A novel approach to secret data concealment with high cover text capacity and security

Muhammad Azeem*

College of Computer Science and Technology,
Faculty of Information Science and Technology,
3+Beijing University of Technology,
Beijing, 100124, China
Email: mazeem.qau@hotmail.com
*Corresponding author

Jingsha He

Faculty of Information Science and Technology,
College of Software Engineering,
Beijing University of Technology,
Beijing, 100124, China
Email: jhe@bjut.edu.cn

Allah Ditta

Division of Science and Technology,
University of Education,
College Road, Township, Lahore, Pakistan
Email: allahditta@ue.edu.pk

Faheem Akhtar

College of Software Engineering,
Faculty of Information Science and Technology,
Beijing University of Technology,
Beijing, 100124, China
and
Department of Computer Science,
Sukkur IBA University,
Sukkur, 65200, Pakistan
Email: fahim.akhtar@iba-suk.edu.pk

Sher Muhammad Daudpota

Department of Computer Science,
Sukkur IBA University,
Sukkur, 65200, Pakistan
Email: sher@iba-suk.edu.pk

Abstract: Massive technological advancement and rapid internet growth has made the security of secret data more challenging for security researchers. Though, steganography techniques were also evolved but still require large cover text to conceal large message. Therefore, to deal capacity issue, present research articulates a novel approach to achieve high data concealment capacity of cover text along with security by using combination of steganography and cryptography. Current algorithm applies four-layer encryption on secret message and seven Unicode characters such as invisible plus (IP), invisible separator (IS), inhabit symmetric swapping (ISS), left to right override (LRO), (ZWNJ), zero width joiner (ZWJ), zero width non-joiner and zero width character (ZWC), are employed to embed secret information into English carrier text to generate stego text object. The result reflects tremendous increase of 300% in carrier text capacity and encryption significantly enhances the security of secret information. Moreover, the identical carrier and stego text reduces hacker's attention.

Keywords: Unicode; bit rotation; carrier media; cryptography; text steganography.

Reference to this paper should be made as follows: Azeem, M., He, J., Ditta, A., Akhtar, F. and Daudpota, S.M. (2020) 'A novel approach to secret data concealment with high cover text capacity and security', *Int. J. Electronic Security and Digital Forensics*, Vol. 12, No. 1, pp.77–98.

Biographical notes: Muhammad Azeem in Islamic Republic of Pakistan. He is currently a PhD scholar at Faculty of Information Technology, Beijing University of Technology, China. He has received a chancellor and vice chancellors' medal by president of Pakistan awarded by Quaid-e-Azam University Islamabad, Pakistan. He has worked in leading technology company of Pakistan as Software Design Engineer. His research interests include information security, steganography, cryptography, wireless sensor networks, machine learning, information retrieval, mobile, desktop and web applications development extended with software engineering.

Jingsha He is working as a foreign faculty member in the Faculty of Information Technology at Beijing University of Technology. His research interests include wireless sensor networks, mobile ad hoc network, WSN security and information security.

Allah Ditta received his MSc and PhD in Computer Science and Technology from the Quaid-i-Azam University (Q.A.U) Islamabad, Pakistan and the College of Computer Science, Beijing University of Technology, China, in 2012 and 2017, respectively. In 2017, he joined Division of Science and Technology and working as an Assistant Professor at University of Education, Township campus, Lahore, Pakistan. His research interests include information security, steganography, cryptography, network security protocols and wireless sensor networks.

Faheem Akhtar received his MS in Computer Science from National University of Computing and Emerging Science NUCES FAST Karachi, Pakistan. He is currently working as an Assistant Professor in the Department of Computer Science Sukkur IBA. Meanwhile he is on study leave from Sukkur IBA to pursue his PhD degree from School of Software Engineering, Beijing University of Technology (2016-2020) Beijing, China. His research interests are data mining, machine learning, information retrieval, privacy protection, internet security, internet of things and big data.

Sher Muhammad Daudpota received his Bachelor's degree in Telecommunication in 2002, followed by Masters and PhD degree in Computer Science from Asian Institute of Technology, Thailand. He joined Isra University as a Lecturer in 2002 in Computer Science department to teach data communication courses. In 2005, he join Sukkur IBA University as an Assistant Professor. Currently, he is working with same university as head of quality assurance where his role is to ensure university mission is being achieved. He has also worked on international assignments including his work with University of Massachusetts, USA for developing an Associate degree in Information Technology at Kabul Polytechnic University under University Support Workforce Development Program. His research areas, where he has contributed in shape of conference and journal publications, include big data analytics, multimedia data mining, computer vision and machine learning.

1 Introduction

In the technological era, a massive information exchange over the internet has raised the security challenges for security researchers. Internet exhibits as open medium that creates security threats (Gutub, 2015) during exchange of confidential information from source to destination. History describes two major approaches steganography and cryptography (Kumar and Sharma, 2014; Morkel, 2005; Sathiyasekar and Krishna, 2014), used for secure information exchange. Steganography techniques conceals secret data into carrier (Jackson et al., 2003) whereas cryptography is the science of analysing, ciphering, deciphering information and cryptograms (Goyal, 2012; Jirwan et al., 2013; Malhotra and Singh, 2013]. The feature of steganography and cryptography has power to arouse interest of defense departments, security researchers and protection divisions.

Securing secret information by using steganography (Cox et al., 2007) has been employed since the time of ancient Greeks. Methodologies has transformed from physical to complex steganography and cryptographic algorithms. Steganography was also suspected to be used in the attacks of World Trade Centre (Bachrach and Shih, 2011). Steganography employs different carrier medium such as text, images, audio and video (Al-Qwider and Salameh, 2017; Khan and Gutub, 2007; Nosrati et al., 2012; Singh et al., 2014). Text is considered as secure carrier and hard to decode but at the same time it's a great challenge to make it high capacitive along with robustness and data integrity (Singh et al., 2014). Cryptography, instead of concealing information, communicate data in disguised form therefore hackers or intruders cannot extract secret message from visible information.

Cryptography consist of various algorithms such as elliptic curve cryptography, RSA, RC4, RC5, DES, Safer, AES, Blowfish and CAST. Most of the cryptography algorithm either symmetric or asymmetric requires a key for encryption and decryption (Patel and Gadhiya, 2015). Generally, a hidden message is highly resistive in detection and an encoded message exhibits suspicion. Furthermore, steganography ensures the integrity and cryptography maintain the confidentiality of secret information over the network.

