Exploiting Omega Networks to Hide Text-in-Text Messages

Ala Hamarsheh
Department of Computer Information Technology, The Arab American University – AAUJ,
Jenin, Palestine

Summary
This paper proposes a new steganography mechanism which can be used to hide text-in-text messages using the Omega Network (Multistage Network). Unlike other steganography methods that are introduced to hide secret messages (i.e. Text-in-image, image-in-image, text-in-audio, etc.), this mechanism does not require to seek for a suitable cover message in order to hide the secret messages. Alternatively, the mechanism must be supported with any preferred dictionary which will be used in creating cover messages.

Keywords: *Steganography, watermarking, encryption, secret message, cover message, secret key.*

1. Introduction

Due to the growth of the number of users over the Internet and large number of network based devices, securing the data over the Internet (storage and transmission) becomes an important topic. Steganography is one of the main methods can be used to protect data (e.g. text, image, video, audio, etc.) from unauthorized access [1,2].
Steganography is an effective technique to secure data transmission over the Internet. Steganography comes with new security measures over other traditional security methods. For example, it does not allow attackers to detect the existence of secrete data that is hidden in a cover message. Cover messages can be of any type, for example text, video, images, and audio files. This paper introduces a new steganography technique to hide text in text messages.
Generally, for data hiding, there are many existing solutions rely on hiding message bits in Descrete Cosine Transformation (DCT) coefficients [3[[4], motion vectors(MVs)[5][6][7], quantization scale[8] or prediction modes. This technique relies on Omega Networks to hide text-in-text messages. Unfortunately, these methods have problems and not efficient enough to hide the secret messages as they need a suitable cover for the embedding process, easy detected by the attackers and take long execution runtime.
This research focuses on one method for hiding secret information, which is the text–in–text steganography. The currently proposed techniques that are support this depend on the line shifting, word shifting [9][10], manipulation position of lines and words [11], HTML files can be used to carry information since adding spaces, tab, invisible characters, and extra line break are ignored by web browsers [12]. However, these methods still have many problems as we mentioned above.
This paper proposes a new method called modified Omega Network for hiding and extracting the text message in/from another text message. The proposed technique used the methodology that was proposed in the original omega network [13, 14] in order to hide text-in-text messages.
The paper is organized as follows: section one describes the structure of the Modified Omega Network. Section two analyzes the embedding and extracting methods. Section three describes the effects of attackers on this mechanism. Section four concludes the paper and states the future work.

2. Overview of Omega Networks

Fig. 1 shows the structure of the multistage omega network which consists of *N* input and *M* output.

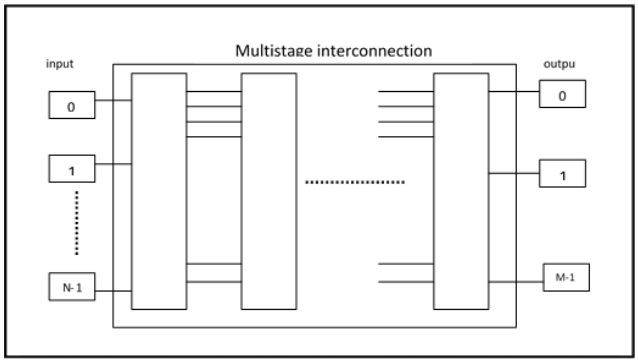

Fig. 1 the schematic of a typical multistage interconnection network

Omega network consists of log N stages (where N is the number of input and output connections). Each stage contains an interconnection pattern which is used to connect N input connections with M output connections. Eq. 1 explains how to connect between input *i* and output *j*.

Manuscript received May 5, 2015
Manuscript revised May 20, 2015

$$j = \begin{cases} 2i, & 0 \le i \le \frac{N}{2} - 1 \\ 2i + 1 - N, & \frac{N}{2} \le i \le N - 1 \end{cases} \quad (1)$$

The equation eq.1 illustrates a left-rotation operation on the binary representation of input i to obtain output j, this interconnection pattern is called a perfect shuffle. Fig. 2 shows an example of perfect shuffle interconnection pattern to connect eight inputs with eight outputs.

