XORSTEG: A New Model of Text Steganography

Tapodhir Acharjee, Ashish Konwar, Rakesh Kumar Ram,Rahul Sharma, Dhrubajyoti Goswami
Department of Computer Science and Engineering
Assam University, Silchar, Assam, India
tapacharjee@gmail.com, ashish.konwar@gmail.com, rakesh.it672@gmail.com, rahul.shrma026@gmail.com, dhrubajg305@gmail.com

***Abstract*—Staganography is the practice of hiding secret information using texts, images and audio as the cover media. But, using text as the target medium is not as easy as compared to the other target media, because of the lack of available redundant information in a text file.This paper proposes a new model of key-based text steganography. Various methods in this type of key-based steganography do not work when the cover file contained single letters and words which have the same starting and ending letter. The proposed approach works on this limitation and conceals a message, without degrading cover, by using XOR operation on the start and end letter of words of the cover.**

Keywords—steganography; text steganography; key-based text steganography; cover file

I. INTRODUCTION

Steganography is a word stemming from the ancient Greek with the meaning of covered writing. Steganography uses the help of a cover file (text, image, audio,video)to hide a secret message. The secret message is embedded with the cover message by using various methods and is then sent to the intended recipient. In steganography the text to be concealed is called embedded data. An innocuous medium, such as a text, audio, image, etc. used to hide the message is called the cover file. The key which is optional and is used in the embedding process is called a stego key. A stego key is used to control the hiding process so as to restrict detection from third parties and/or recovery of embedded data to the parties who know it [1, 4].

Steganography which uses text as a cover medium is termed as text steganography. Text steganography can involve anything from changing the formatting of an existing text, to changing words within a text, to generating random character sequences or using context-free grammars to generate readable texts[5]. The formatting of a text can easily draw attention from third parties if the original plaintext is available [5].

This paper presents an approach of text steganography which does not change the format of the cover text and hides the secret message in the cover file by generating a stego key. A method for this type of key based steganography is proposed in [4]. But it is limited in the way that it cannot hide the message when the cover file contains letters having same starting and ending letter. Our proposed model overcomes this limitation which increases the efficiency of the approach.

The rest of this paper is organised as follows: Section II discusses the existing method and its limitations. Section III describes the proposed approach. Section IV shows the results and compares our proposed approach with the existing approach. The paper concludes with the conclusion in Section V.

II. RELATED STUDY

In this section we discuss an existing text steganography method[4] which uses any paragraph and hides the secret message in the paragraph (cover file) thus generating a stego key; without degrading the originality of the paragraph .

The model works by hiding a message using start and end letter of the words of a cover file. It works on the binary value of a character. After converting the cipher text to a stream of bits,each bit is hidden by picking a word from the cover file and using either the start or the end letter of that word depending on the bit to be concealed.Bit 0 or 1 is hidden by reading a word, sequentially, from the cover file and including the starting letter or the end letter, respectively, of the word in the stego key. A word having same start and end letter is skipped. Since no change is made to the cover, the cover file and its corresponding stego file are exactly the same [4].

A. THE HIDING ALGORITHM

1. Get a cover file.
2. Convert the input file to its binary equivalent (bin).
3. Read a bit (x) from the bin.
4. Read a word from the cover file and write it in stego file.
5. If start and end letter of the word is same, then read the next word of the cover file andwrite it in the stego file.
6. s = start letter of the word and e = end letter of the word.
7. If x = 0, write s in the stego key.
8. Else if x = 1, write e in the stego key.
9. Repeat steps 3 to 8 till the end of the bin file.

10. Send the stego file and the stego key to the receiver.

B. THE SEEKING ALGORITHM

1. Read a character (c) from the stego key.
2. Read a word from the stego file.
3. If start and end letter of the word is same, then skip that word and read the next wordfrom the stego file.
4. Get the start letter (s) and end letter (e) of the word.
5. If c = s, then bit b = 0.
6. Else if c = e, then bit b = 1.
7. Write b in a file.
8. Execute above steps repeatedly till the end of the stego key.
9. Convert the file into its character equivalent.