Steganography schemes attempt to improve carrier capacity, data security imperceptibility, and robustness. Although, text steganography exhibits low suspect rate however it provides very low data hiding capacity due to which large amount of carrier text is required in case of large size of secret message. Furthermore, to keep hidden

message secure even after sense by intruder is still very challenging task. In past, researchers worked on feature-based, inter word and inter paragraph spacing steganography in which they use one or two feature of the text to hide data (i.e., change the size of space or letter). However, in such type of solutions, capacity of carrier text remains very low (i.e., one-bits/space, one-bit/word) due to not utilising all character of cover text for embedding process and imperceptibility get disturbed due to change in feature of text. Moreover, to increase capacity, many compression algorithm were used before embedding but resulted in data loss due to non-lossless compression mechanism like Huffman encoding. In addition, lossless compression methods consist of stego key exchange overhead. To increase security, researcher used either symmetric or non-symmetric cryptography however; such cryptographic techniques put extra burden in term of encryption time and secure exchange of secret key. Therefore, in the age of Nano technologies, to get over flaws and ambiguities in formerly proposed solutions, a novel algorithm is required to deal the challenges of text steganography like carrier capacity, stego text size and data security along with imperceptibility and robustness.

Recent work has performed a comprehensive analysis, experiments and verbalises a novel approach for increasing carrier capacity with controlled stego text size. The study introduces an encrypted steganography approach that swank high capacitive, robust and secure algorithm with minor increase in stego text size for protection of confidential information. The projected approach uses conventional cryptography technique that employs four-layer encryption including bit complement, one-bit insertion and two-time bit rotation. This pattern of encryption strengthens the algorithm with no overhead of secret key generation and exchange from sender to receiver. Moreover, seven Unicode's incorporating invisible plus (IP), invisible separator (IS), inhabit symmetric swapping (ISS), left to right override (LRO), zero width non joiner (ZWNJ), zero width joiner (ZWJ), zero width non-joiner and zero width character (ZWC) are exercised to embed secret bits into text carrier. Even both steganography and cryptography provide security to confidential data but the hybrid implementation of these methods result into greater protection of data. Further, identical input and output text reduces the suspension of private information over the network.

A deep study on text carrier expresses that it contains less number of redundant bits as compare to other carrier such as audio, video and image, therefore text carrier is more challenging (Odeh et al., 2012). Main objective of current study is to handle the text carrier precisely and obtain high carrier capacity. Experimental results are tremendous and remarkable because for the first time, carrier capacity has been raised up to 300%. All characters in the text are wisely utilised and each character can hide 3-bits per character. In addition, instead of employing complex modern encryption mechanism having overhead of encryption key and secure exchange, four layers of conventional encryption keeps the algorithm simple and effective. Seven different Unicode to disguise data into cover text promise the strength of algorithm and objectives of projected research like carrier capacity, controlled stego size, security, integrity, confidentiality, robustness and avoidance of visual attacks accomplish successfully.

The rest of the paper is organised as follows: Section 2 explains related work and weaknesses of formerly proposed schemes. Section 3 provides a detail description of proposed algorithm. Section 4 articulates the experiments and analysis along with obtained results from the proposed approach and comparative discussion with former suggested approaches and finally, Section 5 presents conclusion of the work.

2 Related work

There exist numerous steganographic approaches which employ different languages but English and Arabic text have been point of concentration for researchers (Odeh et al., 2012; Mahato et al., 2014; Memon et al., 2018; Por and Delina, 2008; Shirali-Shahreza, 2008; Shirali-Shahreza and Shirali-Shahreza, 2007). Text steganography approaches includes open spacing, sentence spacing, inter word spacing, merging ZWC with space, hybrid approach, null-based, line and word shifting, feature and abbreviation encoding and point shifting techniques (Memon et al., 2018; Singh et al., 2009; Hariri et al., 2011; Muhammad et al., 2017).

Shirali-Shahreza et al. (2008) presented a novel algorithm for linguistic steganography. They employed feature coding by using Arabic word 'LA لا' for the concealment of data and presented bit-one as regular 'LAM 'ل'' with 'ALIF 'ا'' and bit-zero as commuted 'LA لا' (i.e., a little extension between LAM' 'ALIF'). The resulting stego had high suspect rate due to unmatched cover text and uneven visibility in printing text. Furthermore, all words could not conceal data due to which hiding capacity of carrier was low.

Another data hiding approach by using space character (Steganographic Algorithm, 2017) was proposed by Bender et al. They introduced single-space as bit-one and double-space as bit-zero. This method articulated one-bit/word capacity that was significantly low because data was concealed in space characters therefore, more number of spaces were required in case of large secret message. Moreover, adding a new space character had effect on stego text size. Carrier and stego text also exhibited un-identical visual appearance.

Por and Delina (2008) introduces a hybrid approach by using inter-word and inter-paragraph spacing. Although this technique produced undistinguishable stego but capacity of carrier was very low because only spaces were utilised and large confidential data required large amount of carrier. Furthermore, algorithm was not secure, once stego got suspect; data can be retrieved with minor effort.

Satir et al. recommended novel technique based on LZW compression and stego keys (Bender et al., 1996) for data concealing process. The approach utilised both steganography and cryptography to enhance capacity and security. LZW coding increased the capacity, confidentiality and practiced on forward mail platform for data hiding. Stego key was mandatory to extract hidden message on receiver side. Furthermore, the process of confidential message extraction was complex due to LZW encoding and compression. Stego key exchange mechanism could be compromised.

Vidhya et al. summarised steganography with Malayalam text carrier (Satir and Isik, 2012). They implemented custom Unicode for data hiding. Although, the method was precise but confidentiality was not ensured and had overhead because database was mandatory. Carrier capacity was also unfocused.

Singh et al. (2014) focused on successive RGB colour substitution. They altered colour of Unicode character to next RGB value. This mechanism could use two colours so each character can hide only one bit. Although, it means only one bit can be represented by an alphabet but space characters were remain non-utilised. In addition, printing text could cause visual imbalance and message can be suspected very easily.

Vidhya and Paul (2015) addressed security and capacity by compressing data with Huffman encoding. They used forward email platform as carrier therefore more number

of emails were required to conceal large amount of private data. Furthermore, Huffman encoding was considered as a data loss approach.