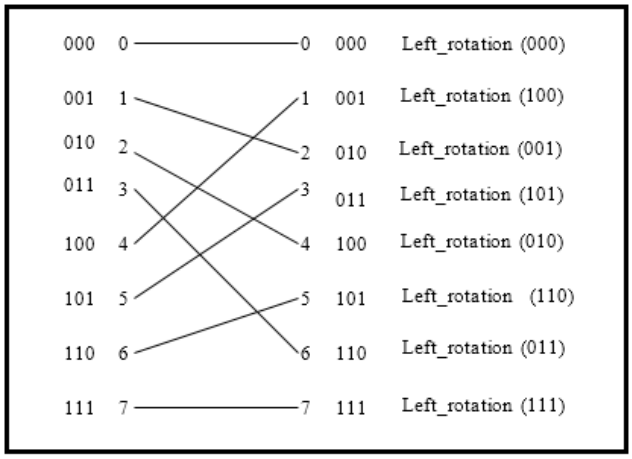

Fig. 2: a perfect shuffle interconnection for eight inputs and eight outputs.

As indicated previously, omega networks consist of multiple stages. The number of switching elements in each stage is N/2 of switching elements. Each switch has two connection modes, pass through and cross over. In the first mode, the inputs are sent straight through to the outputs, figure 3(a) explains this mode.

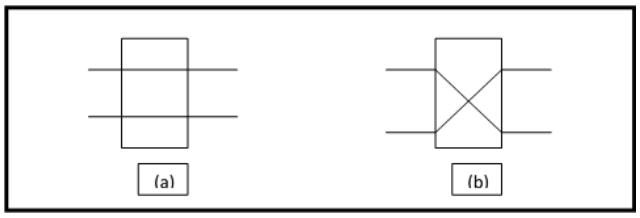

Fig. 3. Two switching configuration of the 2*2 switch
(a) pass-through; (b) Cross-over

An omega network has N/2 × log N switching elements, and the cost of such a network grows as Θ (N log N).

Fig. 4 shows an omega network for eight input. Input nodes of the network and the output nodes.

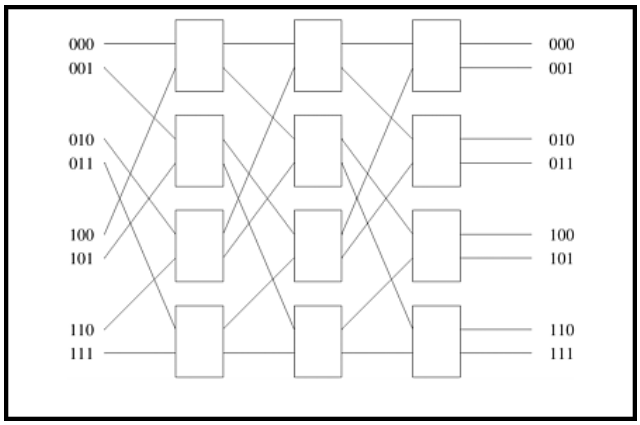

Fig. 4. an omega network connecting eight inputs and outputs

Routing messages in an omega network is accomplished by using a simple scheme. Let “s” and “d” be the binary representations of the source and destination of the message.

The routing concept in the omega networks goes as follows. The message traverses the link until reach the first switching element. The switching element checks the most significant bits of “s” and “d” in the message. If these bits are the same, the switching element forwards the message using pass- through mode. Otherwise, the switching element forwards the massage using crossover mode. This routing concept is repeated at every switching stage using the “next” most significant bit. Traversing log N stages used all log N bits in the binary representation of “s” and “d”. Figure 4 shows message routing over an eight-input omega network from input two (010) to seven (111) [cross, pass, cross]] and from input six (110) to four (100) [pass, cross, pass] [15].