The main limitation of this approach is that this method cannot hide bits of a message in words which have the same starting and ending letter. This can greatly reduce the capacity of the cover file to contain a message, if the cover file contains many words which have the same starting and ending letters. Also, it can be very easy for third parties to find patterns from the stego key as the stego key is only made up of the starting and ending letters of the cover text.

III. XORSTEG: THE PROPOSED METHOD

Our new proposed method of text steganography "XORSTEG" works on the same principle of hiding messages in an innocent looking text paragraph. It works on the binary value of the characters and the value of the XOR between the first and the last words of a line in the paragraph. The main demerit of the existing method of key based steganography [4] is that the secret message has to be hidden in a paragraph which should not contain words which have the same starting and ending letter. In our proposed method the cover file that is used for hiding the message can contain any number of words with the same starting and ending letters.

A. THE HIDING ALGORITHM (SENDER SIDE)

Here, any paragraph is taken as the cover file where the message to be sent to the receiver is embedded. First the secret message is converted into its equivalent binary form. The next step consists of reading a line of the cover file and then writing it into the stego file.

After the creation of the stego file a line is taken from the stego file and the first and last word of that line is taken. From this first word and the last word, the first and last letters of the respective words are taken and their binary equivalent is found out. XOR is taken between these binary values and a comparison is then made. When the XOR result of both the first and last word is same, complement of the value of the last word is taken and the binary values are converted into equivalent decimal values.

The key file is generated by taking a bit from the converted binary secret message. If the bit is 1, the greater decimal value is written in the key file else the smallest decimal value is written. The steps are:

1. Take the secret message and convert it into equivalent binary.
2. Take a normal text file as the cover file.
3. Read a line from the cover file and copy it to the stego file.
4. Take the first and last word from the line.
5. Take the first and last letter from the first word.
6. Convert the letters into their binary equivalent.
7. Take XOR between the values.
8. Perform steps (5)-(7) for the last word of that line.
9. Compare both XOR results (If both are same then take complement oflast XOR result; else: Change binary to equivalent decimal.)
10. Take a bit from the converted binary message.
11. If bit is 1 then write greater decimal in the key file else write the smallerdecimal.
12. Take the next words from both side of that line.
13. Perform steps (5)-(11) until the reading of the words in that line is finished.
14. Perform steps (5)-(13) for the next lines of the stego file.

After using this algorithm on the cover file, a stego file and a key file are generated which are sent to the receiver side.

B. EXTRACTING ALGORITHM (RECEIVER SIDE)

The extracting algorithm does the reverse process of the hiding algorithm and with the help of the stego file and the key file which is send by the sender, extracts the hidden message from the stego file.

1. Read the first decimal as a word and store it in k.
2. Read a line from stego file.
3. Take first word as F1 and last as L2 from that line.
4. Take the last and first letter from first word as F1l and F2s and take last and first letter from last word as L1l and L1s.
5. Take XOR(F1s and F1l) and XOR(L1s and L1l).
6. If results of XORs are same, take complement of last XOR result.
7. Change XOR result to decimal.
8. If k= max(R1,R2) then add 1 to binary else add 0 to binary.
9. Perform steps (3)-(8) until the same word or ending of the word isreached.
10. Read the next line of the stego file.
11. Perform steps (3)-(10) until the words of the key file are ended.
12. Convert the binary to string which will be the required message.

IV. RESULTS AND DISCUSSIONS

Here, we compare and analyze the two methods, discussed in the previous sections. To find the efficiency of the two methods we calculate the capacity ratio. Capacity can be defined as the total bytes of a secret message which can be stored in a cover file. The capacity ratio is computed by dividing the amount of hidden bytes over the size of the cover text in bytes [6].