Ditta et al. revealed Null Cipher steganography (Satir and Isik, 2014) by using English as carrier. Confidential message was concealed by using letter at specific index in each word. Any letter in the word could be targeted character (e.g., first, last, second, or third, etc.). It was a better approach but carrier capacity was too low as one character per word and character sequence could be identified using different steg-analysis tool. Hence, there was no data integrity and security. Here again, big size of confidential message required large amount of cover text.

A format-based algorithm (Ditta et al., 2015) was suggested by Rajeev et al in which they implemented Unicode space and regular space characters by using inter-word, end of line, inter-paragraph and inter-sentence space of MS Word carrier. Data was concealed by embedding Unicode space with regular space and Unicode space size was reduced up to 6.em to maintain the visual similarity. Here, carrier capacity was increased due to multiple targeted places and to hide more confidential message. This scheme increase capacity a little more by targeting more places in text but lot of inter word, end of line, inter-paragraph spaces were required. Stego text size ratio also increased significantly

Another format-based approach (Kumar et al., 2015) was stick in by Mahato et al. in which they exercised variation in whitespace character of MS Word document. According to the technique, deviated and regular font size of space reflected as bit-one and bit-zero respectively. The result showed that carrier capacity was one bit per word so substantial number of spaces were required to hide long message. Hence carrier capacity remained very low. Furthermore, there was no data integrity and security if hackers change space.

Malik et al. presented a colour coding scheme (Mahato et al., 2014) which was practiced on email text carrier. Binary bits were revealed by constructing a table of ten colour from where five colours were mapped as one and five colours were mapped as zero. Carrier capacity of this algorithm was one bit per character. Major drawback of this approach was rich coloured text that could attract intruders immediately.

Ditta et al also stated a linguistic steganography approach (Memon et al., 2018) by using Arabic text as carrier. They used Unicode characters to embed confidential data. They achieved average capacity about 1.6 bits per character in the carrier. They did not used the last character of each word and whitespaces. Besides, there was no resistance to security attacks. Moreover, all characters in cover text were not utilised in embedding process therefore capacity remained low.

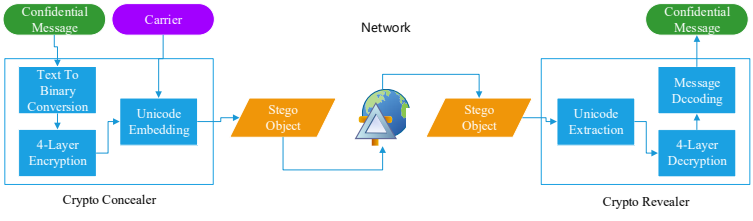
The above presented related work illustrates that previously proposed schemes requires large amount of carrier text to conceal large size of confidential message which causes low carrier capacity. Another problem is, none of the previous schemes utilise all characters in carrier due to which capacity remains low. Furthermore, while using some special characters for embedding purpose, stego size also increases significantly which results into high suspect rate. Many schemes exhibits imperceptibility problems due to change in feature of text therefore non-identical carrier and stego text causes visual attack. In addition, there exist no security of secret message if hidden message is sensed by intruders and existing scheme with security utilises complex cryptography functions that is considered and extra burden.

3 Proposed algorithm

In proposed algorithm, crypto concealer side converts confidential message into binary string, applied 4-layer encryption, embed binary into carrier text by using seven different Unicode and generated a stego object. The sender transmits this stego object over the network. On the receiver side, crypto revealer extracts Unicode, regenerates corresponding binary value and applies 4-layer decryption. Finally, deciphered binary is converted into English text. Equation (1) illustrates the generation of stego object and Figure 1 demonstrates the mechanism of proposed algorithm for both crypto concealer and crypto revealer.

$$\text{Stego Object} = \text{Mapped Unicodes} + \text{Cover Text} \tag{1}$$

Figure 1 Basic mechanism of proposed algorithm (see online version for colours)



In notepad, some Unicode characters demonstrate non-visibility. Present research utilises this remarkable feature to represent secret bits in cover text. Moreover, secret message is secured by four layers of encryption and then concealed into cover text. This section explains working the proposed crypto concealer and crypto revealer algorithm in detail.

3.1 Methodology

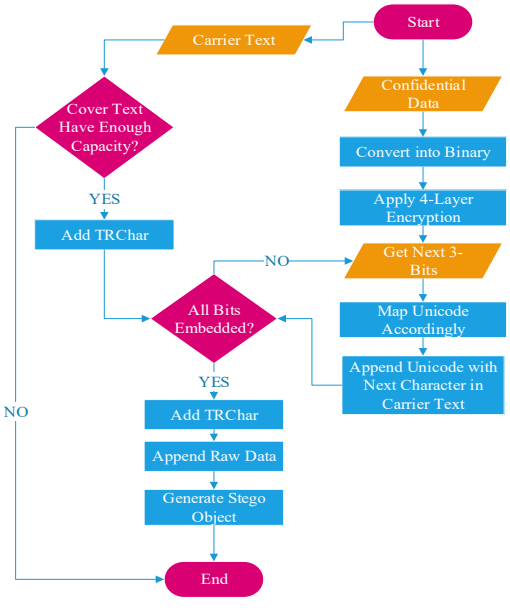
In steganography, to avoid visual attacks, carrier and stego object must be indistinguishable. Notepad text has limited feature therefore, it is challenging to keep cover and stego object identical while increasing cover text capacity. This study has done a deep analysis and found seven Unicode to cope the described challenge. Each Unicode represents three-bit tuple. Table 1 contains list of Unicode and corresponding bit-tuple, which are used in the formation of algorithm.

Table 1 Bit-tuple representation of Unicode

Unicode	*Abbreviation*	*Represents (bit tuple)*
Empty string	“”	000
Zero width character	ZWC	001
Zero width joiner	ZWJ	010
Zero width non-joiner	ZWNJ	011
Invisible plus	IP	100
Invisible separator	IS	101
Inhabit symmetric swapping	ISS	110
Left to right override	LRO	110

The scheme intelligently utilised above mentioned Unicode to attain high capacity and visual similarity of carrier and stego text. Figure 2 elaborates the detailed process of proposed crypto concealer algorithm.

