Copyright ⓒ 2020 KSII

UN-Substituted Video Steganography

Khulood Abu Maria[1*], Mohammad A. Alia[2], Maher A. Alsarayreh[3] and Eman Abu Maria[4]
[1234] Faculty of Science and Information Technology
Al-Zaytoonah University of Jordan
P.O. Box 130, Amman (11733) Jordan
[1] [khulood@zuj.edu.jo]
[2] [dr.m.alia@zuj.edu.jo]
[3] [maher_2007@yahoo.com]
[4] [eman.maria@zuj.edu.jo]
*Corresponding author: Khulood Abu Maria

Received February 19, 2019; revised April 21, 2019; accepted September 8, 2019; published January 31, 2020

Abstract

Steganography is the art of concealing the existence of a secret data in a non-secret digital carrier called cover media. While the image of steganography methods is extensively researched, studies on other cover files remain limited. Videos are promising research items for steganography primitives. This study presents an improved approach to video steganography. The improvement is achieved by allowing senders and receivers exchanging secret data without embedding the hidden data in the cover file as in traditional steganography methods. The method is based mainly on searching for exact matches between the secret text and the video frames RGB channel pixel values. Accordingly, a random key-dependent data is generated, and Elliptic Curve Public Key Cryptography is used. The proposed method has an unlimited embedding capacity. The results show that the improved method is secure against traditional steganography attacks since the cover file has no embedded data. Compared to other existing Steganography video systems, the proposed system shows that the method proposed is unlimited in its embedding capacity, system invisibility, and robustness. The system achieves high precision for data recovery in the receiver. The performance of the proposed method is found to be acceptable across different sizes of video files.

Keywords: Information Hiding, Stego-Key, Exact Matching Algorithm, and Key-Dependent Data Technique.

1. Introduction

In our everyday routine, ensuring the confidentiality of digital information is essential. As the use of the Internet grows, secret information is also required to be secured [1]. Exchanging sensitive data without intruders ' protection is fatal. Consequently, the transfer of sensitive data or secrets must not depend on existing communication channel protection technologies [2][3]. Additional data protection steps should be taken. Military and industrial sectors are looking for a robust technique to protect their data from intrusion or modification. Two methods are used to protect data, cryptography, and steganography. Cryptography is utilized for encoding data in view of some mathematical equations. It is extensively used to make sure that the data is being traded over the Internet, where it is susceptible to attacks leading to detecting the information quickly. The secret information in cryptography is secured using two common keys, the public key and the private key which is unreadable [4]. Attackers can make the data legible by breaking the keys and using their mathematical calculations. Another technique is therefore needed to ensure secure data trading over open channels without any suspicions arising [5].

Steganography is a method based on the secretiveness of a message. Secret information loaded into the digital file is entirely different from the conception of encryption [6][7][8]. Both sender and recipient can agree on a stego key on the extraction stage. Secure steganography approaches should meet specific characteristics, including imperceptibility/invisibility, payload/capacity, and robustness [9]. Imperceptibility implies a perception and a statistical detection of the hidden message knowing that the aim is to keep a strategic distance from any suspected attacker. The quality of the cover files should therefore not be altered, distorted, or delivered between the embedding processes [10]. The capabilities are the measurement of secret information which is concealed in the cover medium without significant statistical or visual changes [11]. The capacity is expressed in bits per pixel, while the percentage is expressed as the maximum hidden capacity. Lastly, image manipulation robustness is considered a significant problem, as the stego medium traverses a communication channel up to its destination. Meanwhile, intentionally or unintentionally an image manipulation such as cropping, resize, and rotation could happen. The incorporation algorithm or method is robust to prevent deliberate or unintended manipulation [12].

An enhanced video steganography method is proposed in this paper. As in other traditional techniques, our approach does not include secret data in the cover file. The entire Exact Matches (EM) between video frame pixel RGB values and secret text is instead detected by the method. Random key-dependent data is generated and used to recover the secret text. The rest of this paper, however, is organized as follows: Section 2 discusses the related work. The proposed method is set out in Section 3. Section 4 discusses the results, while Section 5 shows the conclusion.

