Düzce Üniversitesi
Bilim ve Teknoloji Dergisi

Research Article

A Snake Game Steganography Method based on S-Boxes

Esra ŞATIR[a,*], Kubilay GÜNER[b]

[a] *Department of Computer Engineering, Engineering Faculty ,Duzce University, Duzce, TURKEY*
[b] *Department of Computer Engineering, Engineering Faculty ,Duzce University, Duzce, TURKEY*
* *Corresponding author's e-mail address: esrasatir@duzce.edu.tr*

Abstract

Steganography is the art and science of concealing the existence of information within seemingly innocuous carriers or it is a communication method in such a way that the presence of a message cannot be detected. There are a variety of digital carriers or places like images, audio files, text, html, etc. where data can be hidden. In this study, we proposed a game steganography method using snake game as the cover medium. We aim to hide the secret data by estimating the each move of the bait called "vitamin". In other words, we compute coordinates of the vitamin according to the unit components of secret data. Meanwhile, we plan to complicate the game platform and establish a nonlinear relationship between the vitamin location and secret data by employing DES S-boxes. These operations render the proposed scheme more resilient against the possible steganographic attacks and make the extraction procedure more complicated. Therefore, security and imperceptibility have been kept as the focus of interest in the scope of this study. Besides, significant capacity rates have been obtained by each move. The performed experiments offer significant results to support these claims.

Keywords: *Steganography, game steganography, game theory, snake game, DES S-Boxes*

I. Introduction

THE revolution in digital information has created new challenges for sending a message in a safe and secure way. Whatever method we choose, the most important question is its degree of security. Numerous approaches have been developed for addressing the issue of information security such as cryptography and steganography. Cryptography provides an obvious approach to render the information secure. It scrambles the secret message, such that it becomes meaningless to eavesdroppers. However, this is not always adequate in practice as the encrypted content itself raises attention. Regardless how strong is the encryption algorithm, given enough time and tools, it could be broken. Furthermore, some cases require sending information without anyone noticing that the communication happened. In such cases, steganography is the answer [1].

Steganography is the art and science of concealing the existence of information within seemingly innocuous carriers (e.g. images, audio files, text, html, etc.) or, as defined like *"of communicating in such a way that the presence of a message can not be detected''* [2]. Two important properties, undetectability and embedding capacity, should be considered when designing a steganographic

Received: 22/08/2016, Revised: 14/11/2016, Accepted: 19/11/2016

algorithm. There is always a trade-off between these two properties. [3]. Besides, the success of steganography methods depends upon the carrier medium not to raise attention. [4].

There are a variety of digital carriers or places where data can be hidden. Data may be embedded in files at imperceptible levels as noise. Properties of images can be manipulated including luminescence, contrast and colours. In audio files, small echoes or slight delays can be included or subtle signals can be masked with sounds of higher amplitude. Information can be hidden in documents by manipulating the positions of the lines or the words. When HTML files are written, web browsers ignore spaces, tabs, certain characters and extra line breaks. These could be used as locations to hide information. Messages can be retrieved from text by taking, for example the second letter of each word and using them to produce the hidden message. This is called a null cipher or open code. Information can be hidden in the layout of a document for example certain words in a piece of text can be shifted very slightly from their positions and these shifted words can then make up the hidden message. The way a language is spoken can be used to encode a message such as pauses, enunciation's and throat clearing [5].

Accordingly, [6], [4] and [3] are some recent examples of image steganography methods in the literature, attempting to balance the embedding capacity and imperceptibility. [7] is one of the recent examples of audio steganography that is designed under the restriction of MP3 compression standard. [8-10] are the recent examples of limited text steganography methods in the literature since concealing data in a text document is difficult because of the similar back and fore ground structures. [11], [12] are some examples of a new area; network steganography which uses IP packets for data concealing. Also there are some miscellaneous studies like in [13] and [14]. In [13], the secret message is embedded by employing the various animations in Power Point, benefitting the fact that any animation did not alter the content of a PowerPoint file. In [14], a new system of information transfer within Microsoft Word documents via JPEG images was proposed.

Employing games is another application area of steganography. Some of the limited studies in the literature have been explained in the following lines. Hernandez-Castro et al. (2006) presented a game steganography method that employs game comments and moves to conceal secret data. They focused on the games like Chess, Backgammon, Go and etc. The main idea was to compute for each played position in the game for all the movements which are over a certain threshold value. Then, codifying some bits of the secret message in the selection of the actual played move [2].

In 2009, Desoky proposed a game steganography method called Chestega. Chestega employed chess related covers such as training documents, game analysis, news articles, etc. The secret message was concealed in three steps: First, the aspect of the game was chosen to hold the steganographic code for determination of encoding parameters like chessboard, pieces, moves, etc. Meanwhile, the communicating parties should agree on these parameters. Second, the secret message was interpreted via these encoding parameters. And in the final step, the secret message was camouflaged in a chess game cover. They mentioned that the hidden message did not cause any distortion and the security of the Chestega neither relied on the secrecy of the technique nor required a stego-key [15].

In 2010, Lee et al. proposed a steganographic method using a perfect maze concept. A rectangular maze is denoted as $m \times n$ maze (m cells in width and n cells in height). The maze is called perfect if there exists one and only one path between any two cells. The main idea of the proposed method was to consider multipaths rather than only the solution path to gain more embedding capacity. They first generated a perfect maze with HKMG (Hunt-and-Kill Maze Generating) algorithm. Then, they selected some cells as the start cells and one cell as the common end one of the multipaths. Finally, they solved the perfect maze to obtain the corresponding multipaths and ordered them according to their starting cells from top to bottom and then left to right. By considering the path sequence, they traced each path from the start cell to locate all embeddable cells to obtain a sequence of embeddable cells. Then, the data was embedded according to this sequence [16].

In 2014, Ou and Chen proposed a steganographic method based on online Tetris games. In this method, the secret message was embedded using a generated tetrimino, a Tetris puzzle piece, sequence. The proposed method generated a distinct stegoed tetrimino sequence for each played game. A scenario of the method is: 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 extraction method to obtain the secret message during game play and no other player or warden is aware of the secret communication [17].