Figure 2 Flow chart of crypto concealer algorithm (see online version for colours)



3.2 Four layer security

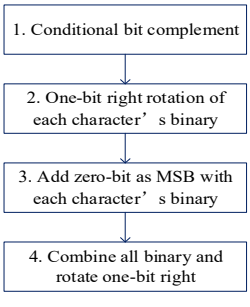
In data concealing process, algorithm applies conventional encoding by converting each character of secret message into binary and get complement on surpass characters as elaborated in our previous work (Muhammad et al., 2017) as first layer of encryption. Surpass characters are those which contains ones count greater than or equal to four whereas non-surpass characters are those whose binary one's count is less than four. Table 2 briefly demonstrates some examples of surpass and non-surpass characters.

Table 2 Examples of surpass and non-surpass characters

Character	*Binary value*	*Ones count*	*Is surpass?*
A	01000001	2	No
A	01100001	3	No
F	01100110	4	Yes
O	01101111	6	Yes
8	00111000	4	No
Semicolon (;)	00111011	5	Yes

For example, there is a secret character 'o' with binary value '01101111' need to conceal. Here, we can see it is a surpass character therefore complement is applied as '10010000'. In the next step, binary of each character is rotated one-bit right like '01001000', that is second layer of encryption. At third layer of security, a zero-bit is inserted as MSB in each character as '001001000' and binary of all secret character are combined. There are two main reasons of inserting zero, one is to increase security and other is to make total bit count of each character as nine. Each Unicode represents three-bit tuple so bit count of each character must be a multiple of three. Binary of each character consist of eight bits so algorithm insert a zero bit for making three tuples of each character. During embedding process, zero tuple does not need any Unicode to be inserted therefore approach insert zero instead of one. Furthermore, concluded binary value is further applied one-bit right rotation as forth layer of security. By using this approach, algorithm ensures the security and confidentiality of secret message. Block diagram of 4-layer security process is shown in Figure 3.

Figure 3 Block diagram of 4-layer security process of proposed algorithm



3.3 Embedding process

After applying four-layer encryption, algorithm starts embedding process, reads next three-bit tuple from the secret bits and map the tuple with Unicode as per defined in Table 1. For example, binary value '001001000' need to be hided. Algorithm read next three bits '001' from the binary and map with Unicode that is ZWC. In next step it gets next character from the cover text (e.g., 's') and append selected Unicode with this character (i.e., s + ZWC) and rebuild a new string that is called stego string. This phenomenon is repeated until the completion all bits concealment. Furthermore, it inserts data tracking bits (TRChar), some raw bits into cover text as explained (Muhammad et al., 2017) and finally generates a text file that is called stego object. Following equation (2) describes number of bits of secret message to hide in cover text.

$$\text{TBH} = \text{SMC} * 9 \tag{2}$$

where TBH refers to total bits to hide, SMC represents secret message characters and constant '9' corresponds to binary bits per character.

3.4 Algorithm for crypto concealer

As shown in Figure 1, proposed algorithm has two parts (i.e., crypto concealer and crypto revealer). Crypto concealer is responsible for encryption and embedding process at sender's side. Primarily, algorithm converts English message into binary string and put on four-layer security. Following is pseudo code of four-layer security.

```
string ApplyFourLayerSecurity(originalBinaryString)
{
Set finalBitstoHide to empty
Set chunkSize to Eight (8)
Set stringLength to originalBinaryString.Length
Set counter to zero (0)
While counter is less than stringLength
    If counter + chunkSize greater than stringLength
            Set chunkSize to stringLength-1
            Set chunk to next eight bits in BinaryString
        If onesCount in chunk is greater or equal to four
            Take complement of the chunk
            Rotate chunk one bit to right + insert '0' as MSB
            append to finalBitstoHide
        Else
            Rotate chunk one bit to right + insert '0' as MSB
Append chunk to finalBitstoHide
Add one to counter
Rotate finalBitstoHide one bit to right
return finalBitstoHide
}
```

Above algorithm applies four layer security and proceeds for concealment process. Algorithm reads next three bits from the bits string, next character from carrier and appends mapped Unicode with carrier character. If 3-bit tuple is '000' then append tpl000. tpl000 denotes empty string. Algorithms maps tuple according to the Unicode in Table 1 and append with carrier English characters. A brief pseudo-code chunk for concealing bits is given below.

```
Set StegoText to empty
Set finalBitstoHide to '01100001'
Set carrierTextString to 'hi, I am not a musician?'
Foreach (nextChar in coverTextString)
{
    Set nextTuple to GetNextTuple (finalBitstoHide)
    If nextBitPair is equals to '000'
        Append nextChar + tpl000 to Stegotext
```

```
    Else if nextBitPair is equals to '001'
        Append nextChar + tpl001 to Stegotext
    Else if nextBitPair is equals to '010'
        Append nextChar + tpl010 to Stegotext
    Else if nextBitPair is equals to '011'
        Append nextChar + tpl011 to Stegotext
    Else if nextBitPair is equals to '100'
        Append nextChar + tpl100 to Stegotext
    Else if nextBitPair is equals to '101'
        Append nextChar + tpl101 to Stegotext
    Else if nextBitPair is equals to '110'
        Append nextChar + tpl110 to Stegotext
    Else // if nextBitPair is equals to '111'
        Append nextChar + tpl111 to Stegotext
}
```

For illustration, suppose there is a cover string 'Hi, I am not a musician?' and confidential data bits after applying four layer security are '111000000'. Following steps demonstrate the crypto concealer algorithm in detail.

Step 1 Input cover text and calculate carrier capacity that is CC = CTC * 3 = 23 * 3 = 69 and concealing tuple count is three (i.e., 111,000,000). Therefore, carrier has enough capacity to hide encrypted bits.

Step 2 Append tracking characters (TRChar) into carrier at the start of embedding procedure.

Step 3 Get next character from carrier text that is 'H' and next bit-tuple from input bits that is '111'.

Step 4 Append tpl111 with 'H' such as 'H + tpl111'.

Step 5 Check, if there are more secret bits to hide, read next character from carrier that is 'i' , take next tuple from input secret bits that is '000'.

Step 6 Append tpl000 with 'i' such as 'i + tpl000'.

Step 7 Again , Check if still more bits to hide, get next character from carrier which is comma ',' and take next bit-tuple from secret bits that is '000'.

Step 8 Append tpl000 with ',' such as ', + tpl000'.

Check again whether all bits has been hidden or not? Here, there is no more secret bits to hide so concealing process is accomplished. Now algorithm terminates the loop and proceeds for next step.

