Proceedings of the 2011 International Conference on Machine Learning and Cybernetics, Guilin, 10-13 July, 2011

HIDING DATA IN TETRIS

ZHAN-HE OU, LING-HWEI CHEN

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
E-MAIL: id4922@ debut.cis.nctu.edu.tw, lhchen@cc.nctu.edu.tw

Abstract:

Many kinds of data hiding methods using different cover media have been proposed, several papers use games to do steganography recently. Tetris is a famous puzzle game, and this game randomly generates the tetrominoes pieces to the player. In this paper, we proposed a steganography system for the Tetris game through the generated tetromino sequences. A three phases embedding algorithm is provided to make the stegoed tetromino sequence look like those generated by a normal way. The experimental results show that this system is undetectable.

Keywords:

Stegonography; Tetromino; Tetris; Undetectable

1. Introduction

Steganography is the art of secret communication which allows user concealing their information within innocuous cover media. The main requirement of stegonography is undetectability which means that it is impossible to determine whether a work embeds secret message or not. According to different purposes, some steganography systems also require a stego key to control the embedding process. For example, set the stego key as a random seed to randomize the secret message, or generate some pseudo random numbers as the function parameters.

Many kinds of data hiding methods using different cover media have been proposed. Image is the most popular cover media, and variant hiding methods allow users not only hiding messages in spatial [1] or frequency [2] domain, but also recovering the original cover image after extraction [3, 4]. On the other hand, steganalysis is widely researched [5-7] to test the security of these methods.

Recently, several papers have provided steganography algorithms through variant games [8-11]. In 2006, Hernandez-Castro et al. [8] proposed a steganographic method in extensive-form games by hiding secret message on the game moves. If there is a searching algorithm can provide the best n moves, then a player can choose one of these n moves to represent a $log_2 n$ bit secret message. However, different extensive-form games need different searching algorithms, and the algorithm development is not a easy task. Furthermore, the stegoed moves might be abnormal and make the method detectable.

Kieu et al. [9] proposed a Sudoku-based image hiding scheme in spatial domain. The hiding system will expand the solved Sudoku matrix into a 261x261 look-up table, and use a pixel pair with values (v_i, v_j) to hide a secret message s by searching the nearest element (v_i', v_j') of the position (v_i, v_j) in the look-up table with value s. Then the values of the pixel pair are replaced by (v_i', v_j'). Because of the property of Sudoku matrix, the changed values in the pixel pair will be less than (±2, ±2). However, both sender and receiver should share the same Sudoku grid information, and the method is weak for compression attack as most of hiding schemes in the special domain.

In 2009, Farn and Chen [10-11] proposed two steganography methods in two kinds of puzzle games. One method hides secrete message in the attached semi-cycles on the puzzle pieces. The semi-cycle type and position are used to hide message [10]. The other hides secrete message through the piece permutation in a jig swap puzzle game [11]. Because these methods do not hide secrete message through pixel values, they are robust under the compression attack. However, the puzzle piece needs a recognizable size for human playing, so larger secrete message needs a bigger image size to cut more pieces. This means that the capacities of these methods are limited by the puzzle image size.

