2015

Urdu text steganography: Utilizing isolated letters

Aliya Tabassum Abbasi
Iqra University

Nuzhat Naqvi
Iqra University

Aihab Khan
Iqra University

Basheer Ahmad
Iqra University,

DOI: 10.4225/75/57b699bed938a
13th Australian Information Security Management Conference, held from the 30 November – 2 December, 2015 (pp. 37-46), Edith Cowan University Joondalup Campus, Perth, Western Australia.

URDU TEXT STEGANOGRAPHY: UTILIZING ISOLATED LETTERS

Aliya Tabassum Abbasi, Syeda Nuzhat Subi Naqvi, Aihab khan, Basheer Ahmad
Iqra University Islamabad, Pakistan
{aliyatabassumabbasi, nuzhat.shah}@gmail.com,
{aihab, drbasheer}@iqraisb.edu.pk

Abstract

This paper presents an imperceptible and high capacity feature based approach which hides a secret message into Urdu text cover media by utilising all isolated letters. Existing techniques are less imperceptible and also not robust against steganalysis attacks and some of these schemes are failed to provide the better capacity rates. Previous lexical based and syntax based schemes are ineffective to provide the better capacity rate and image based approaches are not robust against format attacks. Moreover, Feature based approaches are more perceptible and thus, cannot resist against visual attacks. This paper proposes an improved algorithm that encompasses all isolated letters of Urdu text for hiding data to provide better capacity rates. Furthermore, this technique is more secured by using strong public key encryption algorithm. In addition, scheme is also imperceptible, since it does not affect the external appearance of the text. Implementation shows that the proposed text steganography technique provides high concealing capacity.

Keywords
Urdu text steganography, Public key Encryption, Isolated letters, Capacity and Imperceptibility.

INTRODUCTION

Steganography amongst the major discipline of information hiding has been getting more and more importance now a day. It can be defined as "To hide the secret information in such an intelligent way that nobody figures it out". However, another discipline such as cryptography changes the contents of information which draws the attention of the intruders and hence faces various malicious attacks. The word steganography is made up of two Greek words 'steganos' means "concealed or protected" and 'graphein' means "writing" [1]. The main purpose of steganography is to protect information in a way that intruders cannot determine and catch it. A message is a hidden information in the form of plain text, cipher text, images or anything that can be encoded into a bit stream. This message is embedded in a cover-carrier to create a stego-carrier. Stego medium consisted of cover medium, embedded medium and stego key. After data embedding, the text containing secret data, which is called stego-text, is sent from sender side to receiver side over the Internet. The security concept is central around the idea that no one can easily discover the secrets embedded into the stego-text by using statistical computation or other methods of detection [3]. Text steganography is divided into following categories as shown in Table1.

Table 1: Classification of steganography

TypesofText Steganography	**Techniques**
Technical (TS)	Word shifting, Line shifting, Feature coding, HTML tag, Image, Audio, Video and Network Steganography
Linguistic (TS)	Syntactic, Semantic, Abbreviations and Change of spelling
Random & Statistical	Probabilistic context-free grammar, Character Sequence, Words Sequence and Sequences-text mimicking
Miscellaneous Techniques	Typographical errors and Transliteration

There are three important parameters (criteria) in designing steganography systems such as: perceptual transparency, robustness, hiding capacity [2]. The security or perceptual transparency refers to the ability of an eavesdropper to figure, or suspect the hidden information easily. We can achieve high security by minimising the embedding Impact (distortion). Intuitively, we can try to achieve this by minimising the distortion between the cover text and stego text and by restricting the distortions to the portions of the cover text that are difficult to model. The robustness refers to the ability to protect the unseen data from corruption, especially when transmitted through the internet [3]. The capacity or the embedded rate refers to the ability of a cover media to store secret data, which can be measured by the amount of secret data (bits) that can be hidden in a kilo byte of a

cover media. "The ratio of the size of a secret message to the size of carrier text is used to measure the embedding capacity rate of the algorithm".

There are different types of cover medium used in steganography i.e. text, images, audio, video etc. as in [4-6]. Choosing carrier file is very sensitive as it plays a key role to protect the embedded message. Using text is preferred over other media because the texts occupy lesser space and provide more information efficiently. Text steganography is difficult as text contains little redundancy compared to other media and one other reason is that humans are curious to the text looking unusual. Text steganography schemes must be specifically designed to exploit the specific characteristics of the target language because the grammatical and orthographic characteristics of every language are different [7]. Most of the text steganography methods are applied to English texts however; there are a few text steganography methods applied to other languages as work presented in [7-11]. A few works have been done on hiding information in Arabic texts [12-16] which can be implemented on Urdu text to some extent, but prone to some drawbacks in terms of steganography parameters such as imperceptibility, robustness, and capacity.

Urdu text steganography has not been proposed so far. To the best of our knowledge, this work makes an initial contribution that utilises Urdu text for steganography purpose. This research aims to provide such a framework that overcomes the limitation of previous linguistic approaches toward Arabic steganography and also well suited for other similar scripted languages. The presented approach exploits the characteristic of Urdu text with improved Unicode based approach to achieve high capacity and imperceptibility.