Step 9 Append TRChar for keeping track of hidden data.

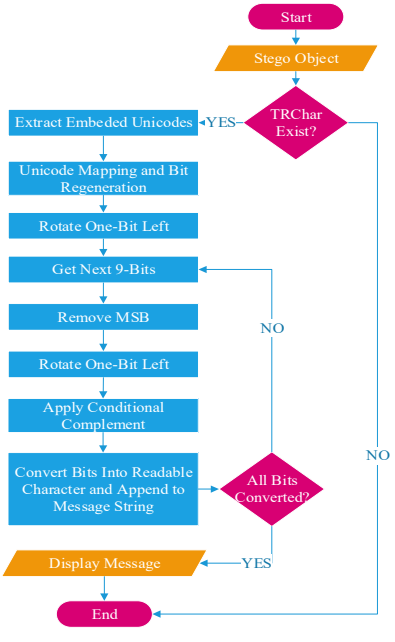
Step 10 Algorithm appends some raw Unicodes in rest of carrier text to ensure the integrity of data.

Step 11 Finally, algorithm generates stego object by writing stego text in notepad file.

3.5 Algorithm for crypto revealer

Crypto revealer is second part of proposed algorithm that is responsible for secret message extraction from stego object. Crypto Revealer works in reverse order of crypto concealer. Figure 4 illustrates crypto revealer algorithm in detail.

Figure 4 Flow chart of crypto revealer algorithm (see online version for colours)



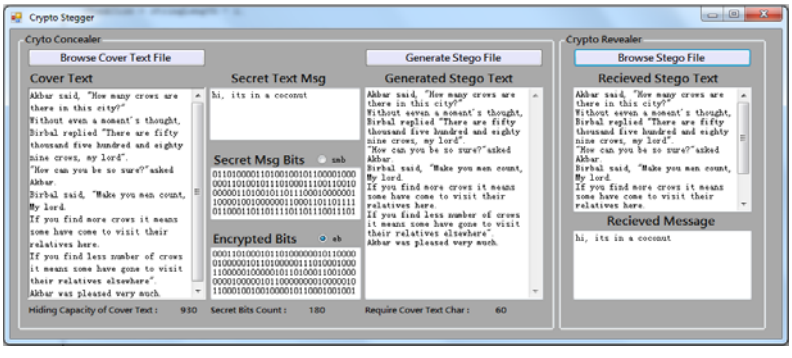
Initially, revealer algorithm checks the presence of TRChar. The presence of TRChar in stego object ensures the existence of secret message. Hence, bits extraction process is started. Unicode characters are retrieved from stego text. A binary string is built by mapping retrieved Unicodes to corresponding binary tuple as described in Table 1. Here bit consistency and data integrity is ensured by taking mod by nine because 9-bits represent a secret character in embedding algorithm. The mod results other than zero describes that data integrity has been compromised. For instance, retrieved bit count is 36 and 36 mod 9 = 0 demonstrates data integrity. After this, retrieved bits are further processed for four-layer security decryption to generate confidential message.

4 Experiments and analysis

Current section contains a detail discussion experiments and analysis on the result of proposed scheme. An application named as 'Crypto Stegger' was developed in C#.Net 4.0 and extensive experiments has been performed to test the proposed algorithm. Crypto stegger contains two main parts i.e., crypto concealer and crypto revealer as shown in Figure 5. As articulated earlier, the current study utilised the combine power of cryptography and steganography. There exist three major metrics to measure the effectuality of a recommended steganography algorithm: concealing capacity of cover text, impressibility and robustness (Malik et al., 2017). Additionally, stego file size is also a considerable factor to reduce suspect rate.

- *Concealing capacity:* this term states the maximum number of secret bits that a carrier text can hide.
- Stego text/file/object size: the size of text/file/object after embedding secret bits is known as stego files size. The less increase in stego size ensures the less suspect rate and more security of the algorithm.
- *Impressibility:* it articulates that human sight or any other statistical method cannot perceive the hidden information. This is a significant feature due to which person's eye unable to identify, differentiate or detect confidential data in stego object.
- *Robustness:* it refers to the ability of the technique to retrieve the hidden data successfully and expose to recipient if data has compromised by intruder.

Figure 5 GUI of crypto stegger (see online version for colours)



Therefore, this algorithm was tested for embedding capacity of carrier, stego text size, identical visual appearance of carrier and stego object, robustness and the security of confidential data.

4.1 Carrier text capacity

Current era belongs to nano-technologies and devices has been decreased in size and increased in capacity. Similarly, small carrier text must hold large confidential data to lessen traffic burden over the network. Proposed algorithm explains that each character in cover text can hide three bits of secret data. Therefore, hiding capacity of cover text is 3-bits/character. Following equation (3) expresses the concealing capacity of cover text.

$$CC = CTC * 3 \tag{3}$$

where CC describes concealing capacity of cover text, CTC correspond total characters in cover text and 3 is constant multiplier as each character can hide 3-bit tuple. According to above equation (3), concealing capacity of the cover text 'I am not a musician' will be as below.

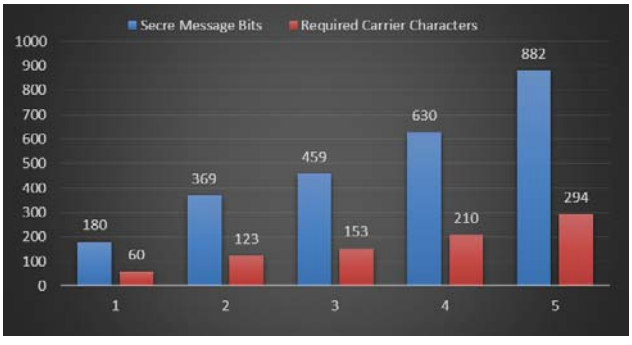
$$CC = 19 * 3 = 57$$

The proportion of data hiding has risen from bits/word to 3-bits/char. Therefore, a significant increase in carrier capacity has been gained and study has achieved an important milestone as embedding capacity of cover text has been enhanced up to 300%. Following equation (4) shows the computation of embedding capacity in percentage.

$$Embedding\ \ Capacity = \frac{\text{Number of Secre Bits to Hide}}{\text{Required Number of Cover Text Characters}} \tag{4}$$

The proposed novel research can hide three bits in each character including whitespace and punctuation marks and numbers. Hiding capacity graph of the present approach is shown in Figure 6.

