*Computers, Materials & Continua*
DOI: 10.32604/cmc.2023.035364
***Article***

Tech Science Press

# A New Generative Mathematical Model for Coverless Steganography System Based on Image Generation

**Al-Hussien Seddik[1], Mohammed Salah[2], Gamal Behery[2], Ahmed El-harby[2], Ahmed Ismail Ebada[2], Sokea Teng[3], Yunyoung Nam[3,*] and Mohamed Abouhawwash[4,5]**

[1]Department of Computer Science, Faculty of Science, Minia University, Minia, Egypt
[2]Department of Computer Science, Faculty of Computers and Artificial Intelligence, Damietta University, New Damietta, Egypt
[3]Department of ICT Convergence, Soonchunhyang University, Asan, 31538, Korea
[4]Department of Mathematics, Faculty of Science, Mansoura University, Mansoura, 35516, Egypt
[5]Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, 48824, MI, USA
*Corresponding Author: Yunyoung Nam. Email: ynam@sch.ac.kr

Received: 18 August 2022; Accepted: 03 October 2022

**Abstract:** The ability of any steganography system to correctly retrieve the secret message is the primary criterion for measuring its efficiency. Recently, researchers have tried to generate a new natural image driven from only the secret message bits rather than using a cover to embed the secret message within it; this is called the stego image. This paper proposes a new secured coverless steganography system using a generative mathematical model based on semi Quick Response (QR) code and maze game image generation. This system consists of two components. The first component contains two processes, encryption process, and hiding process. The encryption process encrypts secret message bits in the form of a semi-QR code image whereas the hiding process conceals the pregenerated semi-QR code in the generated maze game image. On the other hand, the second component contains two processes, extraction and decryption, which are responsible for extracting the semi-QR code from the maze game image and then retrieving the original secret message from the extracted semi-QR code image, respectively. The results were obtained using the bit error rate (BER) metric. These results confirmed that the system achieved high hiding capacity, good performance, and a high level of robustness against attackers compared with other coverless steganography methods.

**Keywords:** Coverless steganography; data hiding; information security; QR code; maze game

## 1 Introduction

In the computer and internet era, securing transmitted data is considered a necessary process. Therefore, an effectiveness system must be an alternative for solving this need [1]. Securing the transmission communication channels against any attackers is an important conventional operation [2]. Many methodologies like steganography are used to secure data [3]. The smarter the generated steganography system, the higher its robustness level.

Steganography is a technique used to hide a piece of information in any other type without changing the second information to appear as an original one [4]. Some researchers define steganography as "hiding in plain sight" because the sent message is secretly out in the open for all to see. Some forms of steganography are even done unnoticed from the sender to the message recipient [5].

Generally, steganography consists of two components. The first is called embedding which is responsible for concealing the secret message within a cover file. The second component is called extraction which retrieves the secret message from the sent stego file, the cover file [6].

In 2013, a group of researchers proposed hiding data without using a cover file. This method is called coverless steganography. Coverless steganography can execute by a cover generation using the secret message bits themselves. This generated cover implies the secret message which may be an image, video, audio, or text file. The other form of coverless steganography involves building an image database consisting a number of natural images, which are then divided into subblocks. Finally, the secret message bits are compared for matching with these image subblocks [6].

Tan et al. [7] used motion analysis of videos to develop a coverless steganography technique. Robust histograms of oriented optical flow (RHOOF) were generated for all videos found in the database which was then indexed. These indices and RHOOF hash sequences are transmitted to the message recipient as a mapping. The RHOOF hash sequences were computed from the sent video which helped in retrieving the secret message. All of the videos used as a cover did not lose any of their respective contents during the transmitting and receiving processes.

Saad et al. [8] generated a jigsaw puzzle image with the aid of the cover image to build a new coverless image steganography technique. This approach uses a natural image as a cover image, then splits it into similar subblocks row by row then column by column. The zeros and ones of the secret message bitstream were represented by blanks and tabs for each puzzle piece, respectively. The created jigsaw puzzle image is received as a stego image wherein the recipient has the ability to retrieve the message from the stego using the secret message retrieval algorithm.

The main contribution of this paper is to design a new generative mathematical model for a coverless image steganography system based on semi-QR code images and maze game image generation driven by secret messages. The generated system is able to secure data through any digital communication channel without exposure to attackers during the transmission process.

The paper is arranged as follows: Section 2 details the proposed method. Section 3 introduces the performance evaluation metric. Section 4 discusses the experimental results. Finally, Section 5 summarizes the conclusion.

## 2 The Proposed System

The proposed system seeks to build a robust system that can securely hide a huge amount of data before transmitting these to the receiver. The system consists of two components: (1) encryption and hiding, and (2) extraction and decryption. Each of the respective sender and receiver terminals has one component. At the first terminal, the secret message bits are encrypted in the form of a semi-QR code image. Then, this encrypted image is converted into the maze game image through hiding process. At the second terminal, the QR code image is extracted from the maze game image during the extraction process. Finally, the secret message bits are retrieved by decrypting the extracted QR code. The structure of the proposed system is illustrated in Fig. 1.

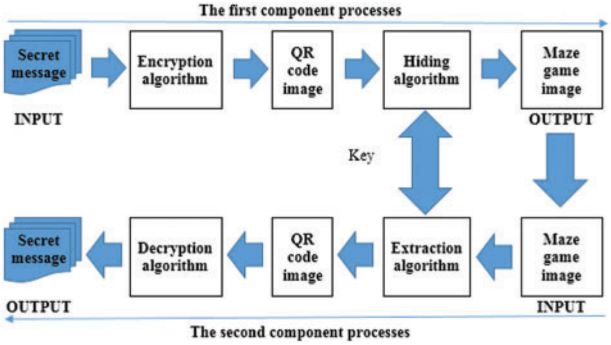


**Figure 1:** The structure of the proposed system

## *2.1 QR Code*

In 1994, the Japanese automotive company, Denso Wave, [9,10] developed the Quick Response (QR) code which is a type of two-dimensional matrix barcode [11,12]. While a barcode is a machine-readable optical label that stores data about an item, a QR code can attach to any data type. In practice, QR codes often contain data such as a tracker, an identifier, or a location that points to a website, an application, or information for a map, respectively. A QR code uses many standardized encoding modes (i.e., numeric, alphanumeric, and byte/binary) to efficiently store data [13]. Recently, QR codes have been used in many industries for their valuable properties such as fast readability and high storage capacity. Other QR code applications include product tracking, item identification, time tracking, document management, and general marketing [14].