The rest of the paper is organised as follows: the related work is underlined in Section 2, the proposed method and framework is detailed in Section 3. In Section 4, proposed Algorithms have been elaborated. Experimental results are shown in Section 5. At Section 6, we arrive at the conclusion.

Related Work

The following is a list of different methods carried out for Arabic text as reported in [17], and thus can be applied on Urdu text to some extent.

In Urdu and similar subscripted languages, the dot is very important and 14 of 28 Arabic letters have one or more dots. In format based method as explained in [7-8, 12], data are hidden by using these letters. Using the approach of the vertical displacement of the points, we can hide information in the texts. However, we need to use a special font created mainly for this purpose [3] and hence, these schemes are more perceptible and also, not robust against text formatting attacks.

Feature-based approach includes diacritic [13-14], kashida [15-16] and Unicode [1, 18-19] based methods. The Arabic language uses different marks or diacritics which can be applied to other similar subscripted languages such that Urdu and Persian, extension character, (Harakat) which are optional to use. The main reason to use these symbols is to distinguish between words that have same letters (i.e., so that every word is pronounced differently). The diacritic, such as ‘‘Fatha", is used to hide bit ‘1'. However, the rest of the diacritics are used to hide a bit ‘0', because it was found out that ‘‘Fatha" represents almost half of the diacritics in any Arabic text as explained in [13-14]. The main disadvantage of this method is that it attracts the attention of the reader and also diacritics are among the fewer practices to be utilised for Urdu text.

Kashida based method add an extension (Kashida in Arabic) to a word to hold secret bit ‘one' and leave the word without extension (kashida) to hold secret bit ‘zero' as explained in [15-16]. Note that letter extension does not have any effect on the writing content. The main disadvantage of this method is that it attracts the attention of the reader, increases the file size and changes the appearance of the text. And also kashida character (elongated character) is rarely used in Urdu text and if used for steganography purposes, immediately draws the attention of the reader.

In Urdu and similar scripted languages, two values of code for the same letters are being used, one for the representation and other for its possible shapes. Benefiting from this unique characteristic of these languages, secret message bits are hidden using Unicode values of the same letter as approached in [1, 18-19]. This method is suitable for e-steganography. However, it is weak against steganalysis attacks. Some techniques of Unicode based steganography provide high capacity, but are not robust and secure against attacks [17] and vice versa. An approach in this paper changes a letter Unicode value to its own other Unicode form such as isolated to general form and thus, protecting the cover message from formatting attacks.

In lexical based steganography, the secret message is hidden to natural language text by using synonym [21]. In synonym-based approach; the cover text may look legitimate from a linguistics point of view given the adequate accuracy of the chosen synonyms. By reusing the same piece of text to hide a message can raise the feeling of

suspicion and the embedding capacity of information is low [20, 21]. In translation based scheme; the message is hidden in the errors (noise) which are naturally encountered in a machine translation (MT). The secret message is embedded by performing a substitution procedure on the translated text using translation variations of multiple MT systems [22]. Typos and ungrammatical abbreviations in a text, e.g., emails, blogs, forums, etc., are employed for hiding data and is called noise based approach. However, this approach is sensitive to the amount of noise (errors) that occurs in a human writing.

Benefiting from unique characteristics of Urdu language, this paper presents the high capacity and more imperceptible feature based approach which hides the secret message by changing the letters Unicode from their isolated to representative form.

PROPOSED METHOD

Before explaining the proposed methodology, this section firstly makes a short glance on the main characteristics of the Urdu language. Secondly, explains the Urdu Unicode standard. Finally, presents the proposed methodology.

Urdu is an Indo-Aryan language. It is the national language of Pakistan and is one of the twenty-three official languages of India. The Urdu Alphabet consists of 37 letters and has many characteristics [23]. Urdu script is a cursive text even when printed and words are formed by connecting the letters. Unlike English, it is written from right to left and also an Urdu letter has one to four shapes called ligature. The shapes of the letters get changed depending upon its position in the word. The four various positions of an Urdu letter are starting (letter beginning the word), end; (letter ending the word), medial; (the letter connecting both the preceding and following characters) and independent or isolated; the character is not connected to both the preceding and the following characters [23].

An Urdu character contains one to three points placed above or below the character that differentiates the characters of similar shape. In Urdu, 18 out of 37 characters have dots. An Urdu word is either fully connected or single. For instance; **محمد،فلسطین** or contains one or more sub-words, these sub-words contain either one or more isolated characters or further connected word [24]. For instance; **پڑھانا,سکول** . There are certain alphabets in Urdu that cannot be connected to the left; these letters break the flow of the pen within a word. These letters are

Table2: Isolated Letters in Urdu Text

Urdu Sentence	"حکومت پاکستان نے سیاحت کے فروغ کے لیے بہت سے اقدامات کئے۔ان اقدام سے سیاحت کے شعبے کو فروغ ملے گا۔														
Separated Words	فروغ		شعبے	سیاحت	ان		اقدامات			بہت	فروغ		سیاحت	پاکستان	حکومت
Isolated Letters	غ	و	Non	Non	ن	ا	ت	ا	ا	Non	غ	و	non	ن	Non

called "Breakers". Since, they cannot be connected from the left, and thus has two forms: final and isolated. When every letter in the word contains breakers, then each letter appears in its isolated form as explained in Table2. In Urdu Unicode standard, each letter contains four different Unicode which helps the shape determination routine to determine the letter glyph. These five Unicode forms of a letter are: isolated, representative, medial, initial and final. For example; the Unicode of the letter "**ب**" is 0628 and the codes FE91, FE92, FE90, and FE8F are being used to represent different forms as beginning **بـ**, medial **ـبـ**, end **ـب** and isolated form **ب** respectively. Proposed method in this paper utilises representative and isolated Unicode of Urdu letter to a secret message. Since, both these forms of Unicode represent the same shape of a letter which does not alter the cover text as discussed in Table3.