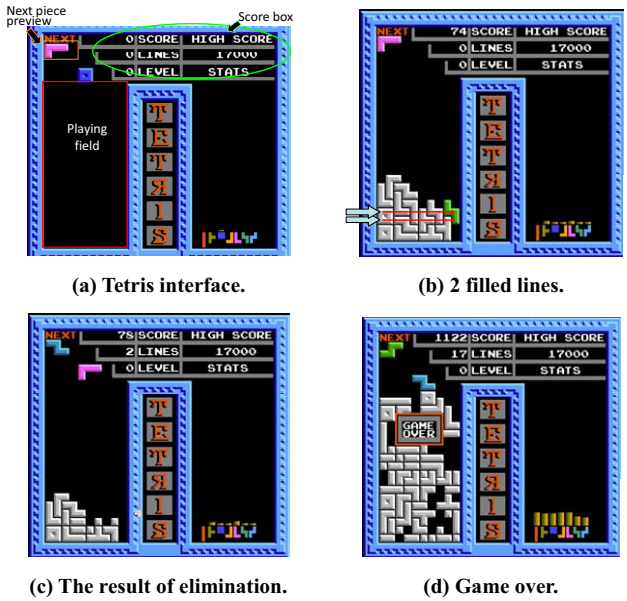
This paper proposed a new steganography method using Tetris to solve the capacity limitation problem. Tetris is a famous video puzzle game from 1984. The game interface is formed by three parts, playing field, score box and next piece preview, as shown in Fig. 1(a). The playing field is usually formed by 10x20 grid-squares, and the pieces for the Tetris game are also called tetrominoes. At the beginning of each game, one tetromino called playing piece will appear above the top middle of the playing field, and the other tetromino called next piece will appear on the next piece preview. After the initialization, the playing piece will keep falling to the bottom. When the first playing piece is blocked by the ground of the playing field, it will

978-1-4577-0308-9/11/$26.00 ©2011 IEEE

be fixed. The fixed tetromino will fill the grid-squares of the playing field, and the filled grid-squares will be regarded as the extending ground, and be able to block the other tetrominoes. After the first playing piece is fixed, the next piece will then become the second playing piece. It appears above the top middle of the playing field and keeps falling down. Another tetromino will immediately appear on the next piece preview wait the playing piece fixed. These processes will repeat until the game over. Note that before the playing piece is fixed, the player can use shifting and rotation operations to control the playing piece to reach a proper place.

Once a horizontal line is filled by some fixed playing pieces, the horizontal line will be eliminated, as shown in Fig. 1(b). Those parts above the eliminated lines will fall down the same number of distances, as shown in Fig. 1(c). When the playing piece is fixed, the game system will give a score shown in the score box. More horizontal lines are eliminated, higher score will be given.

When the player stacks a tetromino over the top of the playing field, the game is over, as shown in Fig. 1(d). After the game is over, the game system usually pops a "Play Again" bottom for the player to try again. The purpose of the Tetris game is to get scores as high as possible.



(a) Tetris interface. (b) 2 filled lines.

(c) The result of elimination. (d) Game over.

Figure 1. An example to illustrate Tetris.

The tetromino is a pattern formed by 4 orthogonally contact squares, and it has seven different shapes. According to the looking of these shapes, they are called I, J, L, O, S, T and Z, as shown in Fig. 2. These tetrominoes will be randomly generated during the game play, and the ordered collection of these tetrominoes is called a tetromino sequence.

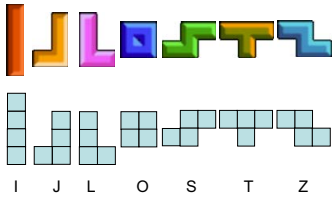


Figure 2. Seven kinds of tetromino shapes.

Most of Tetris games use dice-like generator to generate the tetromino sequence. This generator generates each tetromino like rolling a 7-side dice.

The proposed method will hide secret message through the generated tetromino sequence and make the sequence looks like those sequences from a normal Tetris game. Experiments show that the stegoed tetromino sequence generated by the proposed method is undetectable.

Scenery for the proposed method is also provided. A sender provides the Tetris game with the proposed hiding method in the internet, and anyone can access it through the internet. The receiver can get the secret information during the game play, and any other player or warden is not aware of the secret communication.

At the rest of this paper, Section 2 will describe the detail of the proposed system. Section 3 shows the experimental result. And a conclusion is made at the final section.

2. Proposed system

When a player plays a Tetris game, the game system will keep generating tetrominoes for the player. The proposed system consists of three parts: embedding process, extracting process and scenario. The embedding process will embed the secret message through the sequentially generated tetrominoes. When the secret receiver plays the Tetris game with secrete embedded, through the extracting process, he/she can use the pre-shared stego key to extract the secret message from the played tetromino sequence. Note that the stegoed tetromino sequence is undetectable. In the following, we will introduce the details of these three parts.