The architecture of QR codes consists of black squares arranged together in a square grid on a white background. These squares can be read by any QR reader such as a mobile camera or a scanner. Reed–Solomon error correction is used to proceed with the image interpretation process. Finally, the required data are then extracted from patterns that are present in both the horizontal and vertical components of the image [14].

## *2.2 The First Component*

This component is used at the sender terminal. It consists of two processes: encryption and hiding. During the encryption process, the secret message is transformed into the semi-QR code image. Meanwhile, in the hiding process, the generated semi-QR code is inserted as an input hidden data in the form of a maze game image. These two processes are illustrated in Fig. 1.

### *2.2.1 Encryption Process*

This process plays a vital role in the system by encrypting the secret message into a semi-QR code image. The encryption process presented in Algorithm 1 is as follows: (1) the secret message is inserted into the system and translated into binary (i.e., zeros and ones); (2) a binary image of size 300 × 300 pixels is created and set its pixels to white ones; (3) the QR code is generated using the secret message bits (i.e., if the current secret message bit is zero, set the image pixel to black; else, let it remain white) which is repeated sequentially until all bits of the secret message are represented; (4) the created binary image is fully used except for the regions of the four finder patterns as shown in Fig. 2; (5) finally, the pixel size (PS) and the size of the four finder patterns (FP) are determined from Eqs. (1) and (2), respectively.

**Algorithm 1:** Secret message encryption (Secret message into semi-QR code).

```
START
INPUT: The secret message (SM).
OUTPUT: Semi-QR code image.
Step 1: Create a white binary image of size 300 × 300 pixels.
Step 2: Calculate the pixel size, PS, according to the length of the SM from Eq. (1).
Step 3: Determine the size of the four finder patterns, FP, of the QR code from Eq. (2).
Step 4: Translate the SM into a binary.
Step 5: Add fake bits to the SM bits, as shown in Eq. (3), if the length of the original SM is less than any length mentioned in Eq. (1).
Step 6: For all SM binary bits.
Step 7: If bit == 0 set the current image pixel to black.
Step 8: Else if bit == 1, let the current image pixel be as it is, white.
Step 9: Repeat steps 7 and 8 until all bits of the SM are represented.
Step 10: End if.
Step 11: End.
Step 12: Return the generated QR code image.
END
```

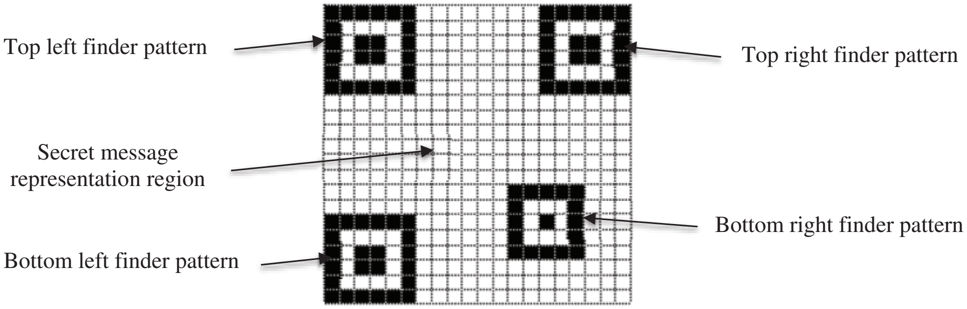

Top left finder pattern
Top right finder pattern
Secret message representation region
Bottom right finder pattern
Bottom left finder pattern

**Figure 2:** Layout of the generated QR code image

$$PS = \begin{cases} W * 0.03 + 1.\,if\ 0 < L \leq 728 \\ W * 0.02.\,if\ 728 < L \leq 2328 \\ W * 0.02 - 1.\,if\ 2328 < L \leq 3432 \\ W * 0.02 * \frac{2}{3}.if\ 3432 < L \leq 5456 \\ W * 0.01.\,if\ 5456 < L \leq 9832 \\ W * \frac{0.02}{3}.if\ 9832 < L \leq 22328 \end{cases} \tag{1}$$

where W is the width of the binary image in pixels and L is the length of the secret message in bits.

$$FP = \begin{cases} PS * 6. \textit{for the first three finder patterns} \\ PS * 5. \textit{for the fourth finder pattern.bottom right} \end{cases} \quad (2)$$

where PS is a matrix of size PS∗PS and its value is calculated as shown above in Eq. (1).

If the real length of the secret message is less than any length mentioned in Eq. (1), then fake bits are added to complete the final length which is represented in QR code. Also, a terminator is used to separate the real bits and fake bits. The final represented bits (FRB) are calculated as follows:

$$FRB = RB + T + FB \quad (3)$$

where RB is the real bit of the secret message, T is a terminator of value "11111111" which is considered an indicator for the end of the real bits, and FB is the fake bit with random binary bitstreams. Fig. 3 illustrates a sample semi-QR code generated by the system with different data lengths and pixel sizes.

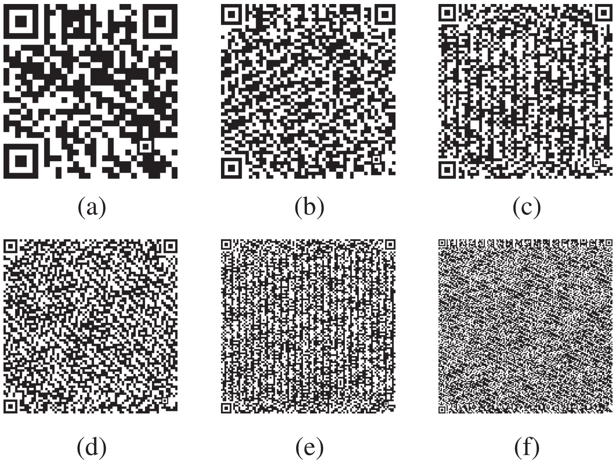

(a) (b) (c)
(d) (e) (f)

**Figure 3:** Samples of the generated semi-QR codes with different pixel sizes and lengths. (a): PS = 10 pixels and L = 728 bits, (b): PS = 6 and L = 2328, (c): PS = 5 and L = 3432, (d): PS = 4 and L = 5456, (e): PS = 3 and L = 9832, and (f): PS = 2 and L = 22328

### *2.2.2 Hiding Process*

In this process, the semi-QR code is transformed into the maze game image. The following mathematical model explains and describes the equations used in the hiding process. These equations determine the number of rows and columns used for generating the maze game image (MSR model).

