Information Sciences 276 (2014) 343–353

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A steganographic method based on tetris games

Zhan-He Ou, Ling-Hwei Chen *

Department of Computer Science, National Chiao Tung University Hsinchu, Taiwan, ROC

ARTICLE INFO

Article history:
Received 26 April 2013
Received in revised form 8 November 2013
Accepted 21 December 2013
Available online 31 December 2013

Keywords:
Steganography
Tetrimino
Tetris
Undetectability

ABSTRACT

Although various steganographic methods have been proposed that use distinct cover media, using games to hide data remains a recent development. The study presents a steganographic method based on online Tetris games, in which secret messages are embedded using a generated tetrimino sequence. Each time a person plays an online Tetris game, a new tetrimino sequence should be generated. The proposed method meets this requirement, generating a distinct stegoed tetrimino sequence for each game played. In addition, the study presents a scenario and a simulator for the proposed method. Theoretical proof and experimental results are provided to demonstrate that the proposed method is undetectable.

1. Introduction

Steganography is the art of secret communication within information systems, allowing users to conceal their information by using various cover media. The primary requirement of steganography is undetectability [4], which makes it impossible to determine whether a secret message is embedded in a system.

In the past decade, images have become the most popular cover media. Various steganographic methods enable users to hide messages in the spatial [16,23] or frequency [26] domains; however, few methods enable users to hide messages in lossy compressed images such as vector quantized images [1,22]. Moreover, certain methods allow the receiver to recover the original cover image after extracting the secret message [13–15,18] and various steganalysis techniques are used [2,9,25] to test the security of these steganographic methods.

Several recent studies have explored using games to practice steganography [5–8,10–12,17,20,24]. In 2006, Hernandez-Castro et al. [10] proposed using a steganographic method to hide secrecy messages in the moves taken during extensive-form games; the proposed method involved the assumption that a searching algorithm could provide a winning score for each move. A move that produced a higher score was considered the more favorable move. Based on the scores generated during the game, the method provides a sorted-move sequence; when a player wants to send a secret message i by using $1 \leqq i \leqq n$, he or she selects the i_{th} move in the sorted move sequence to represent the secret i. However, the optimal move cannot always be chosen when using this method; thus, the stegoed moves might be abnormal, causing the method to be detected. Furthermore, various extensive-form games require distinct searching algorithms, and developing such algorithms is difficult.

Kieu et al. [11] proposed a Sudoku-based image hiding scheme, using a spatial domain in which each pixel pair $\langle i,j\rangle$ in a cover image could hide a secret number $s \in \{1,2,\ldots,9\}$. First, in this scheme, a solved Sudoku 9×9 matrix was expanded 29×29 times into a 261×261 look-up table.

* Corresponding author. Tel.: +886 035712121 54744.
E-mail addresses: id4922.cs96g@nctu.edu.tw (Z.-H. Ou), lhchen@cc.nctu.edu.tw (L.-H. Chen).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.12.024

The gray values of the pixel pair $\langle i,j\rangle$ were assumed v_i and v_j and (v_i, v_j) was considered an index for the look-up table. If the number in (v_i, v_j) was not the same as the secret number s, then the hiding scheme searched the nearest element (v'_i, v'_j) in the look-up table for number s. The gray values of pair $\langle i,j\rangle$ were replaced with v'_i and v'_j. Because of the properties of the Sudoku matrix, the changed values $(v_i - v'_i, v_j - v'_j)$ were less than $(\pm 2, \pm 2)$. However, both the sender and receiver should share the same Sudoku grid information. On the other hand, this scheme is weak against compression attacks.

In 2009, Farn and Chen [7–8] proposed two steganographic methods in two types of puzzle games. The first method involves hiding a secret message in each attached semi-cycle based on its positions and types on a jigsaw puzzle image [7]. The second method involves hiding a secret message by using piece permutation in a jig-swap puzzle game [8]. Because these methods do not involve embedding secret messages in the pixel values, they are robust against compression attacks. However, the size of a puzzle piece must be recognizable to players; thus, these methods are limited by the size of a puzzle image.

Ou and Chen [19] presented a method to hide data by using Tetris, but this method yielded various problems. For example, secret messages are manually extracted, and the preshared secret key must be transmitted using a secret channel. In the current study, a steganographic method is presented and Tetris games are used as cover media to resolve the previous problems. The proposed method is undetectable and exhibits great potential.

Tetris is a popular game that debuted in 1984. The game interface (Fig. 1(a)) comprises a playing field, score box, and next-piece preview field. The playing field is an interactive area, in which a player controls a falling puzzle piece by shifting it left and right or rotating it 90° as it falls into the playing field; this "playing piece" is the only controllable piece in the field of play. At the beginning of each game, the first playing piece appears above the playing field before it enters the playing field; the second piece appears in the next-piece preview pane at the top left of the screen. After the game begins, succeeding pieces continually fall. When a playing piece reaches the ground of the playing field, it is fixed in place, becoming an extension of the ground and halting the progress of subsequent falling pieces. This procedure repeats until the game ends.