2.1. Embedding process

A dice-like generator generates tetrominoes like rolling a 7-side dice. As mentioned previously, there are 7 different shapes, thus, there are 7^n different sequences with n tetrominoes. According to the alphabetical order, the

tetrominoes {I, J, L, O, S, T, Z} can be regarded as {0, 1, 2, 3, 4, 5, 6}, respectively, and each tetromino sequence can represent a 7-based number with n digits. For example, a tetromino sequence (O, L, T, Z, L, L, O, L) will correspond to the 7-based number 32562232_7.

To embed a secret message, S_o, in a tetromino sequence, S_o will first be converted into a 7-based number sequence, and the corresponding stegoed tetromino sequence will be pop out to the player. During the game playing, the player will see these tetrominoes appearing sequentially, and the corresponding number of the played tetromino sequence can be recovered into the secret message S_o.

To make the stegoed tetromino sequence look like these generated by a general Tetris game system, embedding message in a tetromino sequence should satisfy three requirements. First, stegoed tetromino sequences should be different in different games, even the secret message is the same. Second, the length of a stegoed tetromino sequence should also be embedded. Third, after the stegoed tetromino sequence is popped out, the game cannot be stopped before the player leaves the game.

This paper proposed a three-phase embedding process satisfying the above-mentioned three requirements. The block diagram is shown in Fig. 3.

The first phase called random seed generator will produce a random seed to control the secret message randomization in the next phase. The second phase called stegoed sequence generator will use the random seed to generate the stegoed tetromino sequence and pop it out after the generation. When the stegoed tetrominoes are all popped to the player, the last phase called game preserver will keep generating tetrominoes randomly until the player leaves the system.

2.1.1. Random seed generator

In this phase, the system will generate a random seed for the secret message randomization. Note that the random seed is different for each game, this makes the randomization result different.

First, a random number generator is used to generate a random sequence $U_D = (u_1, u_2, \ldots, u_l)$ with $u_i \in \{0, 1, 2, 3, 4, 5, 6\}$. The corresponding tetrominoes are then popped out sequentially to the player. If the game is over before all l tetrominoes popped, the remaining tetrominoes will continually pop out after the player chooses "Play Again". When all corresponding tetrominoes in U_D are pop out, the random seed generator will base on U_D and the stego key to generate the random seed, as shown in Fig. 4. The random sequence U_D represents a 7-based number with l digits, this number will be converted into a decimal number and added with a pre-shared stego key K to result in the random seed. Note that the length l of U_D is a pre-shared information.

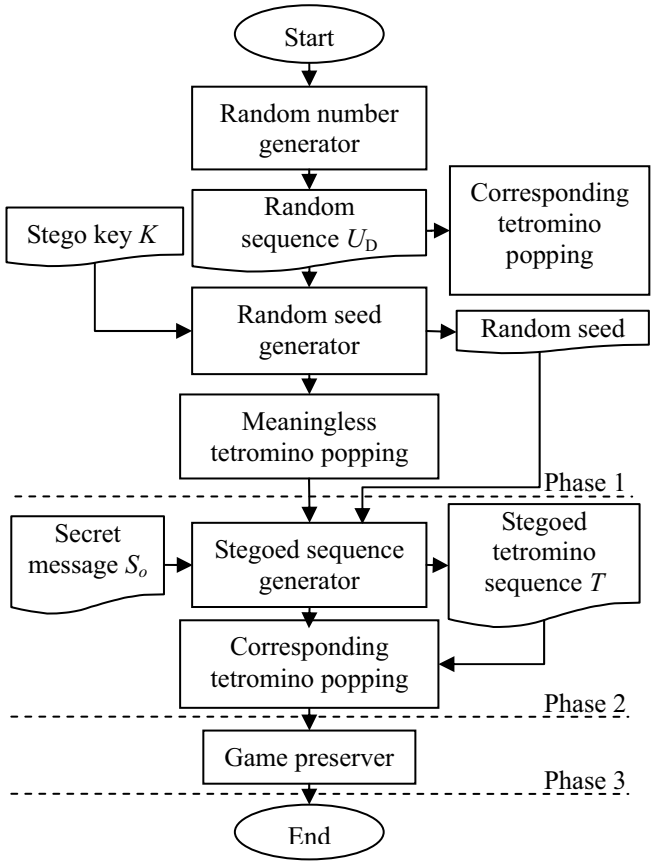


Figure 3. Block diagram of the proposed embedding process.