**The mathematical model for calculating R and C of the maze game image (MSR Model).**

Maze = {String S}; L = length (S).

**Step 1:** According to the length of the semi-QR code image, calculate R and C which refer to the number of rows and columns of the maze, respectively.

**Step 2:** Assume that the generated maze initially consists of a grid with R rows and C = R + 1 columns, for example R = 2 and C = 3, as shown in Fig. 4, as follows:

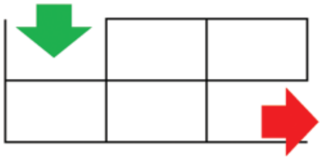


**Figure 4:** A grid with R = 2 rows and C = 3 columns

Note that each row and each column contain a set of vertical and horizontal walls, respectively. Where the green arrow is the entrance of the maze and the red is the exit.

**Step 3:** The maximum number of interior vertical walls, $V = R * (C - 1) = 2 * 2 = 4$ walls and the maximum number of interior horizontal walls, $H = (R - 1) * C = 1 * 3 = 3$ walls.

**Step 4:** The total number of walls should be greater than or equal to the length of the binary bits, L, $V + H \geq L$, this condition should be validated to ensure that the maze can contain all binary bits.

**Step 5:** Substitute for the values of V and H from step 3 to obtain the final value of R and C.

$$R * (C - 1) + (R - 1) * C \geq L \quad \rightarrow \quad RC - R + RC - C \geq L \quad \rightarrow \quad 2RC - R - C \geq L$$

Substitute for $C = R + 1$ as written in the assumption in step 2 to obtain.

$$2R\,(R + 1) - R - (R + 1) \geq L \quad \rightarrow \quad 2R^2 + 2R - R - R - 1 \geq L \quad \rightarrow \quad 2R^2 - 1 \geq L$$

$$\rightarrow 2R^2 \geq L + 1$$

Division by 2 and return the square root to obtain.

$$R^2 \geq \frac{L+1}{2} \quad \rightarrow \quad R \geq \sqrt{\frac{L+1}{2}} \quad \rightarrow \quad \therefore C = R + 1$$

By determining the value of R and C using the MSR model, the maze game image is easy to generate. The hiding process is responsible for transforming the generated semi-QR code image into the maze game image. Algorithm 2 explains hiding the semi-QR code image in the form of a maze game image when the semi-QR code image is inserted as an input: (1) the binary bits which represent the secret message are obtained; (2) after calculating the values of R and C using the previous mathematical model, the MSR model, a grid of size R × C is created; (3) then the binary bits are fed to the grid by scanning it row by row according to the bit value wherein if the bit value equals zero, then remove the current wall, else, let the existence of the current wall remain; (4) the remaining binary bits are represented in the grid column by column; (5) finally, the final form of the grid represents the output maze game image. This image is considered to be the stego image sent to the receiver.

**Algorithm 2:** Hiding process (Semi-QR code image into maze game image).

```
START
INPUT: Semi-QR code image.
OUTPUT: The maze game image.
Step 1: Binary bits are obtained from semi-QR code image.
Step 2: Calculate the values of R and C using an MSR mathematical model.
```

(Continued)

**Algorithm 2:** Continued

```
Step 3: Create a grid of size RxC.
Step 4: For all binary bits do.
Step 5: Scan the grid rows, row by row.
Step 6: If the bit value == 0, then remove the current wall for that row.
Step 7: Elseif the bit value == 1, then let the current wall as it is.
Step 8: End if.
Step 9: Repeat scanning the grid, column by column and apply steps from 6 to 8 until all remaining bits are represented.
Step 10: End.
Step 11: Return the maze game image, the stego image.
END
```

### *2.3 The Second Component*

The receiver terminal uses the second component to retrieve the secret message from the maze game image delivered from the sender. Retrieving the secret message is performed through two processes: extraction and decryption. The extraction process is responsible for extracting the semi-QR code from the maze game image. In the decryption process, the secret message is retrieved from the extracted semi-QR code image.

#### *2.3.1 Extraction Process*

In this process, the semi-QR code image is reobtained from the maze game image. Algorithm 3 describes the steps of semi-QR code reconstruction when the binary bits are obtained from the maze game image: (1) the maze image is scanned row by row to detect all interior vertical walls where each absent wall represents the bit zero and each existing wall represents the bit one; (2) all recovered bits retrieved from the previous scanning are collected together; (3) the maze image is rescanned column by column to detect all interior horizontal walls where the bits are collected as with the previous steps; (4) finally, all collected bits are concatenated, and are used to generate the semi-QR code image which is the input for the decryption process.

**Algorithm 3:** Extraction process (Maze game image into semi-QR code image).

```
START
INPUT: The maze game image.
OUTPUT: Semi-QR code image.
Step 1: The maze image is scanned row by row to detect all the interior vertical walls.
Step 2: Collect all the recovered bits in the Recovered_rows variable.
Step 3: Rescan the maze image column by column to detect all the interior horizontal walls.
Step 4: Collect all the recovered bits in the Recovered_columns variable.
Step 5: Concatenate Recovered_rows and Recovered_columns in the variable Final_bits.
Step 6: Generate the semi-QR code image using the obtained variable Final_bits by applying algorithm 1.
Step 7: Return the semi-QR code image.
END
```

### *2.3.2 Decryption Process*

This process is the final process in the system which is responsible for retrieving the secret message from the semi-QR code image obtained from the extraction process. The decryption algorithm in Algorithm 4 works as follows: (1) the system scans the inserted binary semi-QR code image to determine the size of the FP; (2) calculate the PS to locate the pixel's representation region; (3) check each pixel if its value is zero and return the bit 0, otherwise, return the bit 1; (4) repeat previous step for all pixels, collapse each 8 bit together and translate them into char; (5) finally, concatenate these chars and return the secret message.