A player must control the falling piece so that it is guided or "dropped" to a specific point on the ground. When several fixed pieces fill one or more horizontal lines, those lines disappear and the fixed pieces above the disappeared lines fall to new fixed positions, as shown in Fig. 1(b) and (c). The system awards a score for each fixed piece placed, adding the score to the total displayed in the score box at the top of the game window. When fixed pieces accumulate above the playing field,

Fig. 1. An example to illustrate Tetris. (a) Tetris interface. (b) Two filled horizontal lines. (c) The disappeared lines and new fixed positions. (d) Game over.

the game ends, as shown in Fig. 1(d). Most Tetris games provide a "Play Again" function which allows the player to begin a new game. The player can continue to play additional games until he or she leaves the game system.

A Tetris puzzle piece is called a tetrimino. Tetriminos comprise seven distinct shapes, each of which is formed by four orthogonally contacted squares. Based on the appearances of these shapes, they are called I, J, L, O, S, T and Z (Fig. 2). Tetriminos are randomly generated during game play and their order is called a tetrimino sequence.

Two tetrimino generators operate in Tetris. The first is a dice-like generator that generates a tetrimino as if a 7-sided die was rolled. The second is a shuffle-like generator that generates a tetrimino sequence as if seven shuffled cards were dealt, each representing one of the seven tetriminos. The generators randomly generate tetrimino sequences throughout each game; thus, the generated tetrimino sequences are distinct in each game played.

The stegoed tetrimino sequence appears as a randomly generated sequence that varies in each game; however, the same secret message can be embedded. The secret message is extracted by generating screenshots of the falling tetriminos while the game is being played.

The following is a scenario for the proposed method: a sender uploads a Tetris game on the Internet by using the proposed embedding method and anyone can play this game. The receiver uses the proposed extractor to obtain the secret message during game play and no other player or warden is aware of the secret communication.

The remaining sections of this paper are organized as follows. Section 2 details the proposed embedding and extracting methods. Section 3 introduces the proposed scenario and simulation. Section 4 presents security, proof of theory, and a description of the experiments. Section 5 provides a conclusion.

2. The proposed steganographic method

This paper proposes a steganographic method to conform to two kinds of tetrimino generators. Because the two generators function similarly, for convenience, only the method based on the dice-like generator is described in full. However, a simulator that uses a shuffle-like generator is described in Section 3.2.

2.1. Embedding process

Regarding the dice-like generator, tetriminos are generated as if a 7-sided die were rolled. The number of distinct sequences is 7^n, where n is the number of tetriminos; thus, each tetrimino sequence can represent a 7-based number with n digits. The seven tetrimino shapes I, J, L, O, S, T, and Z are represented by the numbers 0, 1, 2, 3, 4, 5, and 6, respectively. For example, a tetrimino sequence of O, L, T, Z, L, L, O, L corresponds to the 7-based number $32{,}562{,}232_7$.

However, simply transferring a secret message by using a tetrimino sequence is unsafe. This means that a player must always play the same tetrimino sequence before the secret message is changed. To ensure that a stegoed tetrimino sequence appears as if it were generated by a standard Tetris game, embedding a message in a tetrimino sequence should satisfy two requirements. First, when a game begins, a new tetrimino sequence should be generated regardless of whether a secret message is changed. Second, after the stegoed tetrimino sequence is popped out, the game cannot be stopped before the player quits the game.

The embedding process comprises three phases, and Fig. 3 shows its block diagram. In the first phase, a seed R_s is randomly generated. In the second phase, which involves random stegoed sequence generation, R_s is used to generate a stegoed tetrimino sequence, which will be popped out to the player. When the entire stegoed tetrimino sequence is popped out, in the final, or game preservation phase, random tetriminos continue to be generated until the player quits the game.

2.1.1. Initialization

When a game begins, a random number generator generates a random seed R_s, which is designed to randomize the original secret message in the next phase. R_s is encrypted using the public key of the receiver based on a public-key cryptography system [21], and the corresponding m tetriminos of the encrypted R_s are popped out. After the corresponding tetriminos are popped out, random tetriminos continue to be generated and popped out until the player ends the game. When the player clicks "Play Again", the second phase begins.

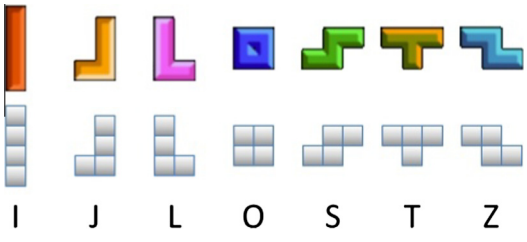

Fig. 2. Seven different shapes of tetriminos.

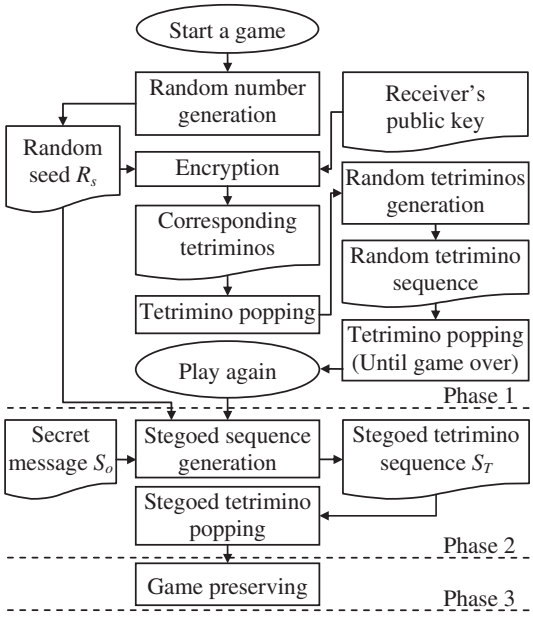

Fig. 3. Block diagram of the proposed embedding process.