Figure 6 Carrier text capacity of proposed scheme (see online version for colours)



According to above results and embedding equation (4), percentage-embedding capacity of proposed scheme is as below.

$$Embedding\ \ Capacity = \frac{180}{60} * 100 = 300\%$$

Experimental results revealed that suggested algorithm overshadowed on the existing text steganography approach based on two-letter word (TWL) (Sumathi et al., 2014) proposed by Baawi et al. TWL scheme split the text into two letter word and employs ZWNJ Unicode for secret bit embedding. Moreover，TWL approach did not use all characters in carrier text but projected approach successfully utilised all characters in carrier. The study performed extensive experiments to evaluate embedding capacity. The critical analysis showed that embedding capacity of suggested scheme was extremely high as compared to TWL scheme. Figure 7 shows embedding capacity comparison graph of TWL approach and current approach. The graph clearly states the tremendous increase in embedding capacity, which was considered as a prime objective of the research.

Figure 7 Comparison of embedding capacity of ALW and proposed scheme (see online version for colours)

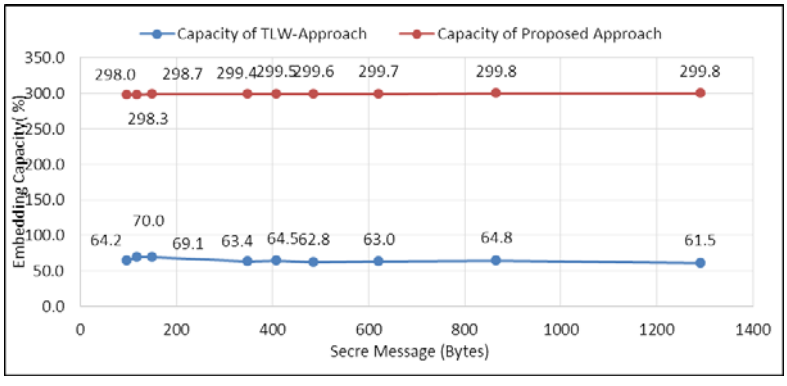
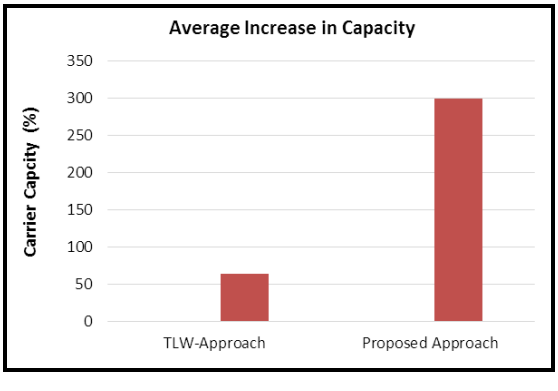


Figure 8 Average embedding capacity comparison (see online version for colours)



The average capacity increased by TWL techniques was 64.54% whereas proposed algorithm has enhanced average capacity of text carrier up to 299.28. The average capacity of both approaches for average input secret message (706 bytes) is shown in Figure 8.

4.2 Stego file size analysis

Stego text size is an important and highly focused factor of steganography. To avoid suspicion, carrier and stego text should have minimum difference in size so that intruder or attacker may not sense secret data. In present approach, conditional bit complement increases the number of zero-bit in secret message due to which less number of Unicode are embedded to conceal data. Consequently, the significant increase in stego text size is very low. Table 3 illustrates the secret bit count and embedded bit count to conceal into cover text. Furthermore, it shows that average secret bits to embedded Unicode ratio is 0.229 that justify a minor increase in stego text.

Table 3 Concealed bits to embedded Unicode ratio

Exp. no.	*Concealed bit count*	*Embedded Unicode count*	*Ratio*
1	90	21	0.233
2	180	41	0.228
3	270	61	0.226
4	369	85	0.230
5	450	104	0.231
6	540	125	0.231
7	630	142	0.225
8	720	165	0.229
9	810	184	0.227
10	900	204	0.227
	Average secret to embedded Unicode ratio		0.229

To find percentage increase in stego text size, the study has formulated equation (5) that is given below.

$$\text{Increase in Stego Size} = \frac{(\text{Stego Text Size} - \text{Carrier Text Size})}{\text{Carrier Text Size}} * 100 \tag{5}$$

Results revealed that increase in stego text size with current scheme is very low. Figure 9 explains the percentage increase in stego text.

Figure 9 Percentage increase in stego file created by proposed approach (see online version for colours)

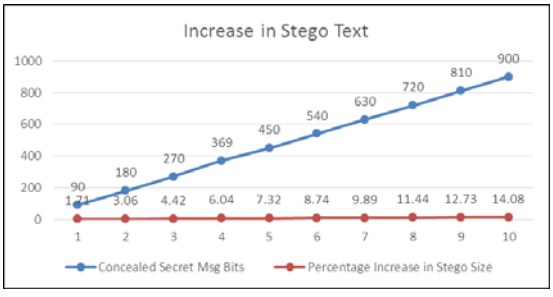
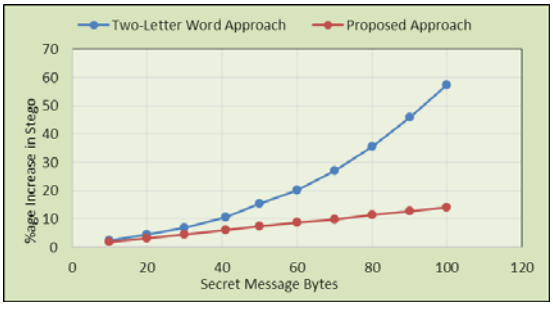


Figure 9 evidently states the success of present algorithm as it implies a minor increase in stego text with the major increase in input confidential message bits. A deep comparative analysis of TLW approach with proposed approach has been done with respect to stego file increase. It is concluded that increase in stego text size suggested approach is significantly less as compared to TLW scheme. Figure 10 demonstrates the stego text increase in TLW and proposed approach.

Figure 10 Comparison of TLW and proposed approach w.r.t increase in stego file size (see online version for colours)



4.3 Imperceptibility

Seven Unicode were employed to embed secret information into carrier text. These were selected after detailed study of Unicode characteristics. The embedded Unicode does not affect the visual appearance of the cover text because they remain invisible and does not occupy any space in the notepad file. Therefore, the imperceptibility of the suggested algorithm is very high due to identical cover and stego text. Human eye cannot detect the hidden message in the carrier text. Visual similarity of carrier and stego text can be clearly seen in Figure 11. The confidential information access is stringently restricted to the accredited recipient.