Table3: General and Isolated Unicode representation of the Urdu letters

letters	ب	ت	ث	ج	ح	خ	د	ذ	ر	ز	س	ش	ص	ط	ظ	ع	غ	ف	ق	ك	ل	م	ن	ه	و	ی
General Unicode	628	062A	062B	062C	062D	062E	062F	630	631	632	633	634	635	637	638	639	063A	641	642	643	644	645	646	647	648	649
Isolated Unicode	FE8F	FE95	FE99	062C	FEA1	FEA5	FEA9	FEAB	FEAD	FEAF	FEB1	FEB5	FEB9	FEC1	FEC5	FEC9	FECD	FED1	FED5	FED9	FEDD	FEE1	FEE5	FEE9	FEED	FEEF

Figure1 shows the proposed model of Urdu text steganography. The model consists of five building blocks: Encryption process, which enciphers a secret message using a strong mathematical public key encryption algorithm such as RSA based; Bit Evaluation process, which converts the enciphered message into binary bits; Randomisation and Swapping process, which divides encrypted secret message string randomly into even blocks

and performs swapped function on each block; Isolated letters Extraction process, which extracts all isolated letters from the cover message; Embedding process, based on the interchange of Unicode from isolated to general form.

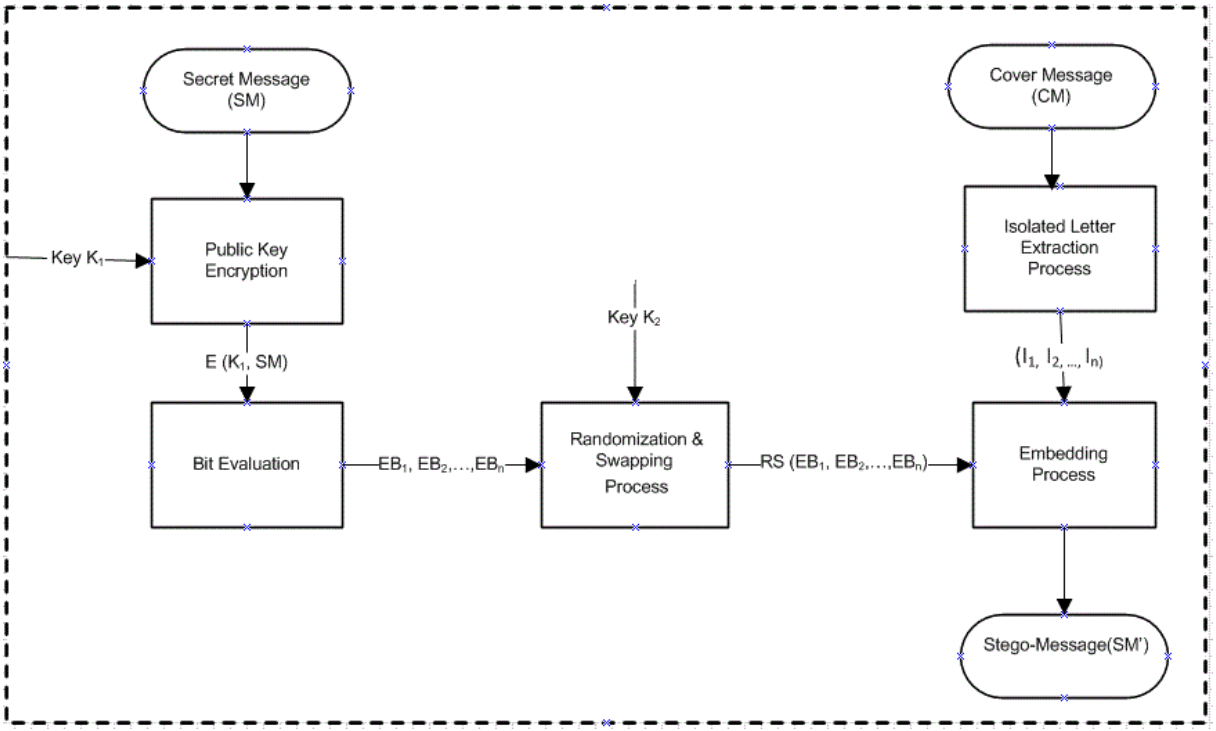

Figure1: Steganography Encryption & Embedding Framework.