2. Related Works

This section presents particular works carried out by previous researches to understand the approaches involved. Based on the non-uniform rectangular partition, [13] proposes a video steganography method. The way is to hide the secret video into a host video stream with almost the same size. Each frame of the secret video is a non-uniform rectangular partition. The obtained partitioned codes are encrypted versions of the original frame. These codes are

hidden in the least four significant bits of each frame of the hosting video. The results of the experiment show that the algorithm can hide the video of the same size without distortion in the hosting video. The DE-interlacing between the video and the RE-interlacing of the video will have an impact on video quality. Modifying the frame pixels will change the quality of the video by breaking the imperceptibility/invisibility and robustness characteristics.

In [14], a video/image steganography technique is presented. The method encrypts the secret message earlier the embedding process. The video file DE-interlaces and converts the images to Joint Photographic Experts Group (JPEG), where each image is divided into blocks of size (3 by 3) pixels. The purpose of searching for each block is to find the minimum pixel value in each block (Pmin). The technique modifies the block pixels' value by finding the distance (d) between (Pmin), and each pixel value (P) at the same block, by subtracting (Pmin) from each pixel at the block (d = P – Pmin). The small values are obtained from the adjacent pixels that have closer values. The technique finds the largest value of (d) for each block (dmax).

Consequently, a bit by bit insertion is carried out and swaps (N) with the rightmost bits of (N) leftmost places. Upon completing the embedding process, the entire image blocks are grouped to form the image back and interlace the images again. Results show that the quality of the stego-video file and the image are low.

Video steganography randomization and parallelization are methods of steganography introduced by [15]. This process encrypts the data in randomly selected frames before the embedding process with a specific key. The Feedback Shift Register (FSR) uses a pseudo-random number generator to eliminate any repeated frames in the embedding process. Data encryption is done by splitting it into equal parts (m) and then XOR'ing with the secret key. Consequently, the result value is hidden by the Least Significant Bit (LSB) method in the randomly selected frames. In the extraction process, the pseudo-random number generator and the FSR identify the selected frames of the embedding process. The secret data is extracted and decrypted accordingly. However, as the method modifies the pixel values using the LSB method, this method is subject to steganography attacks. This method also does not clarify the location of the pixels used in the process of embedding.

A hybrid approach of edge detection and the identical match is used for hiding a secret message inside the video as presented in [16]. The proposed method is a combination of the 4LSB substitution method and the 2-bit identical match (ID). The first phase of this method is to DE-interlace the video into frames and randomly select some frames as covering images, which can embed the secret message behind the edge pixels. The edges are used to hide the secret information since the edges are extremely sharp, and their pixel's intensity values change frequently. The secret message is encrypted using the RSA encryption algorithm before the start of the embedding process. The test results show that this approach achieves high imperceptibility performance and a high hiding capacity and security for the video steganography process. The main fault of the method is that it modifies the pixel frame values, which change the quality of the video by disrupting the imperceptibility/invisibility and robustness.

In [17], data hiding and extraction is suggested for Audio Video Interleave videos, that embeds the image in Bitmap Image File, that has the secret data in a frame of the video by segmenting the bytes of the secret image and insertion them in the video frame providing a higher level of encryption. The proposed technique delivers two-level encryption, thus to decipher the data, how the secret image is initially decomposed, and the frame in which it is embedded should be identified. The quality of the secret image embedded and the size of the video is not changed before and after encryption of the secret data.

3. EMA-RKDD: UN-Substituted Video Steganography System

In most fields of Stego - key design and steganographic method, the power of Stegano - analysts and their progress in mathematical sciences and steganography have broken many steganographic systems. In contrast to the researchers, this evolution in the science of stegano - analysis has reacted to the development of steganography protection algorithms to ensure the highest level of protection for users. The key-dependent generation process, therefore, provides the security of the proposed video steganography as it is based on matching the secret value with the video pixel in which any video pixel value remains unchanged: the Stego - video produced as an original copy, though.