In this study, steganography in snake game has been proposed. We aim to hide the bits of secret data by estimating the each move of the bait called "vitamin". In other words, we compute the coordinates of the vitamin according to the bits of the secret data. Meanwhile, we plan to complicate the game platform and encrypt the bits of secret data at the same time by employing DES (Data Encryption Standard) S-boxes (substitution boxes). These operations render the proposed scheme more resilient against the possible steganographic attacks and the extraction procedure more complicated. That's why, the security issue has been reinforced in the scope of this study. Since the hiding procedure has been performed via the location of the vitamin, imperceptibility of the game depends on the randomness of the vitamin moves. This is the main reason of complicating the game board by employing s-boxes.

The rest of the paper has been organized as five sections. In section 2, a brief information about the used concepts (information hiding in games, snake game and s-boxes) have been presented. In section 3, the proposed scheme has been explained by both mentioning the embedding and extracting phases. In section 4, experimental results and the analysis of the proposed method in terms of capacity, imperceptibility and security have been provided. Finally, a general outcome has been reached in section 5.

II. Research Methodology

In this section, the employed methodologies in the scope of this study have been mentioned. First, the ways of information hiding in games have been explained to represent the basis of the proposed scheme. Then, the employed game platform, snake, has been expressed. Finally, the s-boxes and their usage have been described.

A. INFORMATION HIDING VIA GAMES

Computer games play a significant part, not only in children's life but in adolescents' culture as well. Researchers indicate that the average time a person spends playing digital games during a regular school day is two hours and a half [18]. We preferred game platform in steganography by considering this wide usage. In this section main information hiding approaches in games are going to be investigated. Information hiding by employing games can be investigated under two approaches like playing a new game from beginning to the end and adding extra information to an already played game.

A.I. Playing a new game

In this approach, both parts of the communication channel should have access to exactly the same software and share a common secret key (used for encrypting the hidden contents) and some other parameters (i.e. the board size, the software version, etc.) to ensure that the software's internal states are reproducible by the two parties.

The main idea is computing for each played position and all the movements which are over a certain threshold value T and then codifying some bits of the hidden message in the selection of the actual played move. Say that, at a given position, there are gm (good moves) moves which are not worse than the given threshold T (analogously, we can fix a certain number n and pick the move within the list of best n moves), then sort them according to their value (by considering the evaluation function) and select the *i-th* move to codify the binary representation of the number *i*. In this way, in each position we will be able to hide around $\log_2$ (gm) bits of information. By increasing T, the channel capacity will grow, since there are more gms to embed the hidden bits. Alternatively, decreasing T will result in a higher invisibility of the hidden contents, for chosen strategies that do not deviate significantly from the optimal.

Other possibilities exist, like using a dynamic T which varies depending on the time or which has a random distribution in order to simulate the game between two players with a good and factual knowledge of openings (i.e. not very bad moves at the beginning of the game but not so strong in the middle or endgame). The recipient can easily decode these moves into binary data by arriving at the same position, re-computing the move list and searching for the code of the actually played move.

A.II. Inserting data into a game

Here, the main idea is basically to follow the mentioned procedure in the subsection 2.1.1. for hiding some bits in every move. But in this method, the difference is that the moves correspond to variants of the main line that is actually played on the game. At certain positions in the game the variants of the main played line are introduced. Each variant could embed bits by various ways like choosing movement where the variant begins, the length of the variant and obviously, in every move of the variant, just by using the same algorithm that is described above for codifying hidden data in moves.

Another possible channel for embedding information is the inclusion of user comments during the game. Generally, this embedding technique enables the sender to hide much more information than the first one. Besides it is harder to detect in most cases. However, it also has the important drawback of a much lower robustness: it suffices to delete all comments to erase the hidden data, but at the cost of severely decreasing the attractiveness and usefulness of the game file. [2]

B. SNAKE GAME

In the snake game, several snakes move in a two-dimensional space, trying to reach an object designed as the *vitamin*. A snake is composed by a special first cell called the *head*, possibly followed by several other cells, known as the *body*. If there is at least one cell in the body, the last cell of the sequence is the *tail*. The movement is controlled via the head, and the other cells follow their respective previous neighbour in the sequence [19]. The snake modelled via cells has been demonstrated in Figure 1.

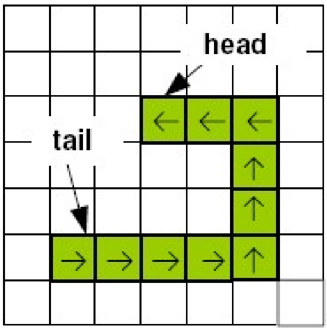


Fig. 1 *The snake model with cells [20]*

When the head of a snake reaches the vitamin, we say that this snake “eats” the vitamin. In this case, the body of that snake grows one cell, and the vitamin appears somewhere else in the space. The goal is to make a snake longer, until reaching a predefined length [19]. In this study, we aim to benefit this specification of the game. Namely, we estimate the movement of the vitamin according to the bits of the secret data (refer to the approach in subsection 2.1.1). Meanwhile, we plan to complicate the game board by considering the s-boxes structure. This operation will render the proposed scheme more resilient against the possible steganographic attacks and increase the desired randomness for the vitamin moves in terms of imperceptibility.