Figure2 shows the extraction and decryption process which contains four blocks: Isolated Letter Extraction process, which extracts isolated letters from the stego text; Bit Extraction process, extracts all the bits as per embedding rules; Swapping Process, performs inverse swapping on extracted bits; Decryption process, decrypts the swapped bits by using a private secret key to get the desired secret message.

An example below illustrates the proposed methodology in detail. Suppose the cover message is a following piece of information taken from newspaper agency and presented method need to hide the secret binary bits in the following Urdu text. Suppose that we are going to embed the following encrypted bits (110010100111) into below cover text:

"یہ ہمارے لیے بدقسمتی کی بات ہو گی کہ اگر ہم سول اداروں کو تباہ ہونے سے نہ روک سکیں"

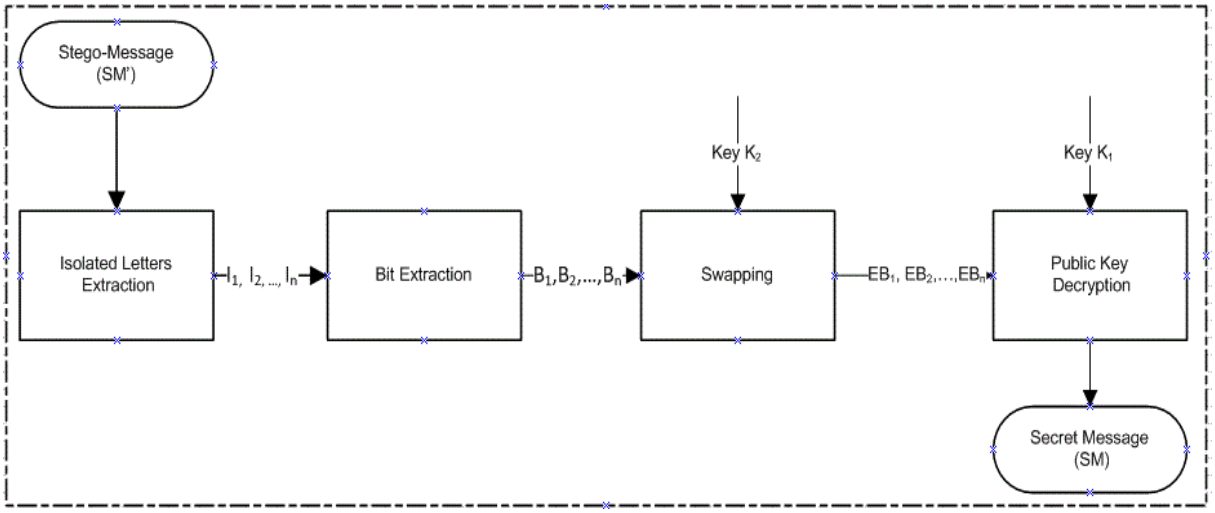

Figure 2: Steganography Extraction & Decryption Framework

Embedding of Secret message

According to the suggested algorithm, we execute the following steps to perform the embedding of secret message:

Step I: Read all isolated letters from the cover message.

Step II: Divide secret message bits (110010100111) randomly into even blocks.

Step III: Perform swap function on each block as per following procedure:

Read the first block "100111"

Swap the 1st bit with nth bit and 2nd with n-1th bit till the whole block bits are being swapped, resultant swapped bits of the first group are "111001".

Step IV: Read first isolated letter word 'ہمارے'which contains two isolated letters "ر" and "ے". Get the first secret message bit in a manner Urdu text has been read (i.e. right to left) from string "111001" which is "1". Change the Unicode of the letter "ر", from isolated to representative form as an indication of hiding "1", and the letter remain unchanged if it would be "0". Now we have hidden the first "1" in this text. Read a 2nd bit of the group which is "0", as the bit is different from the previously embedded bit, so change the Unicode of the letter "ے". Read the next word 'بات'which contains one isolated letter "ت" to embed the next bit "0" of secret string which is same as previously embedded bit, so Unicode of "ت" remain unchanged. We get another word 'اگر'

Table 4: How to hide '111000110010' in Urdu Sentence where ● means that we changed the Unicode of that letter, ◐ means we do not change the code of this letter

Urdu text	" حکومت پاکستان نےسیاحت کےفروغ کےلیےبہت سےاقدامات کیے،ان اقدام سےسیاحت کےشعبےکوفروغ ملے گا۔"													
Isolated words	فروغ			اقدام		ان		اقدامات				فروغ		پاکستان
Isolated letters in words	غ	و	ر	م	ا	ا	ن	ا	ت	ا	ا	غ	و	ن
Secret Message Bits =111000110010														
SWAP(G1)	111000 010011													
Embedding	غ	و	ر	م	ا	ا	ن	ا	ت	ا	ا	غ	و	●
SWAP(G1)	111000 010011													
Embedding	غ	و	ر	م	ا	ا	ن	ا	ت	ا	ا	غ	◐	●
SWAP(G1)	111000 010011													
Embedding	غ	و	ر	م	ا	ا	ن	ا	ت	ا	ا	●	◐	●
SWAPPED	111000 010011													
Embedding	غ	و	ر	م	ا	ا	ن	ا	ت	ا	◐	●	◐	●
SWAPPED	111000 010011													
Embedding	غ	و	ر	م	ا	ا	ن	ا	ت	●	◐	●	◐	●
SWAPPED	111000 010011													
Embedding	غ	و	ر	م	ا	ا	ن	ا	●	●	◐	●	◐	●
SWAPPED	111000 010011													
Embedding	غ	و	ر	م	ا	ا	ن	●	●	●	◐	●	◐	●
SWAP(G2)	011001 010011													
Embedding	غ	و	ر	م	ا	ا	ن	●	●	●	◐	●	◐	●
SWAP(G2)	001011 010011													
Embedding	غ	و	ر	م	ا	ا	◐	●	●	●	◐	●	◐	●
SWAP(G2)	000111 010011													
Embedding	غ	و	ر	م	ا	◐	◐	●	●	●	◐	●	◐	●
SWAPPED	000111 010011													
Embedding	غ	و	ر	م	●	◐	◐	●	●	●	◐	●	◐	●
SWAPPED	00011 010011													
Embedding	غ	و	ر	◐	●	◐	◐	●	●	●	◐	●	◐	●
SWAPPED	000111 010011													
Embedding	غ	و	◐	◐	●	◐	◐	●	●	●	◐	●	◐	●