2. Using Omega Networks to Hide Text-in-Text Messages

Omega networks was originally introduced to interconnect between multi-processors via shared-memory modules. The proposed technique uses only the structure of omega network to generate new words based on characters of the original message. Similarly, the technique retrieves the original message using the stego-message (contains the secret message) that was generated at the sender’s side.

The following sections describe the embedding and extracting processes. The former is used the sender’s site to hide the secret message (a message which is not allowable to be accessed by unauthorized users through the communication process [16]), and the later one is used at the destination side to extract the hidden (secret message).

2.1 Embedding Process

The embedding process is used to hide the secret messages by generating new stego-messages. English language has 26 alphabetic characters. In order to address all of these characters, the structure of omega network must have five stages to represent 32 input/output characters. The stage #3 is used as starting stage to hide secret messages.

The embedding process will be as follows:

1. Start from stage #3, locate the position of the character.
2. Move backward by performing shift right rotate (SRR) to reach the stage #2. This makes stage #2 contains two characters. After that, one of the characters will be picked up randomly, and then locate the position of the selected character. In order to reach stage #1, the SRR is performed again and this result in generating two characters in which one of them will be picked up randomly.
3. Start from Stage#3, move forward to reach stage #4. Hence, Shift Left Rotate (SLR) is performed, and as a result, two characters are generated. One of them is selected randomly and the other will be discarded. The process is repeated until reach the stage #5. Finally, we got 16 probability of covering each character (4 inputs X 4 outputs). Two characters (i.e. one character at the input line and one character at the output line) will be selected to embed each secret character of the original message. Fig. 5 illustrates the embedding process.
4. Concatenate the two resulting characters like A concat. Z = AZ.
5. Look up in the dictionary file to find a word starts with AZ for example will find the word “azure”. If the look up process is failed to find a suitable word in the dictionary, the algorithm generates a random word that starts with A and ends with Z characters.

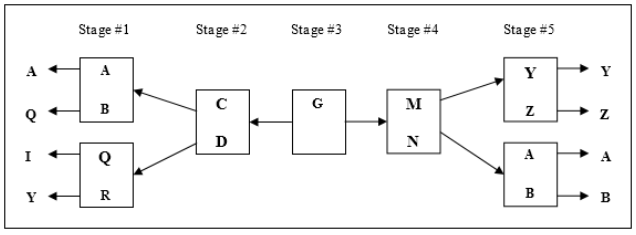

Fig. 5: part of omega network with five stages to embed the character G.

6. Repeat the previous steps until all the characters in the secret message covered properly with generated words.
7. Create the stego-message based on the generated words and send it to the recipients.

Fig. 6 shows a part of the application which used to embed the character ‘G’

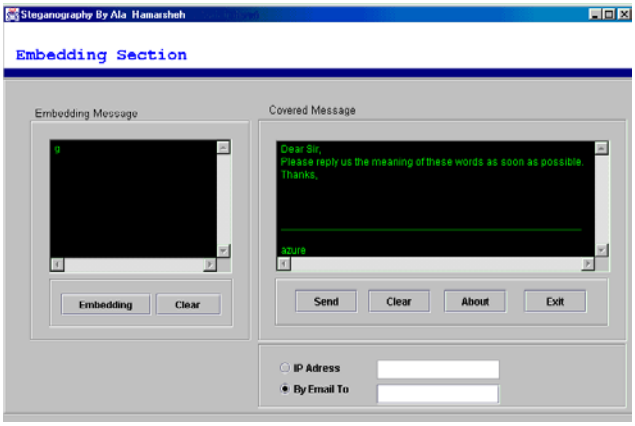

Fig. 6: embedding the character ‘G’

2.2 Extracting Process

The extracting process is used to retrieve the secret message from a stego-message that was generated at the sender’s site. Similar to embedding process, the structure of modified omega network will be used to extract the secret message.