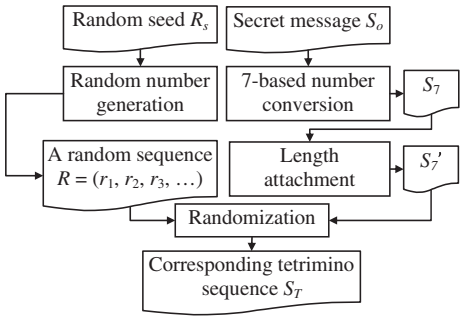

Fig. 4. Stegoed tetrimino sequence generation.

Note that m represents preshared information, R_s is distinct in each game played, and only the receiver who possesses the private key can decrypt the encrypted R_s.

2.1.2. Stegoed tetrimino sequence generation

In the second phase, the stegoed tetrimino sequence is generated and popped out to the player. Fig. 4 shows the generation process. First, the secret message S_o is converted into a 7-based number $S_7 = (s_1, s_2, s_3, \ldots, s_n)_7$, where s_i is the i_{th} digit of this number.

Because the receiver requires a secret message length n for extraction, the length n is attached to S_7 to form S_7', as shown in Fig. 5(a). S_7' contains k segments, each of which uses a continuous indicator c_i and a subsequence Sub_i. If $c_i = 0$, the segment is the final segment; otherwise, $c_i = 1$. The final segment uses an additional length indicator P between the continuous indicator c_k and subsequence Sub_k. P is a 7-based number that comprises h digits and records the length of the last subsequence.

Note that h represents preshared information, and the sequence S_7 is divided into k subsequences in which $k = \lceil n/7^h \rceil$. Each of the first $k-1$ subsequences comprises 7^h 7-based digits, and the last subsequence exhibits a length less than 7^h. Based on this structure, the length n can be calculated as $n = (k-1) \times 7^h + P_{10}$, where P_{10} is the decimal number of P. To facilitate illustration, S_7' is denoted as $S_7' = (a_1, a_2, a_3, \ldots, a_{n+k+h})$. Based on this structure, a secret message of any size can always be converted to S_7'.

For example, a secret message $76{,}543{,}210_{10}$ is first converted into the corresponding 7-based number $1{,}616{,}415{,}022_7$. Assume that h (the length of P) is 1, and the 7-based number is separated in two segments. The first segment contains the first seven digits (1,6,1,6,4,1,5) led by a continuous indicator of 1, that is, (1,1,6,1,6,4,1,5). The final segment contains a

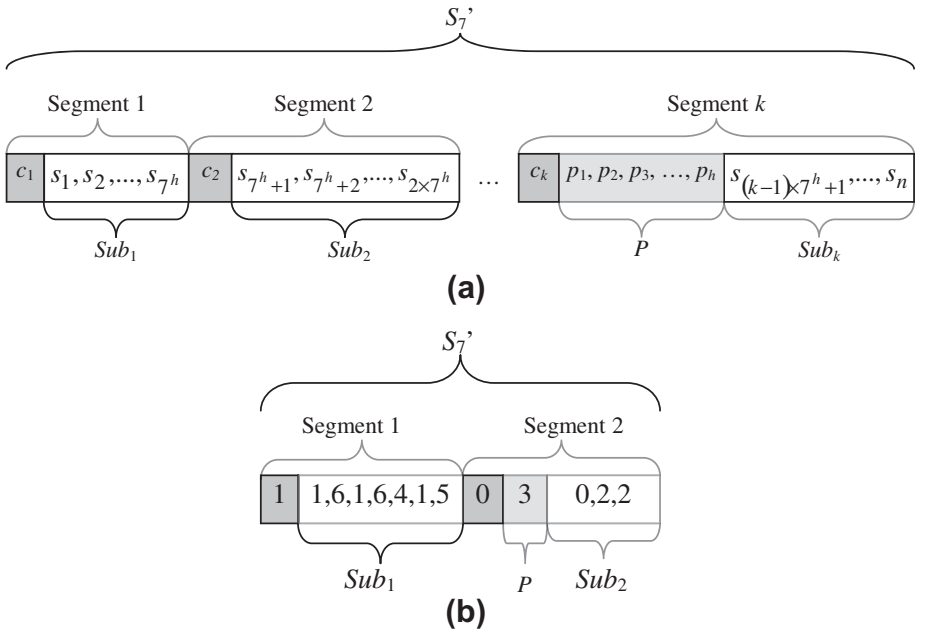

Fig. 5. Structure of S_7'. (a) The Structure of S_7'. (b) The example of S_7' with the secret message 76543210_{10} (1616415022_7).

continuous indicator of 0, a length indicator of 3, and three digits (0,2,2), yielding (0,3,0,2,2). The resulting S_7' is (1,1,6,1,6,4,1,5,0,3,0,2,2), as shown in Fig. 5(b).

After S_7' is obtained, the random seed R_s is used to generate a random sequence $(r_1, r_2, r_3, \ldots)$, where $r_i \in \{0,1,2,3,4,5,6\}$. S_7' is then randomized by using the following equation:

$$t_i = (a_i + r_i) \bmod 7. \tag{1}$$

The randomized sequence is denoted as $T = (t_1, t_2, t_3, \ldots, t_{n+k+h})$, and the corresponding tetrimino sequence, S_T, is the stegoed tetrimino sequence.