Table 5: How to extract a secret message from stego text, where means that this letter is changed, means this letter is not changed

Stego Text	" حکومت پاکستان نےسیاحت کےفروغ کےلیےبہت سےاقدامات کیے،ان اقدام سےسیاحت کےشعبےکوفروغ ملے گا۔"													
Isolated Words	فروغ◐			اقدام			ان●		اقدامات			فروغ		پاکستان
Isolated Letters in Words	غ	و	ر	م	ا	ا	ن	ا	ت	ا	ا	غ	و	ن
Unicode Values			FEAD	FEE1	0623	FE83	FEE5	0623	062A	0623	FE83	063A	FEED	0646
Embedded locations	غ	و	◐	◐	●	◐	◐	●	●	●	◐	●	◐	●
Extracted Bits	غ	و	0	0	0	1	1	1	0	1	0	0	1	1
Randomized Grouping & Swapping	000111010011, SWAP G1(010011)=110010, SWAP G2 (000111)=111000													
Secret Message Bits	111000110010													

which has one isolated letter "ا" and change the letter's Unicode as an indication of hiding "1". Read the next embedding bit (i.e. 1) and according to our algorithm, we get a word 'سول' with isolated word "ل" from the cover text and we don't the letter as to hide "1". Leave the first isolated letter "ا" from the word 'اداروں'unchanged as an indication of hiding "1".

Step V: Repeat *Step III-IV* for embedding of 2nd group bits.

Extraction and Decryption Process

The extraction process of the above example runs under the following steps:
Step I: Read the isolated letters word from the stego text. In our example, we get the word 'ہمارے'with two changed isolated letters 'ر'and 'ے' , so the extracted string is "01".We get the next word 'بات'which has one unchanged isolated letter "ت" and we get "0" and hence the extracting string becomes "001".

Step II: Take the next word 'اگر' with changed isolated letter "ا" which is an indication of hidden "1", now extracting string becomes "1001". The next extracted word 'سول'contains one unchanged isolated letter "ل"which represent the same hidden bit as the previous one. Now the extracting yields to "11001".

Step III: We get another succeeding word 'اداروں'which contains first three unchanged isolated letters 'ادا' as an indication of "111" binary string followed by changed isolated letter 'ر' with succeeding unchanged and changed isolated letter 'و' and 'ں' respectively, so we add "100111" to the extracting string to become "10011111001". Extract the next isolated letter word 'تباہ'with one changed isolated letter "ہ" as an indication of hidden bit "0". Now ultimately extracted string becomes "010011111001".

Step IV: Divide the extracted string into groups using K2.

Step V: Perform the inverse swap operation on each group as indicated in Step III of the embedding process. The two groups after swapped operation are "100111" and "110010". Combine the swapped bits to get the encrypted binary string.

Step VI: Decrypt resultant values by using the private secret K_1 to get the desired secret message.
Table 4 elaborates the embedding procedure in a more precise way and shows that the suggested scheme has high payload due to utilisation of all isolated letters and more imperceptible since, it does not affect the external appearance of the text. To summarise the extraction procedure, the technique is completely opposite to that of embedding process as shown in Table5.

From the above example, we notice that the proposed algorithm encodes the hidden secret and it may change the code of an isolated letter to hide '1' in some place and it may leave the same isolated letter to hide '1' in another place which made the proposed algorithm hard to break statistically.

PROPOSED ALGORITHM

Our proposed method consists of three main algorithms:

- Encryption algorithm
- Swapping & Embedding algorithm
- Extraction & Decryption algorithm

Encryption algorithm

To add the additional layer of security; the proposed method first encrypt the secret message using a strong public key encryption algorithm such as RSA.

Step I: Convert the SM (secret message) to decimal blocks.

Step II: Apply the public key encryption algorithm RSA on each decimal block.

Step III: Return Encrypted decimal value.

Swapping & embedding algorithm

In this subsection, pseudo code of the encoding algorithm is presented.

It contains two sub algorithms; one for randomisation and swapping of secret message bits and other for the encoding these bits.