The extracting process will be as follows:

1. Divide the secret message into separate words.
2. Select the first two characters of each word.
3. Use the structure of omega network and put the binary code of the first character in the input port and the binary code of the second character in the output port.
4. Apply the XOR function between the binary digits of the first and second characters to pass through the structure of omega network until reach stage #3 (because the secret character is embedded in this stage).
5. Route through omega network based on the result of XOR function (zero means pass through, and one means cross over). The following example explains the routing through omega network based on the results of XOR function.

$$\begin{array}{l} A_1, A_2, A_3, A_3, A_5 \\ \qquad\qquad\qquad\qquad\qquad\quad \text{XOR} \\ B_1, B_2, B_3, B_4, B_5, \\ \text{----------} \\ C \;\; P \;\; P \;\; C \;\; P \end{array}$$

In order to extract the character ‘G’ from the word “azure”, the first two characters are ‘A’ and ‘Z.’ After applying the XOR between ‘A’ and ‘Z’ we got:

A = 00000
XOR
Z = 11001

11001 ➔ C C P P C (i.e. C: cross, P: pass).

Fig. 7 shows the extracting process.

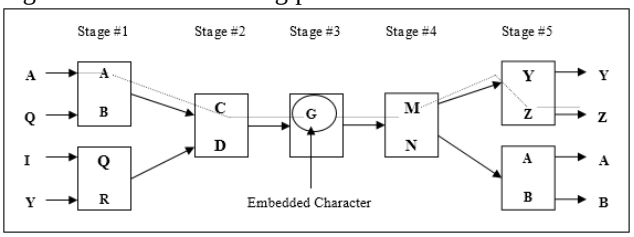

Fig. 7: part of omega network with five stages to extract the character G.

Fig. 8 shows a part of the application which used to extract the character ‘G’

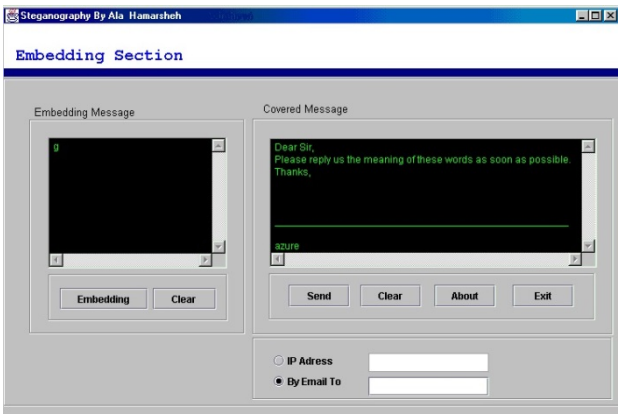

Fig. 8: extracting the character ‘G’

3. Attackers

The cryptography secures the communication (by hiding the exchanged information) between multiple parties when these parties are communicating via unsecured environment. Similarly, Steganography protocols can be used in situations where the exchanged information between parties needs to be hidden. The key point of hiding information in steganography protocols is to decrease the awareness of detecting the secret information from unauthorized users (attackers) [16].

In order to increase the security in this mechanism, the order of the binary code of the secret characters could be changed dynamically in each stage. This makes the embedding process follows different routes in the structure of the omega network every time the sender needs to embed the same character. This will increase the difficulty for the attackers to retrieve the secret character.

4. Conclusions and Future Work

The paper proposed a new mechanism to hide text-in-text messages. The current steganography mechanisms require selection of suitable stego-cover to hide secret messages. Nevertheless, this mechanism has the ability to generate stego-covers based on secret messages. The mechanism offers different probabilities for embedding process which gives better security and increases the difficulty of detecting secret messages by attackers.

The main drawback of the mechanism is finding a suitable word in the dictionary file. As indicated previously, in some situations, particularly when failed to locate a suitable cover word in the dictionary file, the mechanism generates a meaningless word (suitable cover) to hide the secret letter. Additionally, the frequent use of the dictionary file to embed each character which deteriorates the overall system performance. However, the future work might go towards finding solutions to the above mentioned drawbacks. Additionally, structuring and sending meaningful sentences using the generated words instead of sending separate words will be an interesting topic for the future work.