If the sender hides no information, S_7' is a sequence of $h + 1$ zero. The first zero indicates the last subsequence, and the remaining h zeros indicate that the last subsequence contains zero elements.

The stegoed tetrimino sequence S_T is popped out sequentially. If the game ends before the entire S_T is popped out, the remaining tetriminos continue to pop out after the player clicks the "Play Again" button. After all tetriminos are popped out, the embedding process is completed and the final phase begins.

Each time a game starts (i.e., not conducted by "Play Again"), the random seed R_s is randomly generated. This produces r_i elements in a random sequence distinct from those generated in previous games. Thus, S_T is new, even if the secret message remains the same.

2.1.3. Game preservation

To prevent an abnormal halt to the game, the game preservation phase automatically begins after the entire stegoed tetrimino sequence is popped out. During this phase, the system continues to generate random tetriminos until the player quits the game.

2.2. Extraction process

The proposed extraction process involves two phases, and its block diagram is shown in Fig. 6. By accessing the captured screenshots, this extraction process can automatically extract the secret message while the receiver plays the game.

The first phase is conducted after the player begins a game. The start button is detected to indicate the game start. Most Tetris games include a game start button that disappears after a player clicks it. Because this button appears in the same window location, the extraction process can detect this button based on pattern matching.

The extraction process is used to collect m falling tetriminos. Because the tetriminos always appear in the top center of the playing field, the extraction process can recognize these tetriminos by matching patterns. The collected tetriminos are converted into corresponding numbers and these numbers are decrypted by using the private key of the receiver, yielding the random seed R_s.

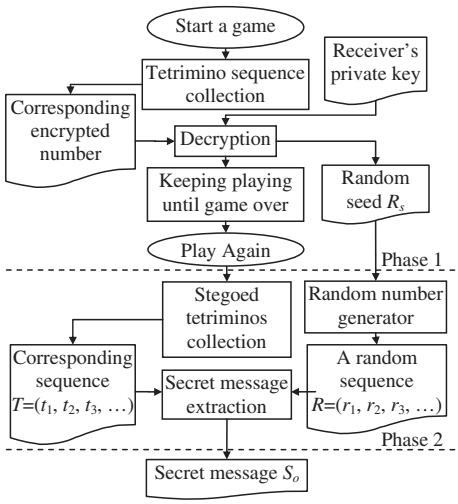

Fig. 6. The block diagram of the extraction process.

After the random seed is obtained, the extraction process pauses until the game ends. The game-over event can be detected when the "Play Again" button appears, which occurs in the same window area after each game ends.

The second phase begins after the player presses the "Play Again" button. This event is detected when the "Play Again" button disappears. In the extraction process, R_s is used to generate random number sequence $R = (r_1, r_2, r_3, \ldots)$. The tetriminos that appear during the game are collected and transformed into a corresponding 7-based number sequence, denoted as $T = (t_1, t_2, t_3, \ldots)$. The sequence $S_7' = (a_1, a_2, a_3, \ldots)$ can then be derived using the following equation:

$$a_i = (t_i - r_i) \bmod 7. \tag{2}$$

Procedure 1 describes the process by which the secret message is extracted from the obtained S_7'.

Procedure 1. Secret message extraction

Step 1. Let $k = 1$, $i = 1$
  Set the first continuous indicator $c_1 = a_1$
Step 2. If $c_1 = 0$, go to Step 6
Step 3. Concatenate 7^h a_j with $j = i + 1, \ldots, i + 7^h$ as Sub_k
Step 4. ∗Prepare the next subsequence∗
  $k = k + 1$
  $i = i + 7^h + 1$
  Set the k_{th} continuous indicator $c_k = a_i$
Step 5. If $c_k = 1$, go to Step 3
Step 6. ∗Obtain the last subsequence∗
  Concatenate h a_j with $j = i + 1, \ldots, i + h$ as P
  $i = i + h + 1$
  Convert P to a decimal number P_{10}
Step 7. Concatenate P_{10} a_j with $j = i, i + 1, \ldots, i + P_{10} - 1$ as Sub_k
Step 8. Concatenate all received subsequences as S_7
Step 9. Convert S_7 into the original secret message S_o
Step 10. Inform the receiver that the extraction has ended.

3. Scenario and simulator

In this section, a scenario is provided to demonstrate how the proposed method operates. A simulator is then implemented to demonstrate the feasibility of the proposed method. It can be downloaded from [27].

3.1. Proposed scenario

This study presents a scenario to illustrate how a secret message is transmitted through an online Tetris game. During the scenario, the following are required:

(1) A public-key cryptography system must be used in which the receiver possesses a pair of keys (public and private keys) and the sender can access the public key.
(2) The length (m) of the tetrimino sequence that corresponds to the encrypted random seed R_s and the length (h) of the indicator P must be shared by both the sender and receiver.
(3) The sender must create an online Tetris game website similar to [28,30], for players to access.