The proposed UN-Substituted video steganography system is aimed at enhancing existing traditional video steganography systems. Its abbreviation is EMA - RKDD, where the EMA is the Exact Matching Algorithm, and the RKDD is the Random Key - Dependent Data. The system emphasizes two primary goals. The first goal is not to alter the selected video to meet the Steganography systems ' imperceptibility attribute. The second goal is to propose a high - capacity information hiding methodology. The main idea behind EMA - RKDD is to find the complete, exact matching of the secret text (message) between the RGB values of the pixels of the extracted video frames. Then generate a Random Key – Dependent Data (RKDD) that is used as part of the recovery process of secret text. From many steganographic perspectives, the proposed system is more efficient than conventional steganographic systems EMA - RKDD system is shown in **Fig. 1**.

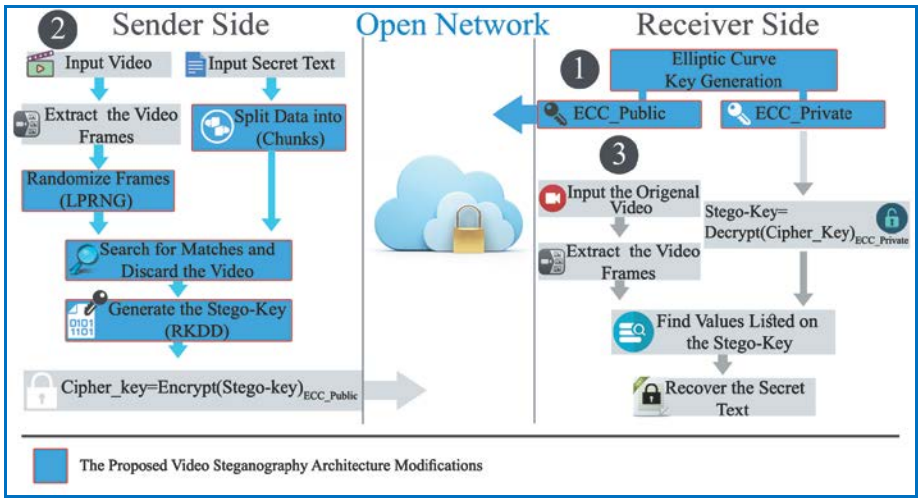

Fig. 1. The structure of the EMA-RKDD block diagram

The system is made up of two parts. These parts cover the sender and receiver. The system starts from the receiver that generates a public key using Elliptic Curve Cryptography. The sender will receive the key from the public. The transmitter system has three main phases. The first phase is the initialization and preprocessing phase. The user selects the necessary video and secret data. The system then begins with the frame extraction process, then converts the secret text into its ASCII code representation, then divides it into several chunks. The second phase focuses on finding the exact matches between the selected video frames ' RGB values and the secret text's ASCII codes. The output of this phase is a list of matched pixels, RGB values, and location of secret text.

Also, the output of this phase is used in formulating the Random Key Data Dependent (RKDD). The sending system is considered to be part of the third stage by the generation of RKDD. The previous stage output is used in this phase for randomly creating the RKDD. During this stage, the last output phase is used to generate the RKDD randomly. The stego - stick (RKDD) is ready for the following process. Huffman Encoding is used to compress the setgo - key for size reduction. The key is then encrypted with the Elliptical Curve (EC) encryption algorithm [18] then submits to the receiver.

The receiver uses the public key provided by the Elliptical Curve. The receiver takes all necessary steps to retrieve the secret text. The receiver must select the same video the sender has used; the receiver decrypts the data to make the secret text after deciphering the key.

3.1 Finding the Exact Matching (EM) between the Frames and Secret Message

As mentioned earlier, this section describes the process of the video frame extraction process and finding the EM between the video frame's RGB values, and the secret text.

3.1.1 Frames Extraction Process

The EMA - RKDD system uses an Audio-Video Interleaved (AVI) as an input video. The video frames are extracted and saved as portable network graphics (".png") file format which supports the lossless data compression. In particular, the system puts some constraints on the selected video. These limitations are listed as follows:

- The video type prefers to be (AVI).
- Selected videos should have more rich information, such as moving objects, backgrounds, and colors. White or only black background videos are considered to be extremely a lousy choice.

EMA-RKDD uses reference-frame extraction method [19] to extract the keyframes. This algorithm generates a reference frame and then extracts keyframes by comparing the frames in the shot with the reference frame.