**Algorithm 4:** Secret message retrieval (Semi-QR code image into secret message).

```
START
INPUT: Semi-QR code image.
OUTPUT: The secret message (SM).
Step 1: Scan the binary semi-QR code image to determine the size of the finder patterns, FP.
Step 2: Calculate the pixel size, PS, from the value of FP.
Step 3: Initialize the secret message variable (SM) is null.
Step 4: For all image pixels.
Step 5: If the pixel value == 0, then SM += “0.”
Step 6: Elseif the pixel value == 1, then SM += “1.”
Step 7: End if.
Step 8: End.
Step 9: Segment SM into chunks of 8 bits and convert them into characters.
Step 10: Concatenate all the converted characters.
Step 11: Return the secret message (SM).
END
```

## 3 Performance Evaluation Metric

To evaluate the performance of the coverless steganography system, with different lengths of the secret message, the BER measurement has been used. BER measures the ability of the system to retrieve the secret message successfully. It applies the XOR operation between the secret message bits and the retrieved bits to calculate the similarity ratio in bits. BER [15] is given by the following:

$$\text{BER} = \frac{e}{n}, e = \sum\nolimits_{i=1}^{n} pi \oplus qi \tag{4}$$

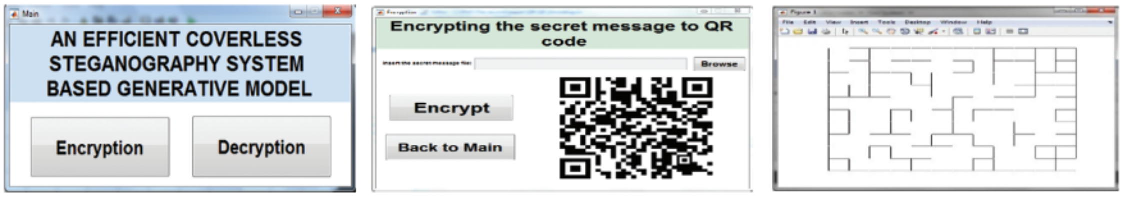
where e is the number of invalid retrieved bits, n is the length of the original secret message, p is the original bitstreams, and q is the retrieved bitstreams. If the value of e equals zero, this means there are no errors (i.e., the original secret message and the retrieved message are identical) and the system escaped from any attack thereby achieving 100% in the robustness ratio. Else, if the value of e is greater than zero, this means that the bits have been altered or lost through the retrieval process and the system is not 100% robust against this attack.

## 4 Experimental Results and Comparisons

This section describes the experiments that have been conducted to study the performance of the proposed coverless steganography system compared with other systems and to check the ability of the system to represent a maximum amount of data in a secure form. MATLAB was used for conducting the results [3,16,17].

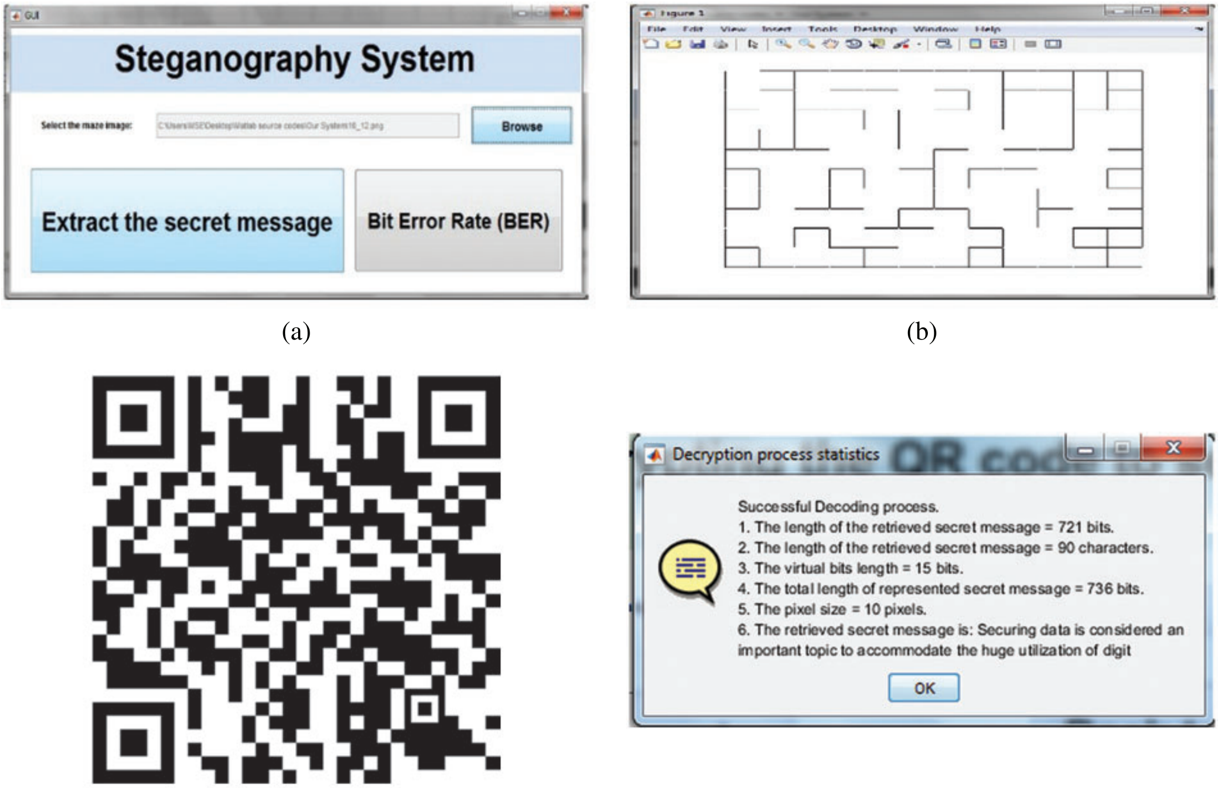
### 4.1 Designed Coverless Steganography System

The proposed system consists of two components wherein each component contains two processes. Fig. 5 shows the interfaces of the two processes of the first component, encryption and hiding, for the system. The two processes of the second component, extraction and decryption, are illustrated with the interfaces in Fig. 6.