C. S-boxes

S-boxes are commonly used in symmetric cryptosystems. S-boxes are usually the only non-linear components of such a cipher, providing local resistance against powerful methods for analysing its security and linear cryptanalysis. The value for the differential uniformity and the value for the non-linear uniformity of an s-box are commonly used as a measure for its resistance against differential and linear cryptanalysis, respectively. Consequently, a fundamental question of both practical and theoretical interest is what the minimum values are for vsrious classes of mappings (S-boxes). In addition, for the design of S-boxes it is valuable to have algorithms available that efficiently generate mappings for which both these values are ‘small’, i.e., generating S-boxes that provide a good resistance against differential and linear cryptanalysis.[21]

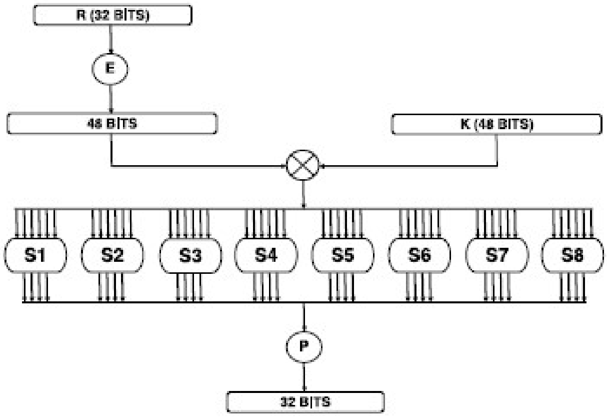
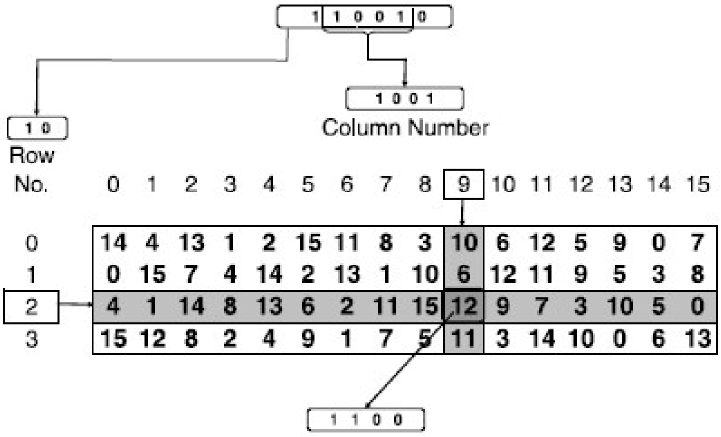


Fig. 2 *Usage of boxes in DES*

Classical S-boxes are tables composed of special configuration of numbers, and serve as tools of non-linear transformation of information in the cipher process. S-boxes are base components of many block ciphers. They operate on fixed-length groups of bit-termed blocks, with use of a fixed transformation. The quality of recently used block ciphers is permanently improved, or substituted by new better constructions. One of well-known application of S-boxes is using them in DES which operates with 8 S-boxes, $S_1 ... S_8$, as illustrated in Figure 2. Each of them is a function expressed as a table composed of 16 columns and 4 rows, and takes a *6-bit* block as an input and yields a *4-bit* block as an output. This transformation has been illustrated in Figure 3. These 8 functions collectively transform the *48-bit* input block into *32-bit* output block.



No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

***Fig. 3** Internal structure of as S-box*

In general speaking, an $n \times k$ S-box is a function $f : B^n \to B^k$ which maps a block-bit strings $(b_0, b_1, . . . , b_n) \to (b_0, b_1, . . . , b_k)$ under condition $k \leq n$ [22]. In this study, we employed S-boxes to provide a non-linear relationship between the input (bits of the secret data) and the output (coordinates of the vitamin in the game) like in DES, but different from circular encryption process.

III. The Proposed Snake Game Steganography Method

In this section, embedding and extracting stages of the proposed scheme have been explained step by step. Meanwhile an illustrative example has been provided for a better understanding.

A. EMBEDDING STAGE

Step 1: Constitute the initialization matrix. This matrix has the size of 6×6. Thus we are able to include 36 different characters:

$$I = \begin{bmatrix} i_{0,0} & \cdots & i_{0,5} \\ \vdots & \ddots & \vdots \\ i_{5,0} & \cdots & i_{5,5} \end{bmatrix} = \begin{bmatrix} a & b & c & d & e & f \\ g & h & i & j & k & l \\ m & n & o & p & q & r \\ s & t & u & v & w & x \\ y & z & 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix} \qquad (1)$$

This matrix has the mission as a global key that is shared between the sender and the recipient beforehand. The structure of *I* matrix is dynamic, namely the location of each character can be in a different cell every time a new secret message is sent. This operation increases the security of the proposed scheme since it makes the extraction procedure more complicated.

Step 2: Constitute 6-bit blocks (*B* array) by encoding each characters of secret message via *I* according to their locations. Let (x, y) denote the line and column number of the secret character in *I*:

$$(x)_2 = b_5 b_4 b_3 \qquad (2)$$

$$(y)_2 = b_2 b_1 b_0 \qquad (3)$$

$$B = \{\forall x, y \in \{0\} \cup Z^+ | b = (x)_2 || (y)_2\}$$

B can be considered as a binary coordinate array whose elements are in the form of $b_5 b_4 b_3 b_2 b_1 b_0$. Here, three MSBs; $b_5 b_4 b_3$ are used to encode the line number, while three LSBs; $b_2 b_1 b_0$ are being used for encoding the column number. Namely, three bits are used for encoding both the line and column numbers. Since $2^3 = 8$, the representation via three bits, totally the block size of six bits, will be enough. Besides, this situation enables us to include 8×8=64 different characters. However, we preferred to include 6×6=36 different characters as a smart beginning.

Step 3: Install the game board. In the proposed scheme, the game board has the size of 8×8. Totally, 64 cells are obtained in order to locate the vitamin in the game:

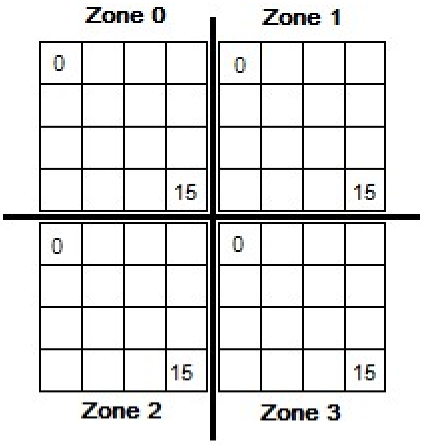


Fig. 4 *Structure of the game board.*

The game board has the structure as illustrated in Figure 4. Size of the game board has been designed to provide the desired randomness for the location of vitamin by considering the output of DES S-boxes. That's why the game board has been separated into four zones each of which consists of 16 cells; from 0 to 15 (Notice that the smallest output of a DES S-box was 0, and the biggest output was 15). Each zone can use the same or different S-boxes. Certainly, the usage of a different S-box for each zone will increases the security (refer to section 4) and imperceptibility since the obtained outputs will be different in case of hiding the same secret pattern.