3.1.2 Frames Randomization Process

A pseudo-random number (PRNG) generator is used to randomly select frames to be used as a covering medium [20]. The Lehmer Pseudo-Random Number Generator (LPRNG) is the basis for using many different generators of random numbers today. The random number generation LPRNG algorithm (also known as the Linear Congruential Generator) is defined in terms of two fixed parameters (large prime integer and integer). The LPRNG Pseudo-Random Number Generator (PRNG) is only used with the linear function, and we assume that only primary (real integration) numbers are used. Also, frames number are an integer in any video. The generators of LPRNG include three integer parameters (a, b, m) and an initial seed value (x_0) [21]. Defines the integers sequence based on the following in Equation (1):

$$x_{n+1} = (ax_n + b) \bmod m \quad (1)$$

Where,

x_n is the x^{th} Number in the sequence

m is a prime integer

a, b integers 2……m-1

As an example by using ($a=7$, $b=0$, $m=5$, and $\boldsymbol{x_0=1}$), the sequence begins with (*2, 4, 3, 1, 2…*), which includes the entire values of (*1…..m*-1) . The multipliers (*a, b*) and the prime modulus (*m*) must be properly selected for a maximum length sequence. The bad selection will provide an output of a short or poor sequence. The system uses the total number of frames after performing the extraction process as modulus (*m*) because the sequence length is *(m-1)* so that

the entire extracted frames are included. To avoid the repetition in the selected frame; the first sequence is taken by excluding the last number *(m-1)*. **Fig. 2** demonstrates the process of the frame randomization process.

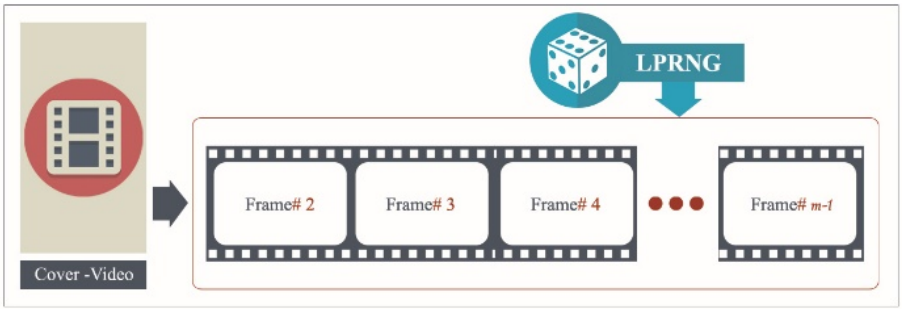

Fig. 2. EMA-RKDD frame randomization process

3.1.3 The Secret Data Splitting Process

To enhance the proposed system security, as an intruder finds it challenging to anticipate randomization process frames and how secret data are divided into small chunks. After finishing the frames' extraction and randomization processes, the secret data are divided into different chunks, which are suitable for the number of extracted frames. The purpose of splitting the data into different chunks is that EMA-RKDD assigns one chunk for each frame in the searching process. Also, to ensure that one character is not allocated to a framework. A single frame should have at least two characters assigned. This action reduces the searching time in the first place. Moreover, this process will keep the entire characters included in the searching process, and if any character(s) is not caught in a frame (s), it can be found in the other frame (s). Upon the accomplishment of the searching process, the EMA-RKDD will discard the whole matches of the repeated characters by only taking one match for each repeated character. The system calculates the total length of the secret text, and then divides the length of characters by the number of the desired chunks (n), as in Equation (2).

Number of Chunks= Total Numbers of Characters/*m*-1 (2)

Where,

m: Number of selected frames

Number of Chunks × (*m*-1) < Total Numbers of Characters

However, the number of the selected frames (*m*-1) does not precisely divide the characters' length; then the process ends up with an extra smaller chunk (reminder) at the end. The next step of the system is the pre-searching process. Its goal is to assign each secret data chunk to a selected frame as shown in **Fig. 3**. Start the process of finding matches between the secret text and the values of extracted frames (RGB), which will be explained in the next subsection. The search time is reduced by parallelization and randomization methods. They will improve system security by making it difficult for any intruder to anticipate the randomization and splitting method frames. It's important to note that the video issue won't be built up again. Another copy will be sent to the opposite side of the un-replaced the video.