Main
AN EFFICIENT COVERLESS STEGANOGRAPHY SYSTEM BASED GENERATIVE MODEL
Encryption
Decryption
Encrypting the secret message to QR code
Browse
Encrypt
Back to Main
File Edit View Insert Tools Desktop Window Help

(a) (b) (c)

**Figure 5:** (a): The main interface for encryption and decryption processes. (b) Generated semi-QR code image. (c) Generated maze game image


GUI
Steganography System
Select the maze image:
Browse
Extract the secret message
Bit Error Rate (BER)
Figure 1
File Edit View Insert Tools Desktop Window Help
Decryption process statistics
OK

```
Successful Decoding process.
1. The length of the retrieved secret message = 721 bits.
2. The length of the retrieved secret message = 90 characters.
3. The virtual bits length = 15 bits.
4. The total length of represented secret message = 736 bits.
5. The pixel size = 10 pixels.
6. The retrieved secret message is: Securing data is considered an
important topic to accommodate the huge utilization of digit
```

(a) (b)

(c) (d)

**Figure 6:** (a): The main interface for the extraction process. (b) The inserted maze game image. (c): Extracted semi-QR code image. (d) The retrieved secret message

### *4.2 Hiding Capacity*

One of the primary goals of the proposed system is to improve hiding capacity. The hiding capacity is considered an important measurement to evaluate the efficiency of any steganography system. This refers to the amount of data that the system can hide successfully [18]. Table 1 compares the hiding capacity of the proposed system with other systems.

**Table 1:** The hiding capacity

| Method | Hiding Capacity (in bits) |
|---|---|
| Li et al. [19] | 8 |
| Cao et al. [20] | 14 |
| Baker et al. [4] | 1∼15 |
| Zhou et al. [21] | 16 |
| Zheng et al. [17] | 18 |
| Tan et al. [7] | 32 |
| Cao et al. [22] | 36 |
| Cao et al. [23] | 68 |
| Zou et al. [24] | 80 |
| Zhou et al. [16] | 384 |
| Saad et al. [8] | 760 |
| Luo et al. [25] | 800 |
| The proposed system | 3432 |

The comparison confirmed that the hiding capacity of the proposed system is the largest among the capacities of the other coverless steganography systems which is 3432 bits. More hiding capacity can be obtained from the system which is up to 22328 bits.

Table 2 shows the number of generated images required to hide different sizes of data. The proposed system uses the smallest number of images compared with other systems which is 2.3 images for hiding 1 kilobyte of data.

**Table 2:** The number of generated images required to hide different sizes of data

| Method | Different data sizes | | | |
|---|---|---|---|---|
| | 8 bits | 80 bits | 800 bits | 8000 bits (1 KB) |
| Zhou et al. [26] | 1 | 10 | 100 | 1024 |
| Yuan et al. [27] | 1 | 10 | 100 | 1024 |
| Zhang et al. [3] | 2∼9 | 7∼81 | 55∼801 | 548∼8193 |
| Zheng et al. [17] | 2 | 6 | 46 | 457 |
| Tan et al. [7] | 1 | 3 | 25 | 256 |
| Saad et al. [8] | 1 | 1 | 1.05 | 10.7 |
| The proposed system | 1 | 1 | 1 | 2.3 |

### *4.3 Robustness*

Robustness can be defined as the ability of a system to resist toward different attacks. This can be measured by evaluating the status of the retrieved message, if retrieved 100% correctly or not. The system was tested, evaluated, and validated through the experimental results driven by different attacks such as image scaling, adding noise, and jpeg compression. Robustness can be measured with the success rate metric (SR) which determines how many bits are retrieved correctly with no errors. SR is calculated based on the BER value defined in Eq. (4), and is defined as the following:

$$\mathrm{SR} = 100\% - \mathrm{BER}\ (\%) \tag{5}$$

The closer the SR value to 100%, the higher the robust level of the system and vice versa. If the value of SR equals to 100% this means that the retrieved message is fully retrieved with no modified or lost bits during the retrieval process. Contrarily, if SR $<$ 100% this is an indication that there are altered bits changed by the attackers.

#### *4.3.1 Image Scaling Attack*

Image scaling (in/out) is considered an effective parameter as an attack due to its ability to modify and destroy the secret message bits [28] during retrieval through the transmission channel. SR values are calculated at different image scaling ratios from 0.3 to 10 as shown in Table 3.

**Table 3:** SR values for image scaling attack

| Image scaling ratio | Zhang et al. [3] | Wu et al. [15] | The proposed system |
|---|---|---|---|
| 0.3 | 85.4% | 98.5% | Failed |
| 0.5 | 94.3% | 99.1% | 100% |
| 0.75 | 96.1% | 99.8% | 100% |
| 1.5 | 98.4% | 97.5% | 100% |
| 2.0 | – | – | 100% |
| 3.0 | – | – | 100% |
| 4.0 | – | – | 100% |
| 5.0 | – | – | 100% |
| 6.0 | – | – | 100% |
| 7.0 | – | – | 100% |
| 8.0 | – | – | 100% |
| 9.0 | – | – | 100% |
| 10.0 | – | – | 100% |

The results confirmed that the system achieved a 100% SR for all scaling ratios, except at ratio 0.3, which means that the system retrieves the secret message correctly with no altered or damaged bits. At scaling ratio 0.3, the system fails to retrieve the secret data in a fully correct form.

#### *4.3.2 JPEG Image Compression Attack*

Almost all transmission communication channels, such as Facebook, WhatsApp, and Yahoo compress sent images [29]. This compression may effect on the transmitted images and damage the data in which this image represents. Jpeg compression is performed at different image qualities, 90%, 80%, 70%, 60%, and 50%, as shown in Table 4, which compares the SR value of the proposed system with the other systems. The format of the original image file was .PNG and its size was 150 KB. The compressed image file sizes were 95, 80, 55, 33, and 20 KB, corresponding to the image qualities mentioned above.