Additional procedures in the scenario are described as follows:

(1) When the sender wants to send a secret message, he or she inputs the public key provided by the receiver and the secret message offline in the proposed Tetris game, then invites the receiver to play.
(2) The proposed method is used to embed the secret message in a tetrimino sequence when a game begins.
(3) The receiver accesses the website and plays the Tetris game online.
(4) When the embedding process receives a game-start request, the first phase of the embedding process begins to generate the random seed R_s and encrypt R_s to pop the corresponding tetriminos.
(5) The receiver plays the Tetris game, and the proposed extraction process is used to collect the tetriminos and decrypt the random seed.
(6) After the random seed is received by the receiver, the receiver continues to play until the game ends.
(7) The Tetris game displays a "Play Again" button, which must be clicked by the receiver to advance to the second phase.
(8) A stegoed tetrimino sequence is generated during the second phase of the embedding process and popped out. The extraction process is then used to collect the played tetriminos and reconstruct the secret message.
(9) If the game ends before the extraction process informs the receiver all secret message collected, the receiver can press "Play Again" to collect the popped tetriminos.
(10) After the extraction process is used to collect the final tetrimino to pass through the length indicator, the secret message is extracted.

3.2. Proposed simulator

To explain the proposed method in detail, a simulator was established on the study website. This simulator comprised the embedding interface, proposed Tetris game, and extraction interface.

The embedding interface is used by the sender to input the secret message and the receiver's public key, as shown in Fig. 7. Based on the entered information, the proposed Tetris game is used to generate corresponding stegoed tetrimino sequences.

The simulator uses an RSA public-key cryptography system [21]. This system requires two distinct prime numbers, p and q, and a pair of numbers (d,e), where $de \equiv 1 \pmod{(p-1)(q-1)}$. Note that e and $(p-1)(q-1)$ are relative prime. For a secret message M, the cipher text C can be calculated using the following equation:

$$C \equiv M^e \pmod{N} \tag{3}$$

where $N = p \times q$. In addition, the encrypted message can be decrypted by using the following equation:

$$M \equiv C^d \pmod{N}. \tag{4}$$

In this public-key cryptography system, the public key is (N,e), and the private key is (N,d).

The proposed method simulates a standard Tetris game by using a shuffle-like generator. Note that, besides it counts the number of played tetriminos and evaluates constant falling speed, the proposed method exhibits nearly identical functions as the Tetris game described in [31]. Fig. 8 shows the proposed Tetris interface.

Note that the secret message is controlled through the embedding interface and must be constant before the sender changes the content. The Tetris game always pops out a new tetrimino sequence when a new game begins, even if the secret message remains unchanged.

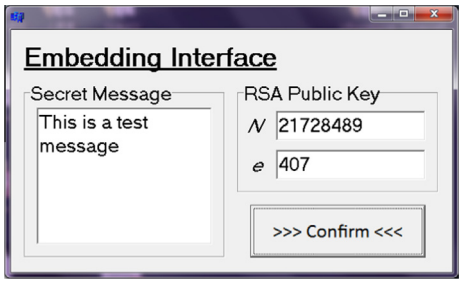

Fig. 7. The embedding interface.

Fig. 9(a) shows the proposed extraction interface, comprising the private key used by the receiver, an extracted secret message, and a blank area used to capture screen shots. The receiver must simultaneously start the extraction interface and Tetris game and the receiver must place the proposed Tetris interface in the capture area on the extraction interface, as shown in Fig. 9(b). The receiver subsequently presses the record button on the extraction interface to begin capturing screen shots, as shown in Fig. 9(c). The extractor waits to begin extraction until the receiver clicks the "New Game" button on the proposed Tetris interface, as shown in Fig. 9(d).

When the receiver plays the Tetris game, the extractor detects the played tetriminos based on the captured screenshots, converting them to a secret message, as shown in Fig. 9(e). When the secret message is obtained, a "Finish" message is displayed on the extraction interface, as shown in Fig. 9(f). The game continues to run after the extraction is completed and the receiver can continue playing or quit the game.

4. Security discussion

A secure steganography scheme is statistically undetectable [3]. In this section, the proposed method is proven undetectable based on theoretical proof and experimental results.

4.1. Theoretical proof for undetectability

This study theoretically proves that the proposed method is undetectable. First, assume that for a dice-like generator, the distribution of different shapes in a tetrimino sequence generated by a standard Tetris game is uniform. Similarly, if the distribution of various shapes in a stegoed tetrimino sequence is uniform, the proposed method is undetectable. The following theorem proves this point.

Theorem. *The distribution of the seven shapes in a stegoed tetrimino sequence is uniform.*

Proof. Let $\Pr^T(t)$ denote the probability of t in a stegoed tetrimino sequence T, $t \in \{0,1,\ldots,6\}$.

$\Pr^{S'}(a)$ denotes the probability of a in sequence S'_7, $a \in \{0,1,\ldots,6\}$.

$\Pr^R(r)$ denotes the probability of r in the random sequence R, $r \in \{0,1,\ldots,6\}$.

Because S'_7 and R are independent, according to (1), the probability of t in the stegoed sequence T can be expressed as follows:

$$\Pr^T(t) = \sum_{a=0}^{6} \Pr^{S'}(a) \times \Pr^R(r)$$

where $t = (a + r) \bmod 7$.