Figure 11 Imperceptibility: optical similarity of cover text and output text (see online version for colours)

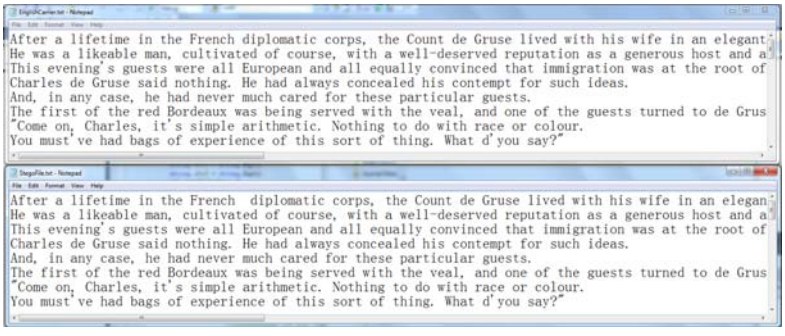


Table 4 contains a comparative analysis of the formerly proposed schemes with current proposed scheme. Although colour-coding technique conceal one-bit per character but does not utilise space characters. Moreover, it cannot resist visual attacks due to rich colour-coding in stego text. Therefore, carrier and stego text remain non-identical. Hybrid scheme conceals four data-bits per whitespace lies between two words or two paragraphs. Hence, carrier and stego object exhibits same visual appearance but carrier capacity remains low because it does not incorporate all characters in carrier. Two Letter Word approach resist against visual attacks but capacity remain letter in words-1 per word. Therefore, whitespaces and many other characters do not utilised. In contrast, suggested algorithm has advantage in all sectors including visual appearance, resistance against attacks, security, confidentiality and high carrier capacity because it employs four-layer encryption and each character of the carrier text can conceal three secret bits.

Table 4 Comparative analysis of previous approaches and proposed approach w.r.t. visual appearance and carrier concealing ability

Approaches	*Capacity*	*Visual appearance*	*Resistance against visual attacks*	*Can use all character in carrier to conceal?*
Regular space with unicode space	1 bit/space	Non-identical	No	No
Hybrid text steganography	4 bit/space	Identical	Yes	No
Double space approach	1 bit/space	Non-identical	No	No
Null steganography	1 bit/word	identical	Yes	No
Colour coding approach	1 bit/character	Non-identical	No	No
TLW scheme	(Letter count in word) – 1/ word	Identical	Yes	No
Proposed approach	3 bit/character	Identical	Yes	Yes

4.4 Analysis of security and robustness

The term security illustrates that only authorised recipient can receive the confidential information. The resistive capacity of an approach to different steg-analysis attacks for accessing confidential information without having permission is robustness. For this purpose, algorithm embeds tracking characters (TRChar) during data concealment process.

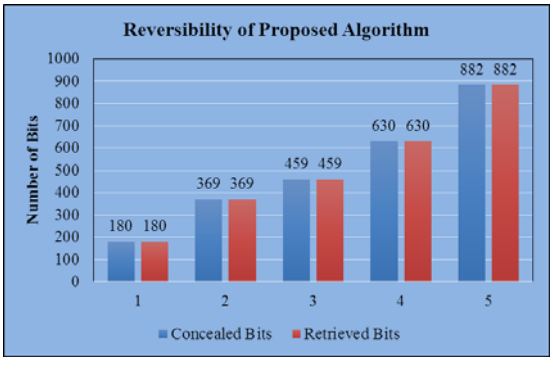
Experiments revealed that four-layer encryption has significant impact on data security. If in case, secret data is sensed still intruders cannot retrieve the confidential message. Therefore, algorithm reflects high security against attacks, ensure confidentiality of data and robustness. Furthermore, if a hacker attempts to temper the stego text then embedding TRChar and Unicodes will be disturbed that can indicate illegal action on receiving side.

Results also revealed that reversibility of algorithm is very high. Reversibility refers to the power of proposed algorithm to retrieve hidden bits and regenerate confidential information successfully on receiver side. Unlike other data compression techniques, suggested scheme has no deficiency of data loss. The bit error rate (BER) of crypto revealer algorithm is zero. BER defines the correction capabilities of a data-hiding scheme and is the ratio (in percentage) between concealed bit count and successful retrieved bit count (Baawi et al., 2017). Bit Error Rate can be can be calculated as given in below equation (5):

$$\text{Bit Error Ratio (BER)} = \frac{\text{Retrieved Bits} - \text{Concealed Bits}}{\text{Concealed Bits}} * 100 \tag{5}$$

Reversibility and bit error ratio is shown in Figure 12. It contains the graph of embedding and retrieving bits on sender and receiver side respectively with different input secret message size.

Figure 12 Reversibility of proposed algorithm (see online version for colours)



5 Conclusions

Currently proposed novel scheme employs seven Unicode such as invisible plus (IP), invisible separator (IS), inhabit symmetric swapping (ISS), LRO, (ZWNJ), zero width joiner (ZWJ), zero width non-joiner and ZWC for the concealment of confidential data into English text carrier. The mentioned Unicode exhibits a significant property of invisibility due to which they remain imperceptible after embedding into carrier text. Each Unicode character represents three bits data. Before hiding data, Algorithm does apply four-layer encryption on secret information to produce cipher text. To achieve the encryption objective, conditional one's complement is applied as layer-1; one-bit right rotation is applied on binary of each character as layer-2; zero bit is added with MSB as layer-3; finally, binary of all secret information is combined and one-bit right rotation is applied as layer-4.

Experimental results exposed that hiding capacity of carrier has been raised to 300% due to hiding three bits per each character. Equation (CC = CTC * 3) is formulated to compute hiding capacity. Proposed algorithm has achieved steganography parameter like capacity, imperceptibility, security and robust successfully. Moreover, the visually identical cover and stego object reduce the suspect rate significantly.

References

Al-Qwider, W.H. and Salameh, J.N.B. (2017) 'Novel technique for securing data communication systems by using cryptography and steganography', *Jordanian Journal of Computers and Information Technology (JJCIT)*, Vol. 3, No. 2, pp.110–130.