**Table 4:** SR values for JPEG image compression attack

| Image qualities | Zhang et al. [3] | Zhang et al. [3] | Zhang et al. [3] | Zheng et al. [17] | Wu et al. [15] | The proposed system |
|---|---|---|---|---|---|---|
| 90% | 97.8% | 95.2% | 99.8% | 100% | 100% | 100% |
| 80% | – | – | – | – | – | 100% |
| 70% | 96.2% | 92% | 99.1% | 92% | 99.8% | 100% |
| 60% | – | – | – | – | – | 100% |
| 50% | 84.9% | 90.2% | 85.4% | – | 99.3% | 100% |

This comparison showed that the system achieved 100% as a SR for all image qualities, this means that the system has not been affected by the JPEG compression attack.

### *4.3.3 Noise Attack*

Adding noise to any image affects its quality and may corrupt the data that the image represents due to this attack alters the values of a random set of the image pixels. For example, if the image type is a binary image, then adding noise inverses the pixel value (i.e., a black pixel is flipped into a white pixel and vice versa). The number of changed pixels is determined according to the noise density, which represents a portion of the image size that the noise is added. These black/white dots can be easily detected by human eyes. The system robustness is evaluated by applying a "salt and pepper" noise attack [30] at different densities varying from 0.01 to 0.09, see Table 5.

**Table 5:** SR values for "salt and pepper" noise attack

| Noise density | Cao et al. [23] | Zhou et al. [21] | Wu et al. [15] | The proposed system |
|---|---|---|---|---|
| 0.01 | 98% | 99% | 100% | 100% |
| 0.02 | 94% | 96% | 100% | 100% |
| 0.03 | 89% | 95% | 100% | 100% |
| 0.04 | 84% | 91% | 99.95% | 100% |
| 0.05 | – | – | – | 100% |
| 0.06 | – | – | – | 100% |
| 0.07 | – | – | – | 100% |
| 0.08 | – | – | – | 100% |
| 0.09 | – | – | – | 100% |

As shown in Table 5, the results showed that the proposed system is the best one among all compared systems. All SR values were 100% which mean that the retrieved message is matched to the original secret message with zero BER value. Salt and pepper noise attack, using different densities, does not penetrate the system to achieve 100% robustness.

#### *4.3.4 Other Different Attacks*

This section studies the strength of the system by applying many different attacks such as communication channel attacks [31] during sending and receiving messages, image format conversion, and color space conversion. Table 6 shows the SR values for all these attacks.

**Table 6:** SR values for many different attacks

| Different attacks | | The proposed system |
|---|---|---|
| Facebook communication channel attack | Sending & receiving | 100% |
| WhatsApp communication channel attack | Sending & receiving | 100% |
| Yahoo communication channel attack | Sending & receiving | 100% |
| Color space conversion | Binary | 100% |
| | Grayscale | 100% |
| Image format conversion (PNG) | BMP (24 bits) | 100% |
| | JPG | 100% |
| | TIFF (32 bits) | 100% |
| | GIF (8 bits) | 100% |
| | 256 color bitmap (8 bits) | 100% |

As shown in Table 6, the system resists all attacks with 100% SR. This is an indication that the system can protect the retrieved message without no damaged or lost bits.

## 5 Conclusions

This paper presented two algorithms for hiding and extracting secret messages. Building any coverless steganography system faces challenges in hiding capacity limitation and the robustness resisting level. Therefore, a new efficient and effective coverless steganography system based on a generative mathematical model and the creation of semi-QR code and maze game images has been designed. The novelty of this system is to generate the Stego image from only the secret message without using any assistant factors. The proposed system works as follows: at the sender terminal, the secret message is fed to the system generating an encrypted semi-QR code image using the encryption algorithm, where the image is inserted as an input to the hiding algorithm to obtain the maze game image which is considered to be the stego image sent to the receiver. On the other hand, the receiver inserts the received maze game image into the extraction algorithm to reobtain the semi-QR code image then the secret message is retrieved from that image using the decryption algorithm.

To evaluate the effectiveness of this system, results have been carried out using two different evaluation measures, namely the bit error rate and SR. When using either the proposed hiding or extraction algorithms, the experimental results showed that the proposed system achieved satisfactory results conclusion, 3432 bits, and 2.3 images were the hiding capacity and the number of images required to hide different sizes of data, respectively. Also, the proposed system can hide and retrieve up to 22328 bits successfully.

In addition, a comparison has been made between the proposed coverless steganography system and other systems using the BER and SR measures. This comparison showed that the two proposed algorithms gave better success rates than the others, with lower BER values; and higher SR, as introduced in Tables 1–5. This indicates that the proposed coverless steganography system has solved almost all problems in other traditional steganography systems.

Finally, in future work, more OSNs [32–35] can be used as communication channels used for transmitting the secret message. Also, image retrieval [36,37] can play a vital role in generating the new natural images used as Stego images for coverless steganography. Different data types, such as text audio, and videos [38–55], may be used for generating many robust coverless steganography systems.

**Funding Statement:** This work was supported by the Korea Technology and Information Promotion Agency (TIPA) for SMEs grant funded by the Korea government (Ministry of SMEs and Startups) (No.S3271954), and the National Research Foundation of Korea (NRF) grant funded by the korea government (MSIT) (No. 2022H1D8A3038040), and the Soonchunhyang University Research Fund.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] H. H. Liu and C. M. Lee, "High-capacity reversible image steganography based on pixel value ordering," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, pp. 1–15, 2019.

[2] D. Stanescu, M. Stratulat, R. Negrea and I. Ghergulescu, "Cover processing-based steganographic model with improved security," *Acta Polytechnica Hungarica*, vol. 16, no. 1, pp. 227–246, 2019.

[3] X. Zhang, F. Peng and M. Long, "Robust coverless image steganography based on DCT and LDA topic classification," *IEEE Transactions on Multimedia*, vol. 20, no. 12, pp. 3223–3238, 2018.

[4] S. A. Baker and A. S. Nori, "Steganography in mobile phone over bluetooth," *International Journal of Information Technology and Business Management (JITBM)*, vol. 16, no. 1, pp. 111–117, 2013.