Because the random sequence R is a uniform distribution, then $\Pr^R(r) = \frac{1}{7},\ \forall r$. This implies the following:

$$\Pr^T(t) = \frac{1}{7} \sum_{a=0}^{6} \Pr^{S'}(a).$$

Because the summation of all probabilities must be 1, the following is derived:

$$\Pr^T(t) = \frac{1}{7},\ \forall t.$$

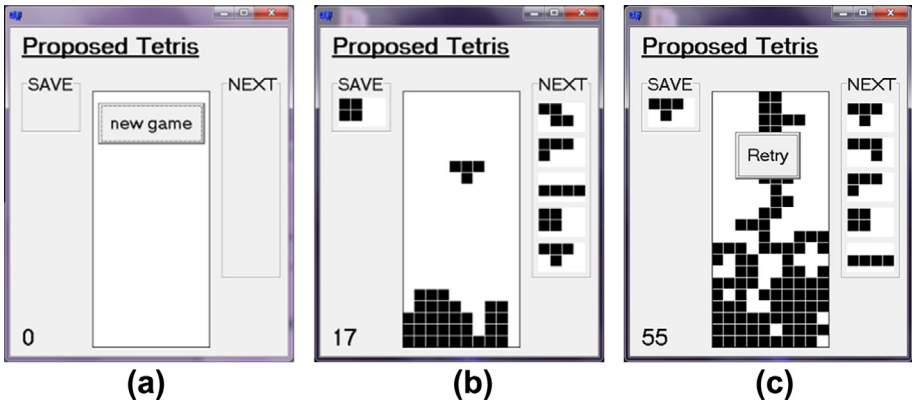

Fig. 8. The proposed Tetris interface. (a) Beginning of the proposed Tetris. (b) During the game play. (c) When the game is over.

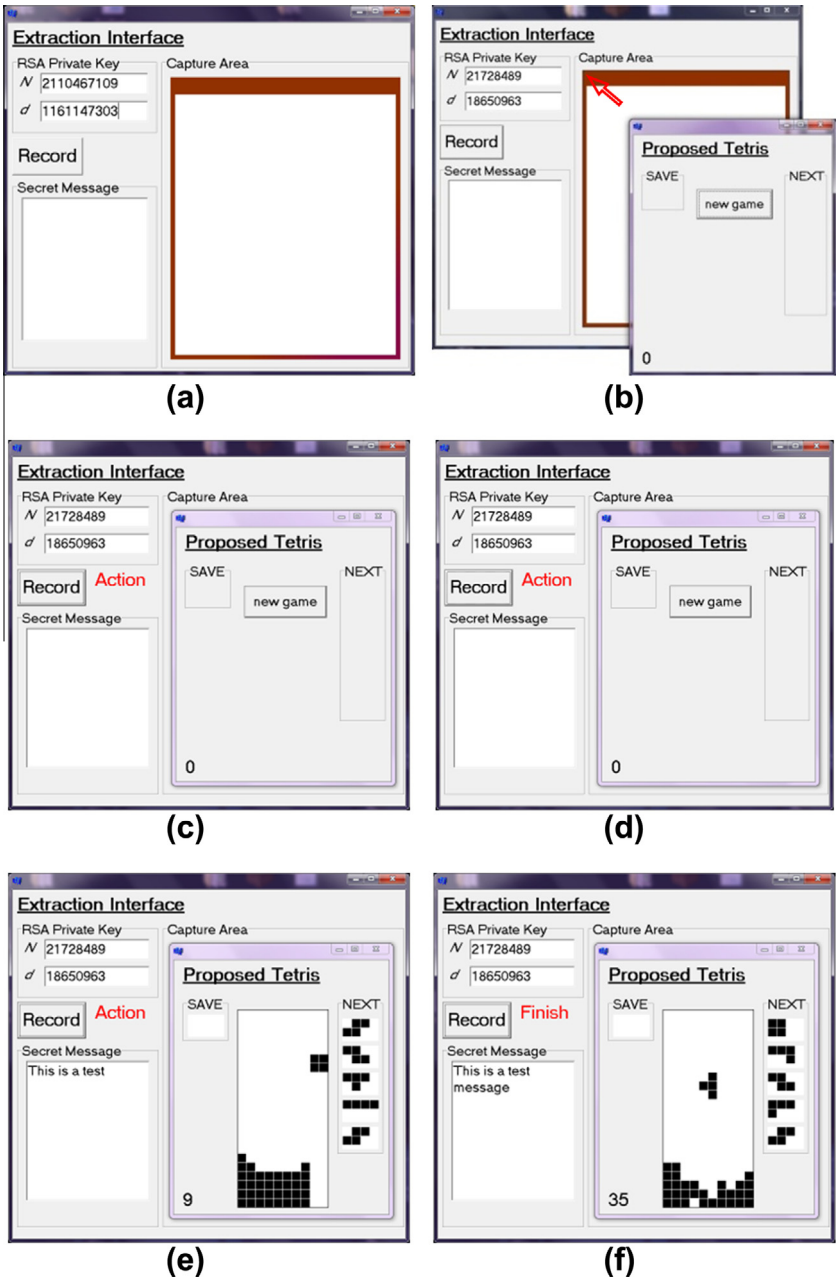

Fig. 9. The proposed extraction simulator. (a) The extraction interface. (b) Put the proposed Tetris interface into the capture area. (c) Press the record button to start capturing the screen shots. (d) Press the new game button to start playing the Tetris. (e) During the extraction. (f) End of the extraction.