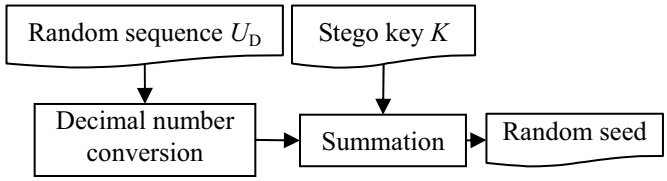


Figure 4. Block diagram of the random seed generator.

After the random seed is generated, the meaningless tetromino popping will keep generating meaningless tetrominoes to player until the game is over. Then the system conducts the second phase.

2.1.2. Stegoed sequence generator

In the second phase, the stegoed sequence generator will generate the stegoed tetromino sequence, and the

corresponding tetrominoes will be pop out to the player. The block diagram of the stegoed sequence generator is shown in Fig. 5. At first, the original secret message S_o will be converted into a 7-based number $S_7 = (s_1, s_2, s_3, \ldots, s_n)_7$, where s_i is the ith digit of this number. S_7 is considered as a sequence with length n. Then, the length n is attached to S_7 through the length attachment. Finally, a pseudo randomization is applied to get the stegoed tetromino sequence.

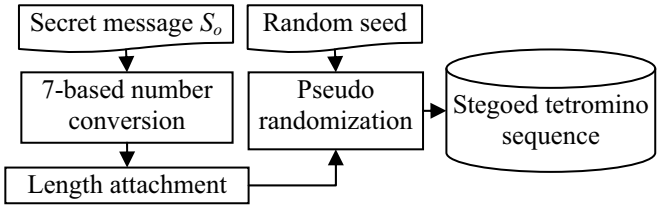


Figure 5. Block diagram of the stegoed sequence generator.

In order to inform the receiver, the length of S_7 is attached to S_7. Fig. 6 shows the structure of the length and S_7. This structure contains k segments, each of which contains a continuous indicator c_i, and a subsequence Sub_i. If $c_i = 0$, the segment is the last one. The last segment has an additional length indicator P between the continuous indicator c_k and subsequence Sub_k. This indicator P is a 7-based number with h digits, and it records the length of the last subsequence.

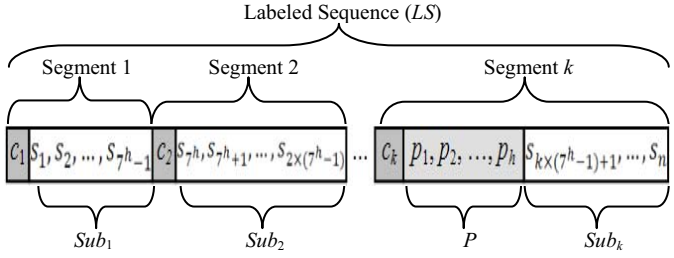


Figure 6. Structure of the length of S_7 and S_7.

Note that the length h is a pre-shared information, and the sequence S_7 will be divided into $k = \lceil n/(7^h\text{-}1) \rceil$ subsequences. Each of the first k-1 subsequences has 7^h-1 digits of S_7 and a continuous indicator $c_i = 1$. The last subsequence has $c_k = 0$, and the length of the last subsequence is not greater than 7^h-1.

After the length of S_7 is attached, the resulting sequence is called labeled sequence (LS), denoted as $LS = (a_1, a_2, a_3, \ldots, a_{n+k+h})$, and each a_i will be randomized through the following function

$$m_i = (a_i + r_i) \bmod 7, \quad (1)$$

where $r_i \in \{0, 1, 2, 3, 4, 5, 6\}$ is the ith random number generated by the random seed. The randomized number sequence, T, denoted as $T = (m_1, m_2, m_3, \ldots, m_{n+k+h})$ will be regarded as the stegoed tetromino sequence.