References

[1] Bender, W., Gruhl, D., Morimoto, N., & Lu, A., "Techniques for data hiding", IBM System Journal, vo1.35 Nos.3&4, pp. 313-336,1996.

[2] Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt," Digital image steganography: Survey and analysis of current methods", Signal Processing, vol. 90, pp.727-752, 2010.

[3] S. Kapotas and A. Skodras, “A new data hiding scheme for scene change detection in H.264 encoded video sequences,” in Proc. IEEEInt. Conf.Multimedia Expo ICME, pp. 277–280, Jun. 2008.

[4] M. Carli, M. Farais, E. D. Gelasca, R. Tedesco, and A. Neri, “Quality assessment using data hiding on perceptually important areas,” in Proc.IEEE Int. Conf. Image Processing, ICIP, pp. III-1200-3–III- 1200-3, Sep. 2005.

[5] Y. Li, H.-X. Chen, and Y. Zhao, “A new method of data hiding based on H.264 encoded video sequences,” in Proc. IEEE Int. Conf. Signal Processing, ICSP, pp. 1833–1836, Oct. 2010.

[6] K. Nakajima, K. Tanaka, T. Matsuoka, and Y. Nakajima, “Rewritable data embedding on MPEG coded data domain,” in Proc. IEEE Int.Conf. Multimedia and Expo, ICME, pp. 682–685, Jul. 2005.

[7] D.-Y. Fang and L.-W.Chang, “Data hiding for digital video with phase of motion vector,” in Proc. IEEE Int. Symp. Circuits Systems, ISCAS, Sep. 2006.

[8] C. Xu, X. Ping, and T. Zhang, "Steganography in compressed video stream," in Proc. Int. Conf. Innovative Computing, Information and Control, ICICIC'06, vol. II, pp. 803–806, 2006.
[9] Lingjun Li, Liusheng Huang, Xinxin Zhao, Wei Yang, Zhili Chen," A Statistical Attack on a Kind of Word-Shift Text-Steganography", International Conference on Intelligent Information Hiding and Multimedia Signal Processing, International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2008.
[10] Altigani, A. ; Barry, B., "A hybrid approach to secure transmitted messages using advanced encryption standard (AES) and Word Shift Coding Protocol" , International Conference on Computing, Electrical and Electronics Engineering (ICCEEE), 2013.
[11] Neil F. Johnson and Sushil Jajodia. Steganography: Seeing the Unseen IEEE Computer, February 1998.
[12] Gyankamal J. Chhajed, Krupali V. Deshmukh, Trupti S. Kulkarni, "Review on Binary Image Steganography and Watermarking", International Journal on Computer Science and Engineering (IJCSE), 2011.
[13] Padmanabhan, K.; and Lawrie, D.H., "A class of redundant path multistage interconnection networks", IEEE Trans. Computers, 1983.
[14] Sharon Rose Govada , Bonu Satish Kumar , Manjula Devarakonda and Meka James Stephen, "Text Steganography with Multi level Shielding", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 3, July 2012.
[15] Bender W., D. Gruhl, N.Morimotot, A. Lu, "Techniques For Data Hiding ", IBM Systems Journal ,1996
[16] Stefan K., Fabien A.P. Petitcolas, " Information Hiding Techniques For Steganography And Digital Watermarking " , Artech Huse ,Boston London, 2000.

Ala Hamarsheh is an assistant professor at the Department of Computer Information Technology of the Arab American University of Jenin. He received his PhD in engineering sciences from Vrije Universiteit Brussel (VUB)/Brussels-Belgium. Prior this, he was working as a full-time lecturer at the Arab American University, Jenin, Palestine. He obtained a BSc degree in computer science at the Faculty of Science, Birzeit University, Palestine, in 2000. He obtained an MSc degree in computer science at the Kind Abdullah II School for IT, The University of Jordan, Jordan, in 2003. He has published numerous papers in international refereed journals and conferences.