Step 4: Estimate the coordinates of the vitamin on the game board. As mentioned in Figure 4, the game board consists of four zones. Let $C = (x', y')$ denote the coordinates of the vitamin on the game board. Finally, we aim to find the cell number of the vitamin. This operation is performed as follows:

$$C = \{\forall x', y' \in \{0\} \cup Z^+ | x' < 3,\ y' < 15\}$$

$$x' = (b_5 b_0)_{10} \tag{4}$$

$$y' = (S(b_5 b_4 b_3 b_2 b_1 b_0))_{10} \tag{5}$$

Here, x' denotes the zone number on the game board beginning from 0 (00) to 3 (11). S denotes the S-box transformation. Namely, we narrow the margin by estimating the zone on the game board and then we define the concerning cell in that zone via S-box transformation. At that point we aim to perform a nonlinear transformation between the secret character and the coordinates of the vitamin. As the result, we obtain *C* array that represents the cell sequence of the vitamin on the game board:

$$C = ((x'_0, y'_0), (x'_1, y'_1), \ldots, (x'_n, y'_n))$$

Here, n represents the length of secret message. Notice that each character of the secret message was represented as the location of the vitamin, directly and accordingly the movement of the player, indirectly. However in the practice, snake game continues till the player ends the game with success or not. That is to say, the movement order which is bigger than n is not necessary for extraction. Therefore the length of the secret message is one of the seed components sent to the recipient in a secure way to prevent any complication and to reduce the computational load of extraction. Briefly, we aim to obtain the cell sequence of the vitamin on the game board at the end of embedding stage.

B. EXTRACTING STAGE

Step 1: Get the sequence of the vitamin location by considering the zone (refer to Figure 4). In this step, we aim to obtain *C* array:

$$C = ((x'_0, y'_0), (x'_1, y'_1), \ldots, (x'_n, y'_n))$$

In C array,x' denotes the zone number and y' denotes the cell number of the vitamin in that zone. So initially, the game board is needed to be splitted into four parts indicated as Figure 4. The rest of extraction has to be performed by considering the length of secret message shared only between the sender and the recipient beforehand. In other words, there is no need a notation and computation for all of the vitamin movements.

Step 2: Decode C array to obtain B array that includes the binary coordinates of original characters in matrix *I.* In C array we know that;

$$y' = (S(b_5b_4b_3b_2b_1b_0))_{10}$$

So by considering the given equation (Eq. 5), we represent y' in base 2 and then perform the inverse of S-box transformation to obtain the original bit stream, $b_5b_4b_3b_2b_1b_0$.

$$\forall b \in B = INV\,S((y')_2) \tag{6}$$

Step 3: Mine the characters of secret message from I matrix by employing B array. At this point, we can estimate C array by splitting B as follows (refer to Eq. 2 and 3, if necassary):

$$x = (b_5b_4b_3)_{10}$$

(7)

$$y = (b_2b_1b_0\ )_{10}$$

(8)

Notice that x indicates the line number and y indicates the column number of the secret character in I matrix. By concatenating x and y, and then representing these 6 bits in base 10, we achieve the elements of B that we estimated in the beginning of the embedding phase.

C. Information set for game construction in extraction stage

In the proposed scheme, each character of the secret message is represented as the location of the vitamin, directly and the movement of the player, indirectly. C array has been constructed with this purpose:

$$C = ((x'_0, y'_0), (x'_1, y'_1), \dots, (x'_n, y'_n))$$

Here, we mentioned that n represented the length of secret message. We have n coordinates (accordingly n moves) for the vitamin in order to embed the secret message. However in the practice, snake game continues till the player ends the game with success or not. With this purpose, the rest of the coordinate information has been defined randomly. But this situation makes the extraction procedure nearly unsolvable for the recipient. Therefore the length of the secret message is one of the seed components that must be sent to the recipient in a secure way to prevent any complication and to reduce the computational load of extraction. The recipient continues extraction procedure by considering this information. The moves whose order exceeds the length of secret message is not necessary to handle. It has been mentioned that the game board was splitted into four zones. Allocation of the vitamin to the concerning cell is performed by the selection of zone and employing S-box transformation (S1-S8) to define the concerning cell in that zone:

$$C = \{\forall x', y' \in \{0\} \cup Z^{+} \mid x' = (b_5 b_0)_{10}, y' = (S(b_5 b_4 b_3 b_2 b_1 b_0))_{10}\}$$

The same S-box (S1-S8) can be used for all of the four zones. But to increase the security, a specific S-box for each zone has to be used. In this case, the order of the defined S-boxes for each zone has to be sent to the recipient. Thus, the extraction will be definite for the recipient but indefinite for the attacker.

D. AN ILLUSTRATIVE EXAMPLE

In this sub section, let's hide a short secret message for ease of implementation. Let the secret message be "game".

Step 1: Construction of *I*:

$$I = \begin{bmatrix} a & b & c & d & e & f \\ g & h & i & j & k & l \\ m & n & o & p & q & r \\ s & t & u & v & w & x \\ y & z & 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 & 8 & 9 \end{bmatrix}$$

Step 2: Construction of *B* array by encoding the characters of secret message via *I*, according to their locations. Secret message has been chosen as "game". By using Eq. 2 and 3, *x, y* and *B* array are estimated as follows:

For "g" (1st line, 0th column)

$$b = (x)_2 || (y)_2 = (1)_2 || (0)_2 = 001000$$

For "a" (0th line, 0th column)

$$b = (x)_2 || (y)_2 = (0)_2 || (0)_2 = 000000$$

For "m" (2nd line, 0th column)

$$b = (x)_2 || (y)_2 = (2)_2 || (0)_2 = 010000$$

For "e" (0th line, 4th column)

$$b = (x)_2 || (y)_2 = (0)_2 || (4)_2 = 000100$$

$$B = \{001000, 000000, 010000, 000100\}$$

Step 3: Construction of game board (refer to Figure 4).