After the stegoed tetromino sequence is generated, the new game will then be started. The corresponding tetrominoes of T will sequentially be pop out. If the game is over after the ith tetromino is popped with $i < n+h+k$, the remaining tetrominoes will be pop out after the player presses the "Play Again" bottom. After all the tetrominoes are popped out, the system will get into the final phase to finish the embedding process.

2.1.3. Game preserver

To avoid the abnormal halt, after the whole stegoed tetromino sequence is popped out, the system will get into the game preserver phase and keep randomly generating meaningless tetrominoes until the player leaves the game. This will make the player not notice any abnormal phenomenon.

2.2. Extracting process

The proposed extracting process also has three phases. The block diagram is shown in Fig. 7. To extract the original secret message, the proposed extractor will first generate the same random seed as that produced in the embedding process. Then based on this random seed, the secret sequence, S_7, can be extracted. Finally, the extracted sequence is converted to the original secret message.

In the first phase, the receiver starts a game and gets the first l tetrominoes. Next, the l tetrominoes are concatenated into a 7-based number sequence $U_D = (u_1, u_2, \ldots, u_l)$. This sequence will be treated as a 7-based number with l digits and converted to a decimal number, the decimal number is then added with the pre-shared stego key K to get the random seed. The whole process is just the same as that in the embedding process. After getting the random seed, the receiver will keep playing until the game is over and then enters the second phase.

In the second phase, the falling tetrominoes will be collected and transformed into a sequence $T = (m_1, m_2, m_3, \ldots)$. Note that the length of this sequence is unknown at the beginning of this phase. The extractor will generate a random number sequence $R = (r_1, r_2, r_3, \ldots)$ using the random seed obtained in the first phase. Then we can get $LS = (a_1, a_2, a_3, \ldots)$ through the following equation

$$a_i = (r_i - m_i) \bmod 7, \quad (2)$$

Note that LS contains several segments, except the last one, each of which has one continuous indicator and a

subsequence. Based on the continuous indicator c_i of the ith segment (for example, a_1 is the first indicator c_1). The extractor can determine the meaning of the following elements. If $c_i = 1$, the following 7^h-1 elements will be a subsequence, and there is another segment next to these elements. Else if $c_i = 0$, the segment is the final one, and the following h elements stand for the length indicator P and are used to get the final subsequence.

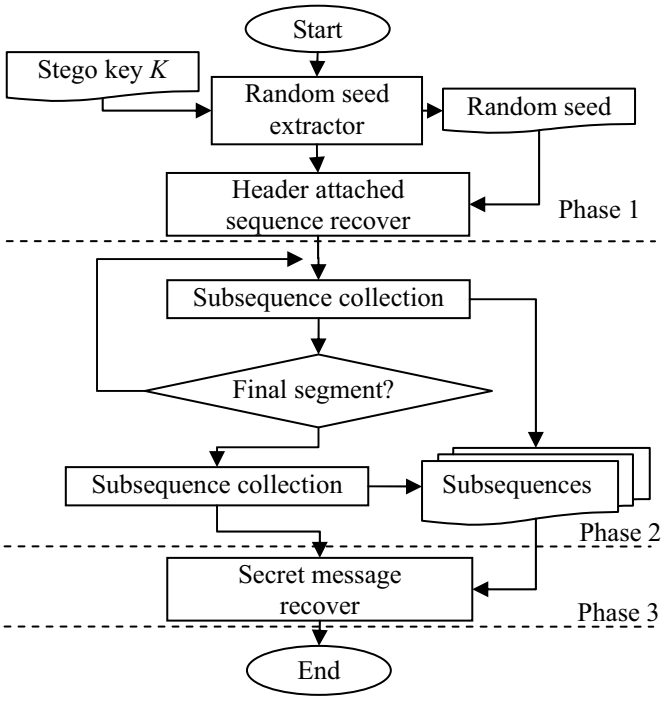


Figure 7. The block diagram of the extracting process.