[5] J. Long, J. Wiles, R. Rogers, P. Drake, R. J. Green *et al.,* "Techno security's guide to managing risks for IT managers, auditors, and investigators," *Elsevier*, vol. 12, no. 2, pp. 1–22, 2007.

[6] S. Deepa and R. Umarani, "A study on digital image steganography," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 1, pp. 54–57, 2013.

[7] Y. Tan, J. Qin, X. Xiang, C. Zhang and Z. Wang, "Coverless steganography based on motion analysis of video," *Security and Communication Networks*, vol. 2021, no. 6, pp. 1–16, 2021.

[8] A. H. S. Saad, M. S. Mohamed and E. H. Hafez, "Coverless image steganography based on jigsaw puzzle image generation," *Computers, Materials & Continua*, vol. 67, no. 2, pp. 2077–2091, 2021.

[9] J. H. Chang, "An introduction to using QR codes in scholarly journals," *Science Editing*, vol. 1, no. 2, pp. 113–117, 2014.

[10] J. C. Cano, P. Manzoni and C. K. Toh, "Ubiqmuseum: A bluetooth and java based context-aware system for ubiquitous computing," *Wireless Personal Communications*, vol. 38, no. 2, pp. 187–202, 2006.

[11] S. H. Hung, C. Y. Yao, C. Yuan, Y. J. Fang, P. Tan *et al.,* "Micrography QR codes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 9, pp. 2834–2847, 2020.

[12] R. Chen, Y. Yu, X. Xu, L. Wang, H. Zhao *et al.,* "Adaptive binarization of QR code images for fast automatic sorting in warehouse systems," *Sensors*, vol. 19, no. 24, pp. 5466, 2019.

[13] M. Arulprakash, A. Kamal and A. Manisha, "QR-code scanner based vehicle sharing," *ARPN Journal of Engineering and Applied Sciences*, vol. 13, no. 10, pp. 3441–3448, 2018.

[14] M. Johnson and R. Dhanalakshmi, "Predictive analysis based efficient routing of smart garbage bins for effective waste management," *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 5733–5739, 2019.

[15] J. Wu, Y. Liu, Z. Dai, Z. Kang, Z. Rahbar *et al.,* “A coverless information hiding algorithm based on grayscale gradient co-occurrence matrix,” *IETE Technical Review*, vol. 35, no. sup1, pp. 23–33, 2018.

[16] Z. Zhou, Y. Mu and Q. M. Wu, “Coverless image steganography using partial-duplicate image retrieval,” *Soft Computing*, vol. 23, no. 13, pp. 4927–4938, 2019.

[17] S. Zheng, L. Wang, B. Ling and D. Hu, “Coverless information hiding based on robust image hashing,” *International Conference on Intelligent Computing*, vol. 10363, no. 5, pp. 536–547, 2017.

[18] R. Wazirali, W. Alasmary, M. M. Mahmoud and A. Alhindi, “An optimized steganography hiding capacity and imperceptibly using genetic algorithms,” *IEEE Access*, vol. 7, no. 6, pp. 133496–133508, 2019.

[19] S. Li, X. Chen, Z. Wang, Z. Qian and X. Zhang, “Data hiding in iris image for privacy protection,” *IETE Technical Review*, vol. 35, no. 1, pp. 34–41, 2018.

[20] Y. Cao, Z. Zhou, Q. M. J. Wu and C. Yuan, “Coverless information hiding based on the generation of anime characters,” *EURASIP Journal on Image and Video Processing*, vol. 36, no. 1, pp. 1–15, 2020.

[21] Z. L. Zhou, Y. Cao and X. M. Sun, “Coverless information hiding based on bag-of-words model of image,” *Journal of Applied Sciences*, vol. 34, no. 5, pp. 527–536, 2016.

[22] Y. Cao, Z. Zhou, X. Sun and C. Gao, “Coverless information hiding based on the molecular structure images of material,” *Computers, Materials & Continua*, vol. 54, no. 2, pp. 197–207, 2018.

[23] Y. Cao, Z. Zhou, C. Yang and X. Sun, “Dynamic content selection framework applied to coverless information hiding,” *Journal of Internet Technology*, vol. 19, no. 4, pp. 1179–1185, 2018.

[24] L. Zou, J. Sun, M. Gao, W. Wan and B. B. Gupta, “A novel coverless information hiding method based on the average pixel value of the sub-images,” *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 7965–7980, 2019.

[25] Y. Luo, J. Qin, X. Xiang, Y. Tan, Q. Liu *et al.,* “Coverless real-time image information hiding based on image block matching and dense convolutional network,” *Journal of Real-Time Image Processing*, vol. 17, no. 3, pp. 1–11, 2020.

[26] Z. Zhou, H. Sun, R. Harit, X. Chen and X. Sun, “Coverless image steganography without embedding,” in *Int. Conf. on Computational Science*, Nanjing, China, Springer, vol. 1, no. 6, pp. 123–132, 2015.

[27] C. Yuan, Z. Xia and S. Xingming, “Coverless image steganography based on SIFT and BOF,” *Journal of Internet Technology*, vol. 18, no. 2, pp. 435–442, 2017.

[28] Y. Zhang, X. Luo, Y. Guo, C. Qin and F. Liu, “Zernike moment-based spatial image steganography resisting scaling attack and statistic detection,” *IEEE Access*, vol. 7, no. 2019, pp. 24282–24289, 2019.

[29] B. S. Edhah, D. M. Alghazzawi and L. Cheng, “Secret communication on facebook using image steganography: Experimental study,” *International Journal of Computer Science and Information Security*, vol. 14, no. 10, pp. 428–432, 2016.

[30] A. K. Sahu and G. Swain, “A novel n-rightmost bit replacement image steganography technique,” *3D Research*, vol. 10, no. 1, pp. 1–18, 2019.

[31] G. Xie, J. Ren, S. Marshall, H. Zhao and H. Li, “A new cost function for spatial image steganography based on 2D-SSA and WMF,” *IEEE Access*, vol. 9, no. 2021, pp. 30604–30614, 2021.

[32] A. Omar, T. M. Mahmoud, T. A. Hafeez and A. Mahfouz, “Multi-label arabic text classification in online social networks,” *Journal of Information Systems*, vol. 100, no. 4, pp. 101785, 2021.