Step 4: Estimate the coordinates of the vitamin, *C* array, via Eq.4 and Eq. 5 as follows:

$$C = ((x'_0, y'_0), (x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3))$$

$$x'_0 = (b_5 b_0)_{10} = (00)_{10} = 0\ (Zone\ 0)$$

$$y'_0 = (S_5(b_5 b_4 b_3 b_2 b_1 b_0))_{10} = (S(001000))_{10} = (0111)_{10} = 7\ (cell\ 7\ )$$

$$x'_1 = (b_5 b_0)_{10} = (00)_{10} = 0\ (Zone\ 0)$$

$$y'_1 = (S_5(b_5 b_4 b_3 b_2 b_1 b_0))_{10} = (S(000000))_{10} = (0010)_{10} = 2\ (cell\ 2)$$

$$x'_2 = (b_5 b_0)_{10} = (01)_{10} = 1\ (Zone\ 1)$$

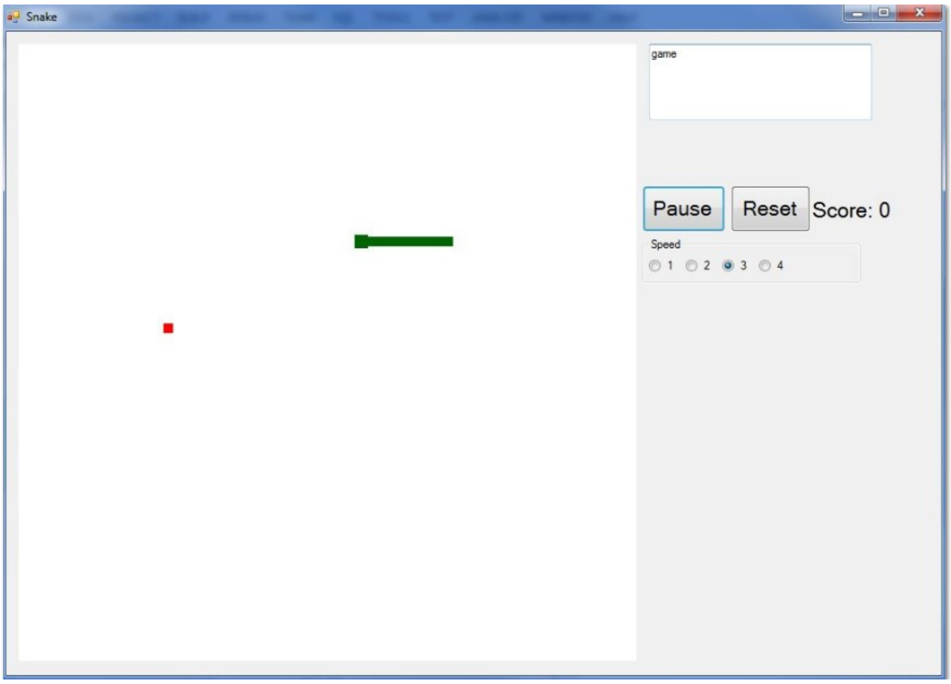
$$y'_2 = (S_5(b_5 b_4 b_3 b_2 b_1 b_0))_{10} = (S(010000))_{10} = (1000)_{10} = 8\ (cell\ 8\ )$$

$$x'_3 = (b_5 b_0)_{10} = (00)_{10} = 0\ (Zone\ 0)$$

$$y'_3 = (S_5(b_5 b_4 b_3 b_2 b_1 b_0))_{10} = (S(000100))_{10} = (0100)_{10} = 4\ (cell\ 4\ )$$