After the last segment is collected, all obtained subsequences are collected to form the 7-based number S_7, then this number is converted into the original secret message S_o.

2.3. Proposed scenario

Here, we proposed a scenario to transmit the secret message through an online Tetris game. In the scenario, the following rules should be obeyed.

1. Both sender and receiver have the same stego key K and the shared information l and h.
2. Sender should create an online Tetris game website similar to [14, 15] which provides users a Tetris game to play online.
3. When sender wants to send the secret message, he/she will prepare the corresponding on the website and invite the receiver to play the Tetris.
4. The receiver will then access the website and play the Tetris game online.
5. When the proposed embedding process receives the start game request from the receiver, it will transform the secret message into the stegoed tetromino sequence and sequentially pop these tetrominoes to the receiver.
6. The receiver will play the Tetris game and use the stego key and the proposed extraction procedure to extract the secret message from the popped tetrominoes.

3. Experimental result and security analysis

In this section, we will give some experiments to prove the undetectability of the proposed method. First, we recorded several tetromino sequences from four different Tetris games provided on the internet. Each game has 10,000 tetrominoes recorded and used to compare with the proposed method. These games includes Tengen's version Tetris game on the Famicom video game console [12], the "N-blox" from the Tetris official website [13], and two free shared Tetris games provided in the internet [14,15]. We also provided some stegoed tetromino sequences with total 10,000 pieces.

Here, we will compare the entropy values between these games and the proposed method. Assume that a normal Tetris game uses a dice-like generator to generate tetrominoes, the distribution of the different shape appearing will be near uniform. To test the distribution, the entropy value is used and defined as

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i, \qquad (3)$$

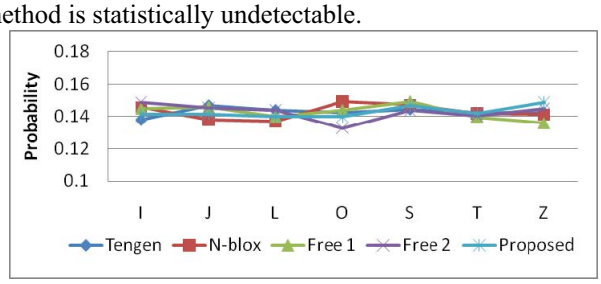
p_i is the probability of shape i. If the distribution is uniform, the entropy would be the maximum value $log_2 n$.

To prove the undetectability, we will show that the stegoed tetromino sequence generated by the proposed method have similar entropy value to those generated by the normal Tetris games.

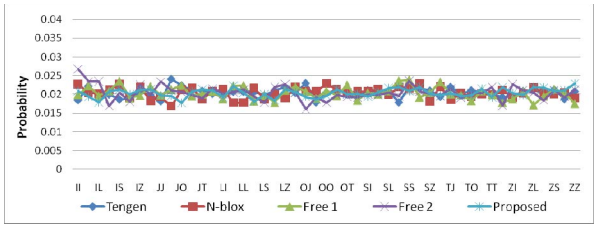
Three different methods are provided to record the appearing frequencies of tetrominoes, these methods count the appearing frequencies of a single tetromino, the two consecutive tetrominoes and the three consecutive tetrominoes, respectively. Fig. 8 shows the appearing probabilities using three methods. From this figure, we can see that different Tetris games and the proposed method have similar appearing probabilities.

On the other hand, Table 1 shows the entropy values of the appearing probabilities in different record methods. From this table, we can see that all games and the proposed method have similar entropy values near to the maximum $log_2 n$. This indicates that the distributions of these shapes are near uniform. That is, the Tetris game provided by the proposed method looks like the normal Tetris games.

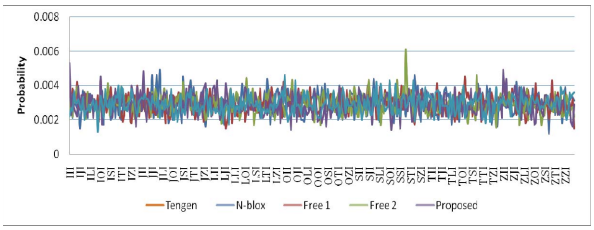
From Fig. 8 and Table 1, we can see that the proposed method is statistically undetectable.