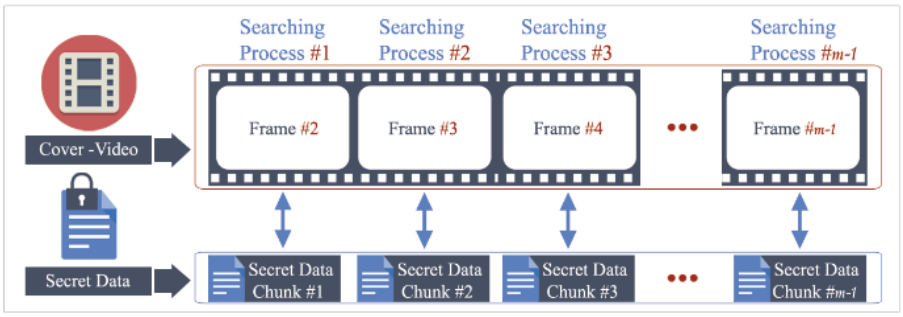

Fig. 3. EMA-RKDD pre-searching processes

3.1.4 Finding the Exact Matching (EM) between the Frames and the Secret Message

The system reads the RGB frames as a form of decimal values stored in a single - dimensional array. The secret text is converted into the representation of the American Standard Code for Information Interchange (ASCII). The algorithm Brute Force is used to determine matches between the RGB decimal frames and the secret text [22]. The brute force is one of the patterns matching algorithms. The searching process is conducted between each text chunk and its assigned frame, as shown in **Fig. 4**. After completing the searching process, the proposed system will return the following values as a list, where this list is used in the key generation process as follows:

1. The matched pixels locate the (x, y) values in each frame.
2. The frame number in which the matched values are located.
3. The locations of the characters in the original secret text.

EMA-RKDD returns the character (s) and its index into the secret text. When there are, no matches found the system changes the color channel to Green (G) and restarts the whole searching process. It is possible may switch to the Red (R) channel and restarts the searching process. As for now, the first phase of this system is completed.

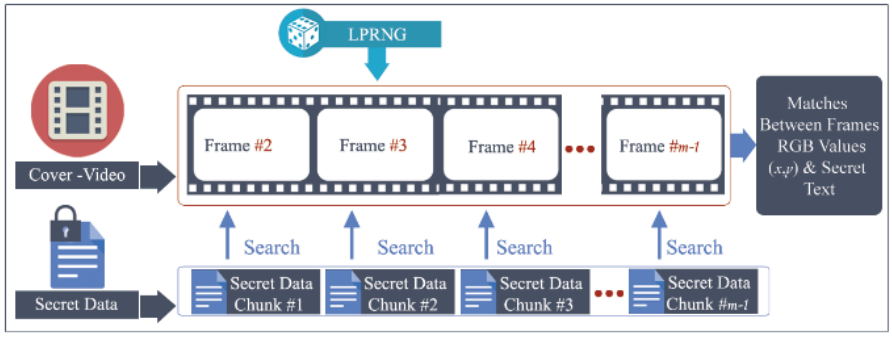

Fig. 4. EMA-RKDD searching process for the exact matches between the secret text and the frames RGB values

The algorithm of brute force is used to determine the matches between the decimal RGB frames and the secret text. The following are the main features of the brute force algorithm [22]:

- The preprocessing phase is not needed;
- A constant extra space is required;
- It always moves the window by just one location to the right;

- Comparisons can be made in any direction;
- The searching phase is considered in the O(mn) time complexity;
- 2^n denotes to the expected text characters comparisons.

Fig. 5 describes the pseudo-code of the searching process, which is performed by EMA-RKDD to find the matches between the secret text ASCII codes and the extracted video frames. The time complexity (Big - O) of the proposed steganographic video system, as discussed above, is based on the searching process for the exact matched value. The Big - O for the proposed steganographic video system – searching process, however, is O (mn).