Therefore, the stegoed tetrimino sequence T is a uniform distribution. □

4.2. Experiments

The current findings demonstrate that a stegoed tetrimino sequence appears similar to those generated in standard Tetris games. Experiments were used to record several tetrimino sequences from four Tetris games. In each game, 10,000 tetriminos were recorded for comparison. The tested games were the Tengen version of Tetris on the Famicom video game console [29], "N-blox" from the Tetris official website [32], and two free shared Tetris games obtained from the Internet [28,30]. For comparison, a stegoed tetrimino sequence comprising 10,000 pieces was examined.

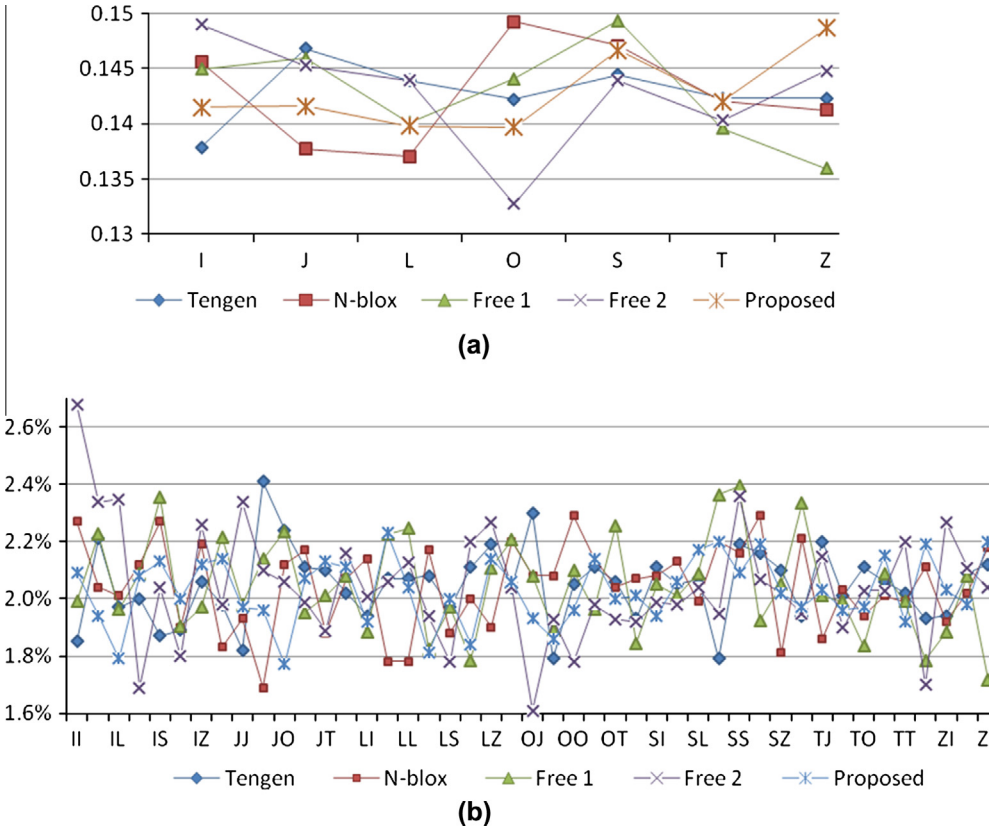

Fig. 10. The appearing probabilities of two different record methods based on the dice-like generation. (a) The appearance probabilities for single tetriminos. (b) The appearance probabilities of two consecutive tetriminos.

The distribution similarity was tested using an entropy value, which measured the associations among the sample distribution. Let X be a discrete random variable with n possible samples $\{x_1, x_2, \ldots, x_n\}$. The entropy value is defined as follows:

$$H(X) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i), \tag{5}$$

where $p(x_i)$ is the probability of x_i. If X is a uniform distribution, the entropy is the maximal number $\log_2 n$.

Three different methods were used to record the frequencies at which the tetriminos appeared. These methods were used to count the number of consecutive tetriminos that appeared in the playing field, whether two or three consecutive tetriminos appeared. Fig. 10 and Table 1 show the results.

As shown in Fig. 10, the proposed method indicates that the probability of each tetrimino appearing in every game is between 0.13 and 0.15. In addition, the probability of each pair of consecutive tetriminos appearing for each game is between 0.016 and 0.026. In a uniform distribution of seven symbols, each symbol exhibits a 1/7 probability (approximately 0.143) of appearing, and the probability of each pair of symbols appearing is 1/49 (approximately 0.020). Fig. 10 shows that the four Tetris games and the proposed game exhibited nearly uniform distributions. Thus, the detected tetrimino counts suggest that the proposed stegoed tetrimino sequence is indistinguishable from the sequences generated in a standard Tetris game.

Table 1 displays the entropy values of various recording methods used in each Tetris game. The first row shows that the entropy value of probabilities of single tetriminos is 2.81 (approximately $log_2 7$). This indicates that the distributions of the seven tetrimino shapes were uniform in all games, including the proposed game. The entropy values of the probabilities of

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

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

two and three consecutive tetriminos appearing are shown in the second and third rows, respectively. These values indicate that the distributions of two or three consecutive tetriminos are nearly uniform, implying that the proposed Tetris game is similar to standard Tetris games.

5. Conclusion

As additional types of carriers are employed to embed data in a steganographic system, the system becomes increasingly secure. A steganographic method involving Tetris games was presented herein, in which a secret message was hidden in a tetrimino sequence. By using the "Play Again" function, the proposed method is used to extend the tetrimino sequence and hide long secret messages. According to the theoretical proof and experimental results, the findings show that the proposed method is undetectable. In addition, a scenario and simulation were provided to demonstrate the feasibility of the proposed method.