(a) The appearance probabilities of a single tetromino.



(b) The appearance probabilities of two consecutive tetrominoes.



(c) The appearance probabilities of three consecutive tetrominoes.

Figure 8. The appearing probabilities of three different record methods.

Table 1. The entropies of the appearing probabilities in different record methods.

	Tengen	*N-blox*	*Free 1*	*Free 2*	*Proposed*
1-tetromino ($n = 7$)	2.81	2.81	2.81	2.81	2.81
2-tetromino ($n = 49$)	5.62	5.61	5.61	5.61	5.61
3-tetromino ($n = 343$)	8.39	8.39	8.40	8.39	8.40

4. Conclusion

As we know, the more kinds of carriers a steganographic system can use to embed data, the more secure the system is. This paper proposed a novel stegonography method on hiding data in the famous Tetris game. A secret message is hidden through the tetromino sequence. Based on the "Play Again" function, the proposed method can extend the tetromino sequence to hide a larger secret message. We have also shown that the proposed method is undetectable and secure.

Acknowledgements

This work is supported in part by National Science Council of Republic of China under grant NSC 98-2221-E-009-117-MY2.

References

[1] W. Luo, F. Huang, and J. Huang, "Edge Adaptive Image Steganography Based on LSB Matching Revisited", IEEE Trans. on Information Forensics and Security, Vol. 5 Issue 2, pp. 201-214, 2010.

[2] L. Zhang, H. Wang, and R. Wu, "A High-Capacity Steganography Scheme for JPEG2000 Baseline System", IEEE Trans. on Image Processing, Vol. 18, Issue 8, pp. 1797-1803, 2009.

[3] J. H. Lee, and M. Y. Wu, "Reversible Data-hiding Method for Palette-based Images", Optical Engineering, Vol. 47, Issue 4, pp. 047008, 2008.

[4] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible Data Hiding", IEEE Trans. on Circuits and Systems for Video Technology, Vol. 16, Issue 3, pp. 354-362, 2006.

[5] Y. S. Chen, and R. Z. Wang, "Steganalysis of Reversible Contrast Mapping Watermarking", IEEE Signal Processing Letters, Vol. 16, Issue 2, pp. 125-128, 2009

[6] G. Gul, and F. Kurugollu, "SVD-Based Universal Spatial Domain Image Steganalysis", IEEE Trans. on Information Forensics and Security, Vol. 5, Issue 2, pp. 349-353, 2010.

[7] J. Zhang, and D. Zhang, "Detection of LSB Matching Steganography in Decompressed Images", IEEE Signal Processing Letters, Vol. 17, Issue 2, pp. 141-144, 2010.

[8] J. C. Hernandez-Castro, I. Blasco-Lopez, J. M. Estevez-Tapiador, and A. Ribagorda-Garnacho, "Steganography in Games: A General Methodology

and Its Application to The Game of Go", Computers & Security, Vol. 25, Issue 1, pp. 64-71, 2006.
[9] T. D. Kieu, Z. H. Wang, C. C. Chang, and M. C. Li, "A Sudoku Based Wet Paper Hiding Scheme", International Journal of Smart Home, Vol. 3, No. 2, pp. 1-12, 2009.
[10] E. J. Farn, and C. C. Chen, "Jigsaw puzzle images for steganography", Optical Engineering, Vol. 48, Issue 7, pp. 077006, 2009.
[11] E. J. Farn, and C. C. Chen, "Novel Steganographic Method Based on Jig Swap Puzzle Images", Journal of Electronic Imaging, Vol. 18, Issue 1, pp. 013003, 2009.
[12] http://www.allgame.com/game.php?id=14911
[13] http://www.tetrisfriends.com/
[14] http://www.playedonline.com/game/1130/Tetris.html
[15] http://sheng.phy.nknu.edu.tw/wjs21flash040.htm