Input: CM (Cover message), SM (Secret bit stream).

Output: SM' (Stego message)

Step I: Divide SM into even binary blocks using pseudo-random seed function.

Step II: Apply the swap function on each binary block.
// Swap the 1st bit with nth bit and 2nd with n-1th bit till the whole block bits are being swapped.

Step III: Initiate the pointer of the reading from the CM to the first word in the text.

Step IV: While the SM bits do not get to the end do the following.

Step V: Read the starting bits of SM and embed as per following rule:
Extract the first isolated letter of CM.
While the SM bits are "0", isolated letter/s remains unchanged. Else isolated letter get changed.

Step VI: Read the remaining SM bits and embed as per following rules:
If the ith bit of the SM is same as previous ith+1 bit, then don't change the isolated letter. Else change the isolated letter.

Step VII: Repeat Step VI till the embedding of all bits.

Extraction & decryption algorithm

Extraction algorithm further contains three main sub algorithms; the first for extracting isolated letter bits, second for swapping the extracted bits and third for decryption of swapped bits to get the desired message.

Input: SM' (Stego Message), Stego- key K_1, Private key K_2
Output: SM (Secret Message)

Step I: Initiate the pointer to the reading of the first word of SM'.

Step II: Read the isolated letters and extract bit in the following manner

- While isolated letters are unchanged, extracted bit is/are considered to be "0"
- Else extracted bit is "1"

Step III: If next isolated letter is unchanged, extract same previous bit for the next extracting bit, else extracting bit is different from the previous one.

Step IV: Repeat *Step III* till the extraction of all SM bits.

Step V: Using the stego key K_2, divide the extracting string into groups and perform inverse swap operation on each group.

Step VI: Decrypt the swapped string using private key K_1 to get the desired SM (secret message).

EXPERIMENTAL RESULT

In this section, we compare the hidden capacity and its ratio of the proposed and existing technique [1]. We perform our experiment on ten different data sets of Urdu and Arabic text which are retrieved from [25]. Datasets include variety of news like sports, politics, and technology. The files resources are selected for computing the payload and payload ratio of proposed method and compare it with existing technique. The capacity ratio is calculated by using the formula:

Capacity Ratio = $\frac{size\ of\ hidden\ seceret\ (x)}{size\ of\ cover\ message\ (n)}$

Table 5: The Experimental Results of the Proposed & Existing Method

The Experimental Results of the Proposed & Existing algorithm.								
Text Size in kilobyte	Urdu Dataset				Arabic Dataset			
	PMHC (Bit)	PMHCR (b/kb)	EMHC (Bit)[1]	EMHCR (b/kb)[1]	PMHC (Bit)	PMHCR (b/kb)	EMHC (Bit)[1]	EMHCR (b/kb)[1]
16.3	918	56.3	595	36.5	1880	115.3	1500	92
12.4	827	66.7	518	41.8	1043	84.1	824	66.5
11.9	723	60.8	508	42.7	955	80.3	706	59.3
10.2	621	60.9	425	41.7	842	82.5	625	61.3
8.62	581	67.4	371	43	796	92.3	603	70
7.59	413	54.4	284	37.4	588	77.5	421	55.5
6.91	586	84.8	416	60.2	571	82.6	419	60.6
6.26	402	64.2	287	45.8	609	97.3	439	70.1
3.1	203	65.5	141	45.5	255	82.3	199	64.2
3.2	247	77.2	173	54.1	217	67.8	163	50.9
Avg Hidden Capacity Ratio (%)		65.8		44.9		86.2		65.04

Table5 shows the results of our proposed and existing scheme [1]. Proposed and existing approach, when experimented on Urdu text data has average hidden capacity ratio of 65.8 bits/kb and 44.9 bits/kb respectively. Our suggested approach when tested on Arabic text shows the concealing capacity ratio of approximately 86.2 and 65.04 bits/kb for proposed and existing approach respectively. Figure3 and Figure4 shows the comparison between PMHC (Proposed Method Hidden Capacity) of adopted technique with EMHC (Existing Method Hidden Capacity) and PMHCR (Proposed Method Hidden Capacity Ratio) and EMHCR (Existing Method Hidden Capacity Ratio) respectively of each method against ten different Urdu and Arabic cover messages. These figures clearly depict that our suggested approach has high peak percent intervals against all data sets both in terms of capacity and its ratio.