$$C = ((0,7), (0,2), (1,8), (0,4)).$$

Figure 5. *Denotes the game outputs for the given secret message; game.*



Snake

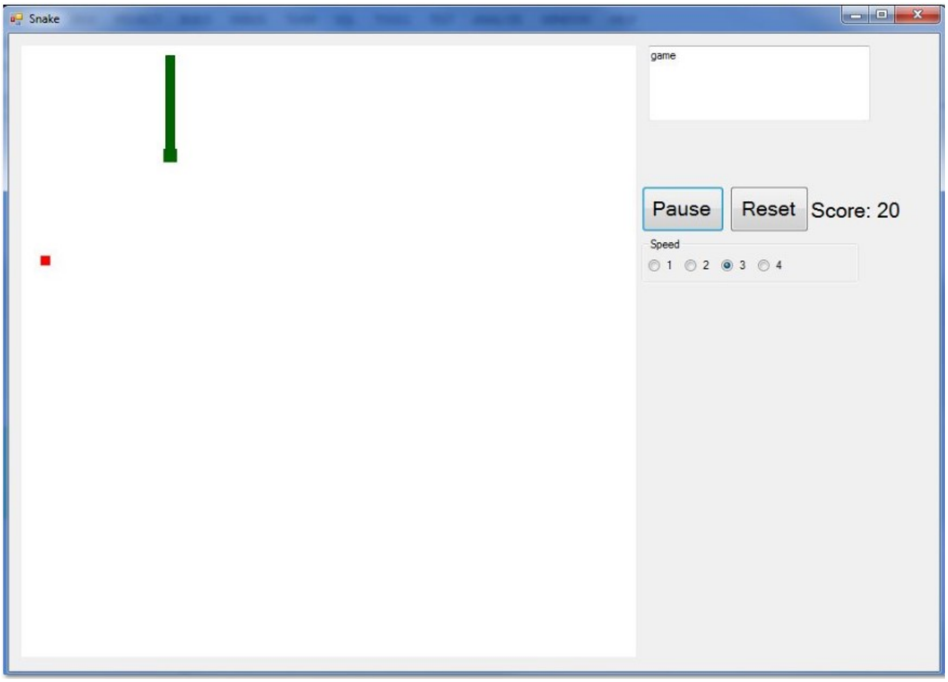
game

Score: 0

Speed

○ 1 ○ 2 ● 3 ○ 4

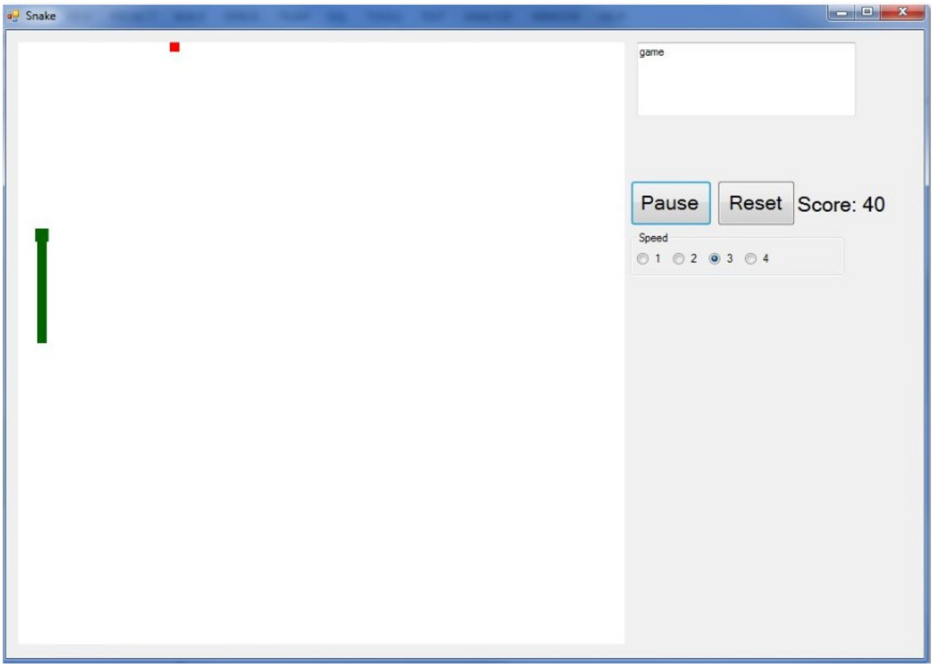
(a) Vitamin location for “g”



Score: 20

Speed: ○ 1 ○ 2 ● 3 ○ 4

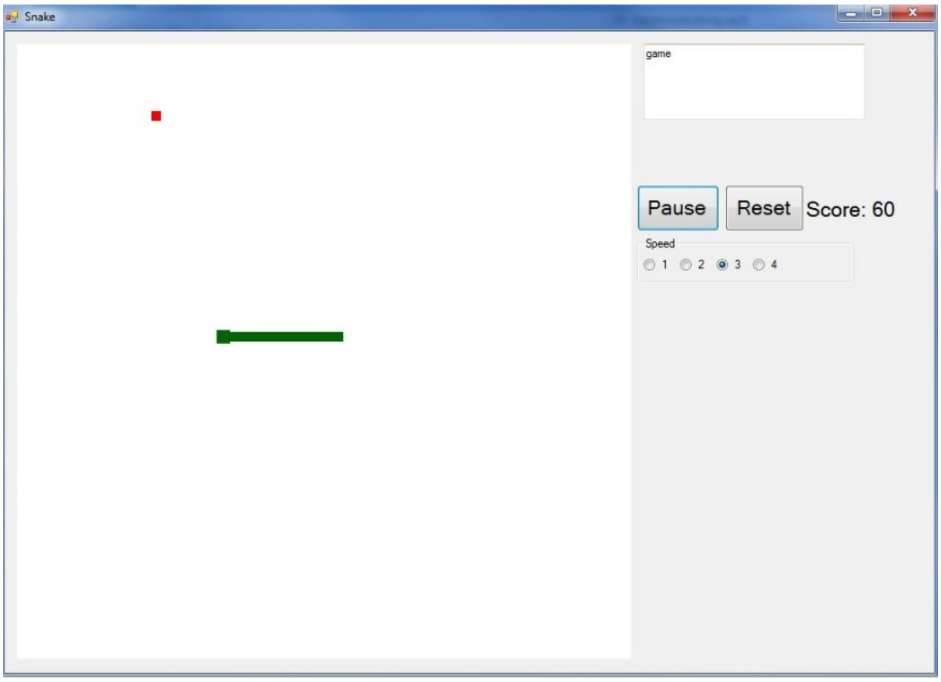
(b) Vitamin location for “a”



Score: 40

Speed: ○ 1 ○ 2 ● 3 ○ 4

(c) Vitamin location for “m”



game

Score: 60

Speed ○ 1 ○ 2 ● 3 ○ 4

(d) Vitamin location for “e”

***Fig. 5 a, b, c, d** Vitamin locations for the given secret message; “game”*

IV. Experimental Results

In this section, we both examine and analyse the proposed method in terms of capacity, imperceptibility and security. The experiments in the scope of this study have been performed by employing the Winstein Lexical Steganography dataset [23].

A. CAPACITY ANALYSIS

In the proposed scheme, location of the vitamin has been estimated according to each character of secret message. That is to say, the move of the player is being guided via the vitamin location. C array has been constructed with this purpose that has the structure as follows:

$$C = ((x'_0, y'_0), (x'_1, y'_1), \dots, (x'_n, y'_n))$$

Here, we mentioned that n represented the length of secret message. We have n coordinates (accordingly n moves) for the vitamin in order to embed the secret message. We hide a character (8 bits) for each move of the vitamin. Table 1 indicates the results of capacity measurements.