Baawi, S.S., Mokhtar, M.R. and Sulaiman, R. (2017) 'New text steganography technique based on a set of two-letter words', *Journal of Theoretical & Applied Information Technology*, Vol. 95, No. 22, pp.6247–6255.

Bachrach, M. and Shih, F.Y. (2011) 'Image steganography and steganalysis', *Wiley Interdisciplinary Reviews: Computational Statistics*, Vol. 3, No. 3, pp.251–259.

Bender, W. et al. (1996) 'Techniques for data hiding', *IBM Systems Journal*, Vol. 35, Nos. ¾, pp.313–336.

Cox, I. et al. (2007) *Digital Watermarking and Steganography*, Morgan Kaufmann, San Francisco, CA, USA.

Ditta, A. et al. (2015) 'Using different techniques of steganography deducting null cipher in plain text', *International Journal of Wireless and Mobile Computing*, Vol. 9, No. 4, pp.317–324.

Goyal, S. (2012) 'A survey on the applications of cryptography', *International Journal of Science and Technology*, Vol. 1, No. 3, pp.137–140.

Gutub, A. (2015) 'Social media & its impact on e-governance', *ME Smart Cities 2015-4th Middle East Smart Cities Summit*.

Hariri, M., Karimi, R. and Nosrati, M. (2011) 'An introduction to steganography methods', *World Applied Programming*, Vol. 1, No. 3, pp.191–195.

Jackson, J.T. et al. (2003) 'Blind Steganography detection using a computational immune system: a work in progress', *International Journal of Digital Evidence*, Vol. 4, No. 1, p.19.

Jirwan, N., Singh, A. and Vijay, D.S. (2013) 'Review and analysis of cryptography techniques', *International Journal of Scientific & Engineering Research*, Vol. 4, No. 3, pp.1–6.

Khan, F. and Gutub, A. (2007) 'Message concealment techniques using image based steganography', *The 4th IEEE GCC Conference*, Gulf International Convention Centre, Bahrain, 11–14 November.

Kumar, P. and Sharma, V.K. (2014) 'Information security based on steganography & cryptography techniques: a review', *International Journal*,Vol. 4, No. 10, pp.246–250.

Kumar, R., Chand, S. and Singh, S. (2015) 'An efficient text steganography scheme using unicode space characters', *International Journal of Forensic Computer Science*, Vol. 10, No. 1, pp.8–14.

Mahato, S., Yadav, D.K. and Khan, D.A. (2014) 'A novel approach to text steganography using font size of invisible space characters in Microsoft Word Document', in *Intelligent Computing, Networking, and Informatics*, Springer, pp.1047–1054.

Mahato, S., Yadav, D.K. and Khan, D.A. (2014) *A Novel Approach to Text Steganography Using Font Size of Invisible Space Characters in Microsoft Word Document*, Springer India, pp.1047–1054.

Malhotra, M. and Singh, A. (2013) 'Study of various cryptographic algorithms', *IJSER*, Vol. 1, No. 3, pp.77–88.

Malik, A., Sikka, G. and Verma, H.K. (2017) 'A high capacity text steganography scheme based on LZW compression and color coding', *Engineering Science and Technology, an International Journal*, Vol. 20, No. 1, pp.72–79.

Memon, M.Q. et al. (2018) 'Information hiding: Arabic text steganography by using unicode characters to hide secret data', *International Journal of Electronic Security & Digital Forensics*, Vol. 10, No. 1, p.61.

Morkel, T. (2005) *Steganography and Steganalysis*, ICSA Research Group. University of Pretoria, South Africa.

Muhammad, A. et al. (2017) 'A secure and size efficient approach to enhance the performance of text', *International Conference on Computer Systems, Electronics and Control (ICCSEC)*, pp.0146–0150.

Nosrati, M., Karimi, R. and Hariri, M. (2012) 'Audio steganography: a survey on recent approaches', *World Applied Programming*, Vol. 2, No. 3, pp.202–205.

Odeh, A. et al. (2012) 'Steganography by multipoint Arabic letters', *Systems, Applications and Technology Conference (LISAT)*, IEEE Long Island, IEEE.

Patel, Z. and Gadhiya, S. (2015) 'A survey paper on steganography and cryptography', *RESEARCH HUB – International Multidisciplinary Research Journal (RHIMRJ)*, Vol. 2, No. 5, pp.835–838.

Por, L.Y. and Delina, B. (2008) 'Information hiding: a new approach in text steganography', *7th WSEAS Int. Conf. on Applied Computer & Applied Computational Science (ACACOS '08)*, pp.689–695.

Sathiyasekar, D.K. and Krishna, S.K.S. (2014) 'A research review on different data hiding techniques', *International Journal of Engineering and Computer Science*, January, Vol. 3, No. 1, pp.3655–3659.

Satir, E. and Isik, H. (2012) 'A compression-based text steganography method', *Journal of Systems and Software*, Vol. 85, No. 10, pp.2385–2394.

Satir, E. and Isik, H. (2014) 'A Huffman compression based text steganography method', *Multimedia Tools & Applications*, Vol. 70, No. 3, pp.2085–2110.

Shirali-Shahreza, M. (2008) 'A new persian/arabic text steganography using 'la' word', *Advances in Computer and Information Sciences and Engineering, Proceedings of the 2007 International Conference on Systems, Computing Sciences and Software Engineering*.

Shirali-Shahreza, M. and Shirali-Shahreza, M.H. (2007) 'Text steganography in SMS', *IEEE/IFIP International Conference in Central Asia on Internet*.

Singh, H., Diwakar, A. and Upadhyaya, S. (2014) 'A novel approach to text steganography', *International Proceedings of Computer Science and Information Technology*, Vol. 59, p.7.

Singh, H., Singh, P.K. and Saroha, K. (2009) 'A survey on text based steganography', *Proceedings of the 3rd National Conference*.

Steganographic Algorithm (2017) *International Conference on Computer Systems, Electronics and Control (ICCSEC)*, Part A, p.5.

Sumathi, C., Santanam, T. and Umamaheswari, G. (2014) 'A study of various steganographic techniques used for information hiding', *International Journal of Computer Science & Engineering Survey*, Vol. 4, No. 6, pp.9–25.

Vidhya, P. and Paul, V. (2015) 'A method for text steganography using Malayalam text', *Procedia Computer Science*, Vol. 46, pp.524–531.