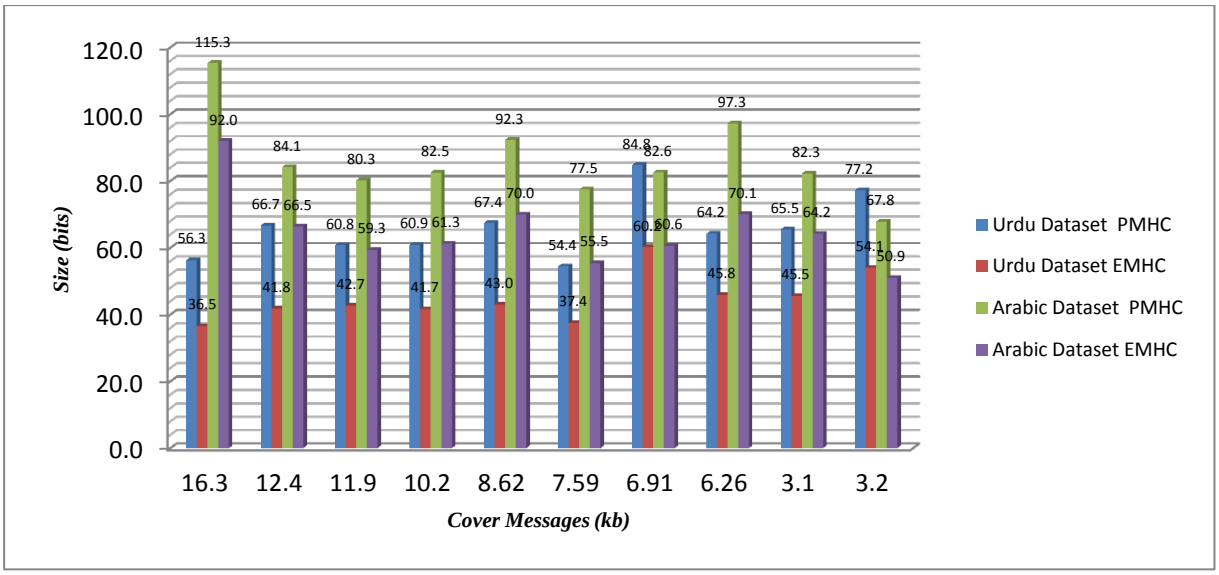

Figure 3: Comparison of hidden capacity in (bits) between Proposed & Existing Method

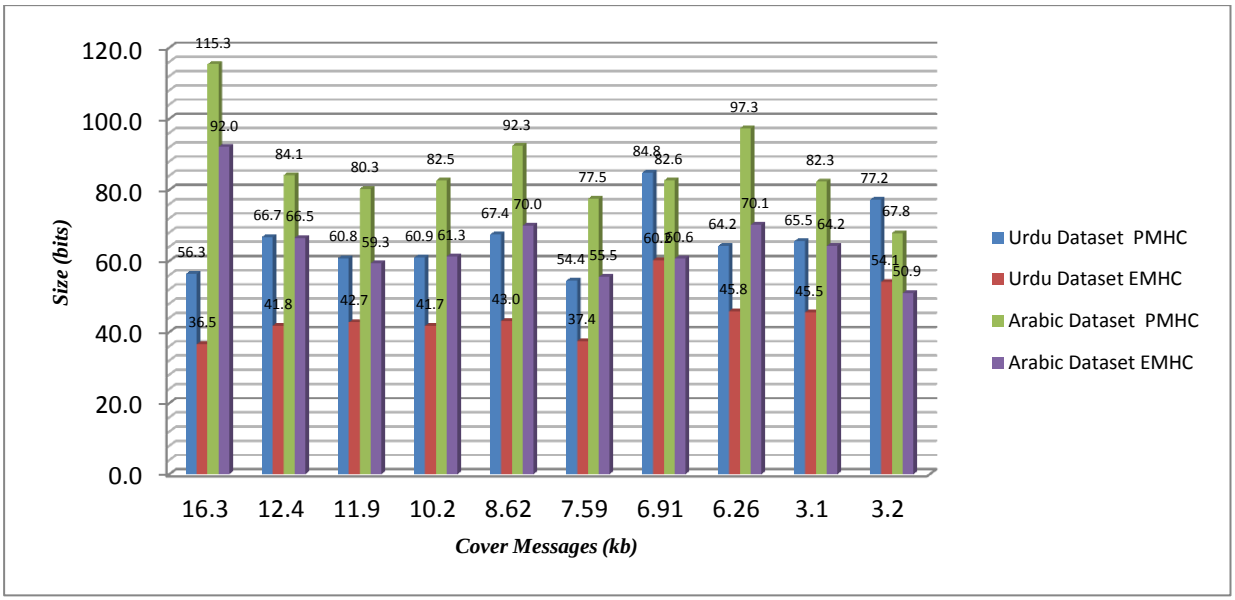

Figure 4: Comparison of hidden capacity ratio in (bits/kb) between Proposed & Existing Method

CONCLUSION

In this paper, an Urdu text steganography scheme is presented to enhance the hidden capacity by utilising all isolated letters of cover text. The proposed method is based on zero text steganography approach that does not alter the external appearance of the text, and thus enhances imperceptibility of stego-message. To protect from malicious attacks, we applied strong mathematical public key encryption algorithm along with randomisation and swapping function on secret message. Experimental results compare the suggested strategy payload and payload ratio with existing approach and concluded that our method has high hidden capacity on average. Besides Urdu text, this method can also be applied to other similar scripted languages like Arabic, Persian and Pashto etc. In the future, we will analyse robustness of proposed technique using probabilistic framework. A further improvement is expected to make it compatible for the printing documents as well.

REFERENCES

[1] A.A. Mohamed, "An improved algorithm for information hiding based on features of Arabic text: A Unicode approach," *Egyptian Informatics Journal* (2014) 15, 79–87.