Capacity ratio = (Amount of hidden bytes) / (Size of the cover text in bytes)

In the existing method and our proposed method, every bit of the message is hidden by taking a word of the cover file. So, if a word contains its starting and ending letter same, the existing method skips that word. In the existing method if the number of words in the cover file is less than the number of bytes in the message then the message cannot be embedded in the cover file.Some examples of cover files that hide the message have been given below:

The samples of embedded data are:

1. America (7 bytes)
2. Every man has two sides to him (29 bytes)
3. With spikes growing out of their skulls (39 bytes)
4. The name milky-way is derived from Greek mythology (50 bytes)
5. Black was the colour of sorrow and pain, black was the colour outside the train. (80 bytes)

The message that has to be hidden is taken to be the same as the cover file. Table 1 shows the capacities of the two methods:

Sample Number	1	2	3	4	5
Existing Method	0	0	0	0	1
Proposed method	1	1	1	1	1

Table 1: Capacity of the two methods

In the samples 1, 2, 3 and 4 the existing method's capacity ratio turns out to be 0, 1.21, 1.95, 1.16 but is written as 0 because it cannot hide the message in the cover file as the total number of bytes in the message is greater than the number of words in the cover file, as some words are neglected by the algorithm because they contain the same starting and ending letters. It is also seen that single letter words cannot be used in the existing method unlike our proposed method.

The results of hiding an example message of "This is a new text steganography method" after implementing our proposed method in Python are given in Figure 1, 2 and 3.

Mechanical analog computers started appearing in the first century and were later used in the medieval era for astronomical calculations. In World War II, mechanical analog computers were used for specialized military applications such as calculating torpedo aiming. During this time the first electronic digital computers were developed. Originally they were the size of a large room, consuming as much power as several hundred modern personal computers (PCs).

Figure 1: The cover file

Mechanical analog computers started appearing in the first century and were later used in the medieval era for astronomical calculations. In World War II, mechanical analog computers were used for specialized military applications such as calculating torpedo aiming. During this time the first electronic digital computers were developed. Originally they were the size of a large room, consuming as much power as several hundred modern personal computers (PCs).

Figure 2: The generated stego-file

0 17 15 10 11 21 1 0 13 17 79
11 26 13 0 7 9 2 54 0 28 92 0
23 16 9 21 24 5 23 73 15 0 15
23 1 4 5 22 11 9 67 49 8 17 16
9 29 5 15 67 5 0 5 33 16 16 3
12 7 17 2 5 5 4 94 17 0 9 1 4 20
2 54 74 18 16 8 1 6 16 18 17 20
79 27 18 18

Figure 3: The generated key-file

From the above figures we see that the stego file and cover file does not differ, unlike other format based methods. This stego file and key file which is generated is then sent to the intended recipient.

V. CONCLUSIONS

This type of key based steganography is more efficient than format based methods which alter the visual properties of a text to hide the message. But similar to cryptography, if the key and the stego file gets into the hands of any third party, he/she can easily extract the message.

Moreover, we see that the cover file which is used to hide the message has to be larger than the message which is to be hidden. This can be thought of as another limitation of these key based steganography methods.

The future work of this study can be done mainly on increasing its efficiency by being able to hide a message into any length of a cover file. Support for other languages other than English can also be added. To further increase the

security, we can use both cryptography and steganography methods to hide and encipher the message and the key and then transmit the key to the receiver.

References

[1]F. A. P. Petitcolas, R.J. Anderson, and M. G. Kuhn, *"Information hiding- a survey,"* InProceedings of IEEE, vol.87, pp. 1062-1078, 1999.

[2] L. Y. Por, and B. Delina, "*Information hiding- a new approach in text steganography*", 7th WSEAS Int. Conf. on Applied Computer and AppliedComputational Science, 2008, pp. 689-695.

[3] L. Y. Por, T. F. Ang, and B. Delina, "*WhiteSteg- a new scheme in information hiding using textsteganography,"WSEAS Transactions on Computers,* vol.7, no.6, pp. 735-745, 2008.

[4] Monika Agarwal*,"Textsteganographic approaches: a comparison"* International Journal of Network Security Its Applications (IJNSA), Vol.5, No.1, January 2013.

[5] K. Benett, *"Linguistic steganography- survey, analysis and robustness concerns for hiding information in text"* Purdue University, CERIAS Tech. Report 2004-13, 2004.

[6] F. A. Haidari, A. Gutub, K. A. Kahsah, and J. Hamodi, *"Improving security and capacity forArabic text steganography using "kashida"extensions"* 2009 IEEE/ACS Int.Conf. on ComputerSystems and Applications, 2009, pp. 396-399.