Table 1 Capacity measurement details

Secret data(bits)	8	40	80	120	160	200	240	280	320	360	400
Number of moves	1	5	10	15	20	25	30	35	40	45	50
Game size (bits)	21504	21504	21504	21504	21504	21504	21504	21504	21504	21504	21504

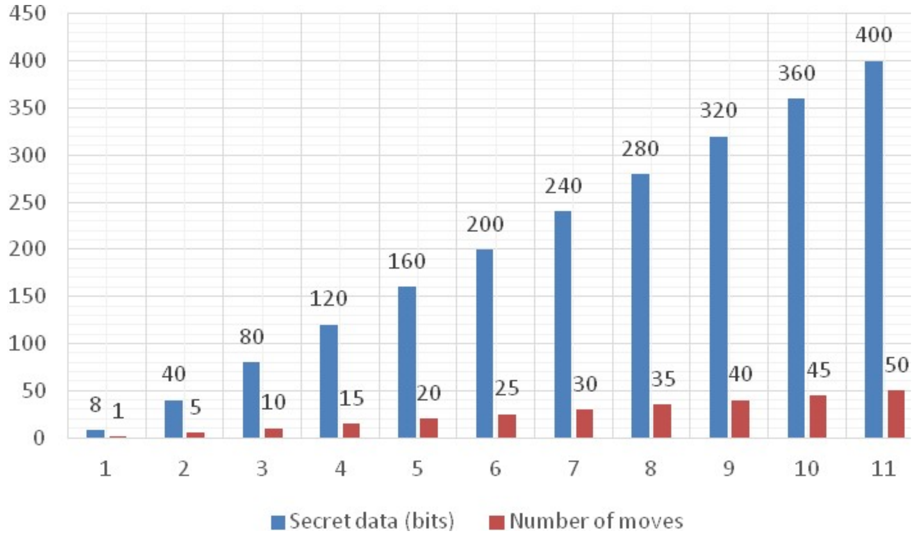


***Fig. 6** Capacity graph indicating the hidden bits with respect to the performed moves*

Figure 6 denotes the capacity graph in terms of the amount of hidden bits vs. the number of performed moves. y axis indicates the amount of hidden bits while x axis is indicating the number of performed experiments. Size of the game is fixed as indicated in Table 1. It can be seen that the amount of hidden bits increased as the number of moves in the game increased. Namely, 8 bits have been embedded per move (8 bpm).

B. IMPEREPTIBILITY ANALYSIS

According to Shannon's theory, information entropy is one of the main randomness measurements of information. High entropy values express a high degree of randomness and for any message coded on m bit, the upper bound of the entropy is m. The formula of entropy has been provided below [24]:

$$H = -\sum_{i=0}^{2m-1} p_i Log_2 p_i \qquad (9)$$

As mentioned in [17], a secure steganography scheme is desired to be statistically undetectable. In this subsection, we investigated the distribution of the vitamin location for an unbiased evaluation. The distribution of the vitamin location has to seem like having no relationship with the secret message and with the length of secret message. That's why, we estimated the entropy values for the given length of secret messages in Table 2 and presented the tendency of entropy in Figure 7.

***Table 2.** Estimated Entropy values*

Character Length	8	40	80	120	160	200	240	280	320	360	400
Entropy	3	4.031	4.453	4.438	4.834	4.823	4.712	4.895	4.934	4.963	4.893

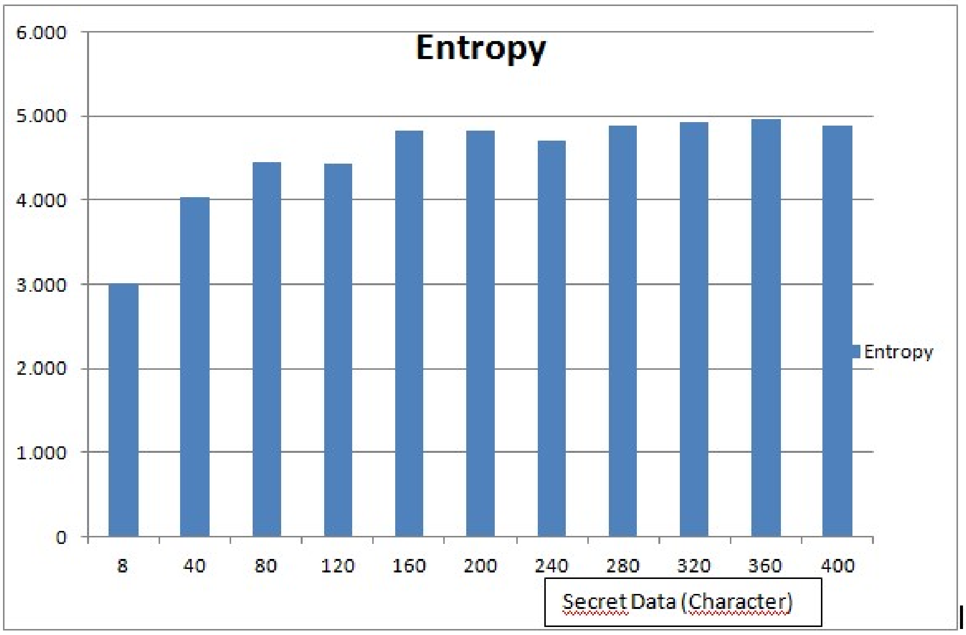


***Fig.** 7 Entropy graph indicating the estimated entropy values with respect to size of secret data*

In table 2, the increment rate in the character length for each secret message is 20 during the experiments. In Figure 7, the increment rate in the character length for each secret message is 10 during the experiments, for a sensitive evaluation. As seen both in Table 2 and Figure 7, the entropy values increased with the size of secret data exceeding 2 which is near 4. This result is significant for us since the secret data is coded by four bits representing the cell number of the vitamin. This means that the relationship between the vitamin location and the secret data has decreased from the beginning to the end. In other words, the current randomness increases as the length of secret data increases. The bias curve on the provided graph shows the tendency of the entropy distribution which has a significant R value.

C. SECURITY ANALYSIS

In steganography, there are two stages to break a steganographic system: First, an attacker has to detect the existence of a secret message in a steganographic system. Second, the attacker has to extract the embedded message [16]. In this study, we initially, focused on these two issues. For the first item; each character of the secret message has been transformed to the location of the vitamin during the game. Thus, we represented the secret message as a coordinate array for the vitamin; an appropriate component to the nature of the game (refer to subsection 3.1, step 4 and for a concrete outcome subsection 3.4, step 4). For the second item, we set a nonlinear relationship between the secret character and the vitamin location via DES S-boxes (refer to subsection 3.1and for a concrete outcome subsection 3.4).