[2] Shirali-Shahreza, M., & Shirali-Shahreza, M. H. (2008, August). An Improved Version of Persian/Arabic Text Steganography Using" La" Word. In*Telecommunication Technologies 2008 and 2008 2nd Malaysia Conference on Photonics. NCTT-MCP 2008. 6th National Conference on* (pp. 372-376). IEEE.

[3] Li, L., Huang, L., Zhao, X., Yang, W., & Chen, Z. (2008, August). A statistical attack on a kind of word-shift text-steganography. In *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on* (pp. 1503-1507). IEEE.

[4] Mohan, M., & Anurenjan, P. R. (2011, September). A new algorithm for data hiding in images using contourlet transform. In *Recent Advances in Intelligent Computational Systems (RAICS), 2011 IEEE* (pp. 411-415). IEEE.

[5] Gopalan K, "Audio steganography using bit modification," in*Proceedings of the IEEE international conference on acoustics, speech, and signal processing, (ICASSP '03),* vol. 2; 2003. p. 421–

[6] Doe¨rr G, Dugelay JL,"Security pitfalls of frame by-frame approaches to video watermarking,"*IEEE Trans Signal Process, Supply Secure Media* 2004;52(10):2955–64.

[7] Shahreza, M. S. (2006). A new method for steganography in HTML files. In*Advances in Computer, Information, and Systems Sciences, and Engineering*(pp. 247-252). Springer Netherlands.

[8] Khairullah, M. (2009, December). A novel text steganography system using font color of the invisible characters in Microsoft Word documents. In *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on*(Vol. 1, pp. 482-484). IEEE.

[9] Talip, M., Jamal, A., & Wen-Qiang, G. (2012, October). A proposed steganography method to Uyghur script. In *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on* (pp. 125-128). IEEE.

[10] Yuling, L., Xingming, S., Can, G., & Hong, W. (2007, July). An efficient linguistic steganography for Chinese text. In *Multimedia and Expo, 2007 IEEE International Conference on* (pp. 2094-2097). IEEE.

[11] Changder, S., Ghosh, D., & Debnath, N. C. (2010, November). Linguistic approach for text steganography through Indian text. In *Computer Technology and Development (ICCTD), 2010 2nd International Conference on* (pp. 318-322). IEEE,

[12] Odeh, A., Alzubi, A., Hani, Q. B., & Elleithy, K. (2012, May). Steganography by multipoint Arabic letters. In *Systems, Applications and Technology Conference (LISAT), 2012 IEEE Long Island* (pp. 1-7). IEEE.

[13] Aabed, M., Awaideh, S. M., Elshafei, A. R. M., & Gutub, A. (2007, November). Arabic diacritics based steganography. In *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on* (pp. 756-759). IEEE.

[14] Bensaad, M. L., & Yagoubi, M. B. (2011, April). High capacity diacritics-based method for information hiding in Arabic text. In *Innovations in Information Technology (IIT), 2011 International Conference on* (pp. 433-436). IEEE.

[15] Al-Azawi AF & Fadidhil MA. (2010). Arabic text steganography using Kashida extensions with Huffman code. Journal Application Science; 436–9.

[16] Al-Haidari, F., Gutub, A., Al-Kahsah, K., & Hamodi, J. (2009, May). Improving security and capacity for arabic text steganography using 'Kashida'extensions. In *Computer Systems and Applications, 2009. AICCSA 2009. IEEE/ACS International Conference on* (pp. 396-399). IEEE.

[17] Memon, J. A., Khowaja, K., & Kazi, H. (2008). Evaluation of steganography for Urdu/Arabic text. *Journal of Theoretical and Applied Information Technology*,*4*(3), 232-237.

[18] Por, L. Y., Wong, K., & Chee, K. O. (2012). UniSpaCh: A text-based data hiding method using Unicode space characters. *Journal of Systems and Software*,*85*(5), 1075-1082.

[19] Shirali-Shahreza, M., & Shirali-Shahreza, S. (2008, September). Persian/arabic unicode text steganography. In *Information Assurance and Security, 2008. ISIAS'08. Fourth International Conference on* (pp. 62-66). IEEE.

[20] Kessler, G. C., & Hosmer, C. (2011). An overview of steganography. *Advances in Computers*, *83*(1), 51-107.

[21] Alabish, A., Goweder, A., & Enakoa, A. (2013). A Universal Lexical Steganography Technique. *International Journal of Computer and Communication Engineering*, *2*(2), 153-157.

[22] Grothoff, C., Grothoff, K., Alkhutova, L., Stutsman, R., & Atallah, M. (2005, January). Translation-based steganography. In *Information Hiding* (pp. 219-233). Springer Berlin Heidelberg.

[23] Bhurgri, A. M. (2006). Enabling Pakistani Languages through Unicode.*Microsoft Corporation white paper at http://download. microsoft. com/download/1/4/2/142aef9f-1a74-4a24-b1f4-782d48d41a6d/PakLang. pdf*..

[24] Durrani, N., Sajjad, H., Fraser, A., & Schmid, H. (2010, July). Hindi-to-Urdu machine translation through transliteration. In *Proceedings of the 48th Annual meeting of the Association for Computational Linguistics* (pp. 465-474). Association for Computational Linguistics.

[25] www.bbcurdu/bbcarabic.com