strings is, the more similar the strings are. The score is normalized such that 0 equates to no similarity and 1 is an exact match [11]. The Jaro-Winkler metric is calculated as:

$$\text{Jaro-Winkler}\,(s1, s2) = Jaro(s1, s2) + (L*p\,(1-Jaro(s1, s2))) \qquad (3)$$

$$Jaro\,(s1,\ s2) = \frac{1}{3} * \left(\left[\frac{m}{s1} \right] + \left[\frac{m}{s2} \right] + \frac{m-t}{m} \right) \qquad (4)$$

, where s1 and s2 are the two strings whose similarity is to be measured, L is the length of common prefix, p is the scaling factor whose standard value is 0.1, m is the number of matching characters and t is the number of transpositions. Two characters from s1 and s2 respectively are considered matching only if they are not farther than

$$\left[\frac{max|s1|,|s2|}{2} \right] - 1 \qquad (5)$$

Each character of s1 is compared with all its matching characters in s2. The number of matching (but different sequence order) characters divided by two defines the number of transpositions.

In this paper, in both proposed embedding approaches, since data is hidden without altering the cover text, the cover and its corresponding stego files are exactly the same getting the Jaro score of “1”.

5.3. Security Consideration

The proposed methods produce the same stego text in appearance as the cover text. Thus, it will not reveal to public about the existence of any hidden data. Even if a snooper obtains both cover and stego files, any difference between the two files will not be figured out. In addition, in the proposed approaches, more than two specific letters of words are used for hiding the secret bit. Therefore, at the security point of view, it is impossible for anyone to extract the hidden data from the stego media without knowing the stego key. Moreover, the steganalysis in Myanmar text steganography is more complex for intruders than that in English text steganography as foreign person cannot comprehend Myanmar language. Therefore, the proposed approaches can provide good confidentiality requirement for secret data.

6. Conclusion

For more robust information security system, two new embedding algorithms for information hiding in text are presented in this paper. The proposed algorithms perform secret data embedding by using specific letters of a word of any natural looking meaningful piece of English and Myanmar text. Depending on the bit sequence to be secreted, the corresponding characters are directly written in the stego key. According to the Jaro Winkler similarity measure, the average Jaro score of proposed approaches are exactly one as they do not modify the content of the cover in embedding operation. Hence, the proposed approaches satisfy the security requirement of steganographic system. Besides, the approaches can withstand the modifications of stego text with various changes such as retyping or opening the stego text with word processing program and protect the correct format of hidden data.

Moreover, in the proposed algorithms, even if larger secret message is hidden, there is no need to consider the larger cover size. Therefore, it can be assumed that the larger the size of secret message, the greater the percentage capacity of the proposed approaches. Thus, the data hiding capacity of proposed approaches is possible to be higher than most of the existing approaches which work on the binary value of embedded characters. In this work, the observed percentage capacity of proposed approaches is better than the existing HDPara approach even in the case of without looping the cover text.

As mentioned above, the proposed approaches fulfill all three needed aspects of steganography. Therefore, they can be effectively applied in many security concern applications such as protecting the private data or password of a banking system or data storage center. As a limitation, the proposed approaches allow message hiding only in cover text which uses the font type based on Unicode standard and they can access only text file format. As further extension, the concepts of the proposed approaches can be modified and extended to be suit in other languages which have similar font structure and they should be developed to access other file format.

References

[1] Adnan Abdul-Aziz Gutub, and Manal Mohammad Fattani. “A novel Arabic text steganography method using letter points and extensions”. World Academy of Science, Engineering and Technology, International Journal of Computer, Information, Systems and Control Engineering, vol. 1, pp.483-486, 2007.

[2] K.F. Rafat. “Enhanced text steganography in SMS,” in Proc. of the 2nd Int. Conf. Computer, Control and Communication, 2009, pp.1-6.

[3] K. Alla and R.S.R. Prasad. “An evolution of Hindi text steganography,” in Proceeding of the 6th International Conference on Information Technology: New Generations (ICIT09), 2009, pp.1577-1578.

[4] Shirali-Shahreza. “Text steganography by changing words spelling,” presented at the 10th International Conference Advanced Communication Technology, 2008 (ICACT 2008), 17-20 Feb. 2008.

[5] P. Megha. “A new approach for text steganography using Hindi numerical code”, International Journal

of Computer Applications, vol. 1, 2010, pp. 56-59.

[6] Mujtaba S. Memon and S. Asadullah. “A novel text steganography technique to Arabic language using reverse Fatha”, Pak. j. eng. technol. sci. vol. 1, 2011, pp.106-113.

[7] S. Alameti et.al. “A new approach for steganography in Telugu texts by shifting inherent vowel signs”, International Journal of Engineering Science and Technology, vol. 2, 2010, pp. 7203-7214.

[8] A. Monika. “Text steganographic approaches: a comparison”, International journal of network security & its applications (IJNSA), vol.5, no.1, pp. 91-106, January 2013.

[9] M. Zin Maung and M. Yoshiki,, “A rule-based syllable segmentation of Myanmar text”, in Proc. IJCNLP-08 Workshop on NLP for Less Privileged Languages, January 2008, pp.51-58.

[10] H. Martin. Representing Myanmar in Unicode: details and examples version 3, SIL International and Payap University Linguistics Institute, Chiang Mai, Thailand, 2007, pp. 01-45.

[11] F. A. Haidari, A. Gutub, K. A. Kahsah, and J. Hamodi. “Improving security and capacity for Arabic text steganography using “kashida” extensions,” IEEE/ACS Int. Conf. on Computer Systems and Applications, 2009.

[12] S. D. Pandya, P. V. Virparia. “Testing various similarity metrics and their permutations with clustering approach in context free data cleaning”. Int. journal of computer science and security, vol.3, pp. 344-350 2009.