However, there is a strong possibility that the attacker can attempt to extract the secret message by “brute force”. In this case, the attacker must guess correct components to set up the game as in the recipient side (refer to subsection 3.3) and the global key shared between the sender and the recipient beforehand (refer to subsection 3.1, step 1). Namely, the attacker must guess the length of secret message (n) to investigate only the necessary vitamin moves instead of investigating all of the moves. Besides, the attacker must guess the correct S-box order on zones, Z1, Z2, Z3, Z4 (refer to Figure 4), to perform the mining operation on the defined moves without an error and finally the global key; initialization matrix. That is to say, the attacker must guess all of these three components, correctly in order to extract the secret message by means of the vitamin location during the game.

In the proposed scheme, we combined cryptography and steganography to increase the security level as stated in [1]. To embed the secret message via the vitamin location, we employed DES S-boxes. For each zone, four different S-boxes can be used in one game session to increase the security as mentioned in subsection 3.3. With the addition of initialization matrix, an attacker has to perform numerous combinations to extract the secret message. Accordingly, we formulated the number of possible combinations as follows:

$$A = 36! \times 8^4 \times \binom{m}{n} \tag{10}$$

Here, A denotes the combination number of a brute force attack. m denotes the total number of vitamin moves and n denotes the length of secret message (notice that n ≤ m). Namely, an attacker has to perform the same number of combinations expressed by the Equation 10. Hence the probability of the attack is:

$$P = \frac{1}{36! \times 8^4 \times \binom{m}{n}} \tag{11}$$

This is a very small number (nearly 6×10^{-46}) when we consider the worst case where $m=n=1$. By considering these results, we can claim that the proposed scheme is secure.

V. Conclusion

In this study, we proposed a game steganography method using snake game as the cover medium. We concealed secret data by employing the vitamin location. For each played game, every move of the vitamin represents a secret character. However, there is no security and imperceptibility by performing this hiding operation directly. So we employed DES S-boxes to establish a nonlinear relationship between the secret data and the vitamin location. By this way, security and imperceptibility have been provided and 8 bits have been hidden by each move. Besides, different outputs are obtained in case of hiding the same secret message, if a different S-box sequence is used. Namely, the vitamin will be in different locations on the game board.

However, as a criticism, we used 6×4 bits s-boxes in this study. There is a very advantageous opportunity to render the proposed scheme more resilient in terms of imperceptibility and security. The number of cells in the game board and the variety of secret symbols can be increased by employing more complicated S-boxes like in AES (Advanced Encryption Standard). This is the main issue that will be tackled in further investigations.

ACKNOWLEDGEMENTS: This study was not supported by any profit/non-profit organization.

VI. REFERENCES

[1] M.M. Sadek, A.S. Khalifa, M.G.M. Mostafa (2015) **DOI: 10.1007/s11042-014-1952-z.**

[2] 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, Computers & Security **25 (1)** (2006) 64-71.

[3] W. Luo, F. Huang, J. Huang (2011) **DOI: 10.1007/s11042-009-0440-3.**

[4] H. Sajedi, M. Jamzad (2009) **DOI: 10.1007/s10207-009-0089-y.**

[5] K. Bailey, K. Curran (2006) **DOI: 10.1007/s11042-006-0008-4.**

[6] V. Sabeti, S. Samavi, S. Shirani (2013) **DOI: 10.1007/s11042-011-0975-y**

[7] D. Yan, R. Wang (2011) **DOI: 10.1007/s11042-009-0430-5.**

[8] A. Desoky *Int. J. Inf. Secur.* **8 (4)** (2009) 247–261.

[9] E. Satir, H. Isik *J. Syst. Softw.* **85 (10)** (2012) 2385–2394.

[10] E. Satir, H. Isik (2014) **DOI: 10.1007/s11042-012-1223-9.**

[11] W. Mazurczyk, K. Szczypiorski, Evaluation of steganographic methods for oversized IP packets, Telecommun Syst 49 (2) (2012) 207–217. **DOI: 10.1007/s11235-010-9362-7.**

[12] B. Jankowski, W. Mazurczyk, K. Szczypiorski, PadSteg: introducing inter-protocol steganography, Telecommun Syst 52 (2) (2013) 1101–1111. **DOI: 10.1007/s11235-011-9616-z.**

[13] W.C. Yang, L.H. Chen, A steganographic method via various animations in PowerPoint files, Multimed. Tools Appl. 74 (3) (2015) 1003–1019. **DOI: 10.1007/s11042-013-1708-1.**

[14] D. Uljarevic´, M. Veinovic´, G. Kunjadic´, D. Tepšic´ (2015) **DOI: 10.1007/s00530-015-0492-3.**

[15] A. Desoky, M. Younis (2009) **DOI: 10.1002/sec.99.**

[16] L.H. Lee, C.F. Lee, L.H. Chen (2010) **DOI: 10.1016/j.jss.2010.07.054.**

[17] Z.H. Ou, L.H. Chen (2014) **DOI: 10.1016/j.ins.2013.12.024.**

[18] M. Virvou, S. Papadimitriou, *Simple arithmetic lessons through an adaptive snake game,* **In: Proceedings of International Conference on Computer, Information and Telecommunication Systems (CITS 2013)**, (2013) 71-75.

[19] V. Di Iorio, R.S. Bigonha, M.A.S. Bigonha, A. Oliveira, E. Miguel (2005) **DOI: 10.1016/j.entcs.2005.03.008.**

[20] M. Temmerman, E.G. Daylight, F. Catthoor, S. Demeyer, T. Dhaene (2007) **DOI: 10.1016/j.sysarc.2006.11.008.**

[21] P. Roelse (2007) **DOI: 10.1007/s10623-006-9012-y.**

[22] M. Szaban, F. Seredynski (2011) **DOI: 10.1007/s11227-010-0398-y.**

[23] K. Winstein, Lexical steganography, http://alumni.imsa.edu/~keithw/tlex, Accessed 01 April 2016

[24] F.K. Mohamed (2014) **DOI: 10.1016/j.jestch.2014.04.001.**