[33] A. Omar, T. M. Mahmoud and T. Abd-El-Hafeez, “Building online social network dataset for arabic text classification,” in *Int. Conf. on Advanced Machine Learning Technologies and Applications (AMLTA2018). AMLTA 2018. Advances in Intelligent Systems and Computing*, Cham, Cairo, Egypt, Springer, vol. 723, no. 5, pp. 486–495, 2018.

[34] A. Omar, T. M. Mahmoud and T. Abd-El-Hafeez, “Comparative performance of machine learning and deep learning algorithms for arabic hate speech detection in OSNs,” in *Int. Conf. of Artificial Intelligence and Computer Vision (AICV2020). AICV 2020. Advances in Intelligent Systems and Computing*, Cham, Cairo, Egypt, Springer, vol. 1153, no. 4, pp. 247–257, 2018.

[35] T. M. Mahmoud, T. Abd-El-Hafeez and A. Omar, “A highly efficient content based approach to filter pornography websites,” *International Journal of Computer Vision and Image Processing (IJCVIP)*, vol. 2, no. 1, pp. 75–90, 2012.

[36] M. R. Girgis and M. S. Reda, "A study of the effect of color quantization schemes for different color spaces on content-based image retrieval," *International Journal of Computer Applications*, vol. 96, no. 12, pp. 1–8, 2014.

[37] M. R. Girgis and M. S. Reda, "Content-based image retrieval using image partitioning with color histogram and wavelet-based color histogram of the image," *International Journal of Computer Applications*, vol. 104, no. 3, pp. 17–24, 2014.

[38] M. Abouhawwash and K. Deb, "Karush-kuhn-tucker proximity measure for multi-objective optimization based on numerical gradients," in *Proc. of the 2016 on Genetic and Evolutionary Computation Conf. Companion, ACM*, Denver, Colorado, USA, pp. 525–532, 2016.

[39] A. H. El-Bassiouny, M. Abouhawwash and H. S. Shahen, "New generalized extreme value distribution and its bivariate extension," *International Journal of Computer Applications*, vol. 173, no. 3, pp. 1–10, 2017.

[40] A. H. El-Bassiouny, M. Abouhawwash and H. S. Shahen, "Inverted exponentiated gamma and its bivariate extension," *International Journal of Computer Application*, vol. 3, no. 8, pp. 13–39, 2018.

[41] A. H. El-Bassiouny, H. S. Shahen and M. Abouhawwash, "A new bivariate modified weibull distribution and its extended distribution," *Journal of Statistics Applications and Probability*, vol. 7, no. 2, pp. 217–231, 2018.

[42] M. Abouhawwash and M. A. Jameel, "KKT proximity measure versus augmented achievement scalarization function," *International Journal of Computer Applications*, vol. 182, no. 24, pp. 1–7, 2018.

[43] H. S. Shahen, A. H. El-Bassiouny and M. Abouhawwash, "Bivariate exponentiated modified weibull distribution," *Journal of Statistics Applications and Probability*, vol. 8, no. 1, pp. 27–39, 2019.

[44] M. Abouhawwash and M. A. Jameel, "Evolutionary multi-objective optimization using benson's karush-kuhn-tucker proximity measure," in *Int. Conf. on Evolutionary Multi-Criterion Optimization*, East Lansing, Michigan, USA, Springer, pp. 27–38, 2019.

[45] M. Abouhawwash, M. A. Jameel and K. Deb, "A smooth proximity measure for optimality in multi-objective optimization using benson's method," *Computers and Operations Research*, vol. 117, no. 2, pp. 104900, 2020.

[46] M. Abouhawwash, K. Deb and A. Alessio, "Exploration of multi-objective optimization with genetic algorithms for PET image reconstruction," *Journal of Nuclear Medicine*, vol. 61, no. 1, pp. 572, 2020.

[47] B. Gomathi, S. Balaji, V. K. Kumar, M. Abouhawwash, S. Aljahdali *et al.,* "Multi-objective optimization of energy aware virtual machine placement in cloud data center," *Intelligent Automation & Soft Computing*, vol. 33, no. 3, pp. 1771–1785, 2022.

[48] M. Kumar, K. Venkatachalam, M. Masud and M. Abouhawwash, "Novel dynamic scaling algorithm for energy efficient cloud computing," *Intelligent Automation & Soft Computing*, vol. 33, no. 3, pp. 1547–1559, 2022.

[49] R. S. Ram, K. Venkatachalam, M. Masud and M. Abouhawwash, "Air pollution prediction using dual graph convolution LSTM technique," *Intelligent Automation & Soft Computing*, vol. 33, no. 3, pp. 1639–1652, 2022.

[50] A. J. Basha, N. Rajkumar, M. A. AlZain, M. Masud and M. Abouhawwash, "Fog-based self-sovereign identity with RSA in securing IoMT data," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp. 1693–1706, 2022.

[51] G. Ravikumar, K. Venkatachalam, M. A. AlZain, M. Masud and M. Abouhawwash, "Neural cryptography with fog computing network for health monitoring using IoMT," *Computer Systems Science and Engineering*, vol. 44, no. 1, pp. 945–959, 2023.

[52] R. Rajdevi, K. Venkatachalam, M. Masud, M. A. AlZain and M. Abouhawwash, "Proof of activity protocol for IoMT data security," *Computer Systems Science and Engineering*, vol. 44, no. 1, pp. 339–350, 2023.

[53] G. Ravikumar, K. Venkatachalam, M. Masud and M. Abouhawwash, “Cost efficient scheduling using smart contract cognizant ethereum for IoMT,” *Intelligent Automation & Soft Computing*, vol. 33, no. 2, pp. 865–877, 2022.

[54] R. Li, J. Qin, Y. Tan and N. N. Xiong, “Coverless video steganography based on frame sequence perceptual distance mapping,” *CMC-Computers Materials & Continua*, vol. 73, no. 1, pp. 1571–1583, 2022.

[55] R. J. Mstafa, “Reversible video steganography using quick response codes and modified elgamal cryptosystem,” *CMC-Computers Materials & Continua*, vol. 72, no. 2, pp. 3349–3368, 2022.