Acknowledgement

This work is supported in part by National Science Council of Republic of China under grant NSC-100-2221-E-009-140-MY2.

References

[1] C.C. Chang, C.Y. Lin, Y.P. Hsieh, Data hiding for vector quantization images using mixed-base notation and dissimilar patterns without loss of fidelity, Inform. Sci. 201 (2012) 70–79.
[2] Y.S. Chen, R.Z. Wang, Steganalysis of reversible contrast mapping watermarking, IEEE Sig. Process. Lett. 16 (2) (2009) 125–128.
[3] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, second ed., Morgan Kaufman, 2008. p. 54.
[4] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, second ed., Morgan Kaufmann, 2008. pp. 469–495.
[5] A. Desoky, M. Younis, Chestega: chess steganography methodology, Sec. Commun. Netw. 2 (6) (2009) 555–566.
[6] M. Diehl, Secure covert channels in multiplayer games, in: Proceedings of the 10th ACM Workshop on Multimedia and Security, 2008, pp. 117–122.
[7] E.J. Farn, C.C. Chen, Jigsaw puzzle images for steganography, Opt. Eng. 48 (7) (2009) 077006.
[8] E.J. Farn, C.C. Chen, Novel steganographic method based on jig swap puzzle images, J. Electron. Imag. 18 (1) (2009) 013003.
[9] G. Gul, F. Kurugollu, SVD-based universal spatial domain image steganalysis, IEEE Trans. Inform. Foren. Secur. 5 (2) (2010) 349–353.
[10] J.C. Hernandez-Castro, I. Blasco-Lopez, J.M. Estevez-Tapiador, A. Ribagorda-Garnacho, Steganography in games: a general methodology and its application to the game of go, Comput. Secur. 25 (1) (2006) 64–71.
[11] T.D. Kieu, Z.H. Wang, C.C. Chang, M.C. Li, A Sudoku based wet paper hiding scheme, Int. J. Smart Home 3 (2) (2009) 1–12.
[12] H.L. Lee, C.F. Lee, L.H. Chen, A perfect maze based steganographic method, J. Syst. Softw. 83 (12) (2010) 2528–2535.
[13] J.D. Lee, Y.H. Chiou, J.M. Guo, Lossless data hiding for VQ indices based on neighboring correlation, Inform. Sci. 221 (2013) 419–438.
[14] J.H. Lee, M.Y. Wu, Reversible data-hiding method for palette-based images, Opt. Eng. 47 (4) (2008) 047008.
[15] D.C. Lou, C.H. Hu, LSB steganographic method based on reversible histogram transformation function for resisting statistical steganalysis, Inform. Sci. 188 (2012) 346–358.
[16] W. Luo, F. Huang, J. Huang, Edge adaptive image steganography based on LSB matching revisited, IEEE Trans. Inform. Foren. Sec. 5 (2) (2010) 201–214.
[17] S.J. Murdoch, P. Zieliński, Covert channels for collusion in online computer games, in: Proceedings of Information Hiding 6th International, Workshop, vol. 3200, 2005, pp. 419–429.
[18] Z. Ni, Y.Q. Shi, N. Ansari, W. Su, Reversible data hiding, IEEE Trans. Circ. Syst. Video Technol. 16 (3) (2006) 354–362.
[19] Z.H. Ou, L.H. Chen, Hiding data in Tetris, in: Proceedings of IEEE International Conference on Machine Learning and Cybernetics, vol. 1, 2011, pp. 61–67.
[20] M.H. Shirali-Shahreza, M. Shirali-Shahreza, Steganography in SMS by Sudoku puzzle, in: Proceedings of Computer Systems and Applications, 2008, pp. 844–847.
[21] D.R. Stinson, The RSA Cryptosystem and Factoring Integers in Cryptography Theory and Practice, third ed., Chapman & Hall/CRC, 2006. pp. 161–232.
[22] W.J. Wang, C.T. Huang, C.M. Liu, P.C. Su, S.J. Wang, Data embedding for vector quantization image processing on the basis of adjoining state-codebook mapping, Inform. Sci. 246 (2013) 69–82.
[23] Z.H. Wang, C.C. Chang, M.C. Li, Optimizing least-significant-bit substitution using cat swarm optimization strategy, Inform. Sci. 192 (2012) 98–108.
[24] S. Zander, G. Armitage, P. Branch, Covert channels in multiplayer first person shooter online games, in: Proceedings of Local, Computer Networks, 2008, pp. 215–222.
[25] J. Zhang, D. Zhang, Detection of LSB matching steganography in decompressed images, IEEE Sig. Process. Lett. 17 (2) (2010) 141–144.
[26] L. Zhang, H. Wang, R. Wu, A high-capacity steganography scheme for JPEG2000 baseline system, IEEE Trans. Image Process. 18 (8) (2009) 1797–1803.
[27] http://www.debut.cis.nctu.edu.tw/Demo/StegoTetris/StegoTetris.htm.
[28] http://www.sheng.phy.nknu.edu.tw/wjs21flash040.htm.
[29] http://www.allgame.com/game.php?id = 14911.
[30] http://www.playedonline.com/game/1130/Tetris.html.
[31] http://www.tetrisfriends.com/games/Marathon/game.php.
[32] http://www.tetrisfriends.com/games/NBlox/game.php.