1. **Start**.
2. **Input**: Required video, secret data, and seed value for the LPRNG.
3. **Extract:** Key frames from the video sequences.
4. **Generate:** A pseudo random number for the frame randomization.
5. **Randomize:** Fames by making use of the generated random numbers starting from the value (5).
6. **Divide:** Secret data into multiple chunks.
7. **Convert:** Secret data into the ASCII (decimal) code.
8. **Read:** Frames pixel values (decimal values).
9. **Start Searching**: For the exact matches between the frame and secret data.
10. **Store:** Matched pixels location (*x, y*) in the frames, and their locations in the secret data.
11. **Select**: One match from the frames values, which represent a character in the secret data.
12. **Store**: Matched pixels values in a list.
13. **Output**: A list of the matching pixels locations (x, y) and character's locations in the secret text.
14. **End.**

Fig. 5. The searching process pseudo code

3.2 The Random Key–Dependent Data Generation Process

The key generation process is a significant issue that is carefully taken into consideration. The EMA-RKDD system uses a new approach in formulating the key depending on the secret data and on the matching pixels in the video frames. The characters' locations values of the secret text are represented using integer numbers. In particular, each integer is represented as a four-bit binary digit number. This tactic helps us to cut the size of each element on the key from 8 to 4 bits.

Every key must have four fundamental values for each character. In the secret text, every character has an array position in the text. The array index indicates the location of the secret text character. The first value is the corresponding position of the pixels (x, y) in the video frames. The second is the frame number of the corresponding pixels. The third value is the location of the secret text character. Finally, for each key element, the fourth value is the

number of digits. The key structure of RKDD is shown in Fig. 6 as each digit on the key is shown separately.

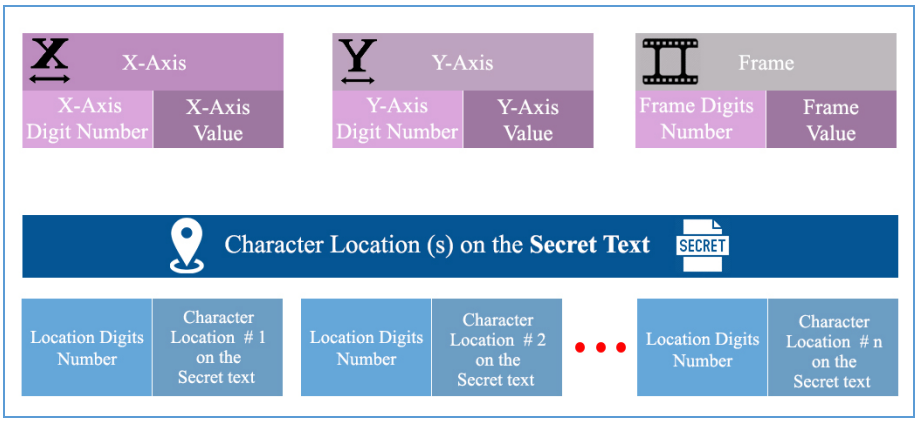

Fig. 6. The RKDD structure

Each matched character from the secret text will be removed from the search list during the search process and transferred to the next character. For (n) times, the proposed system will not store any repeated character location in the secret text, but will only take one match from the RGB values of the frame. After the last character has been found, the entire matched locations (x, y) in the frame will be stored in an array and the locations of the characters in the secret text. A sample of a secret text being searched in a selected video is shown in **Table 1**. The "thesis test" refers to the secret text. The total number of characters in this example is (11). Notice that the secret text repeats the characters (t), (e), and (s). Character (t) is repeated at (1), (8) and (11) locations where the character (e) is displayed at (3) and (9) locations. Character(s) exists in (4), (6), and (10) locations. In this case, only one matching value will be selected randomly by the system and stored for the entire repeated characters represented in the key generation process. This process, especially with large amounts of secret texts, reduces the search time.

Table 1. The Matching Values for the Sample Secret Data

X-Axis	Y-Axis	Char.	Char. ASCII Code	Char. Location On the Secret Text	Frame Number
0	85	T	116	1	4
15	150	H	104	2	7
8	85	E	101	3	6
0	101	S	115	4	3
7	85	I	105	5	5
0	101	S	115	6	3
0	33	Space	32	7	2
0	85	T	116	8	4
8	85	E	101	9	6
0	101	S	115	10	3
0	85	T	116	11	4

The character (t) is found in the number of the frame (4) in the pixel of the frame (0, 85). This character is repeated in three locations like (1, 8, and 11) in the secret text. **Fig. 7** clarifies the key representation example.

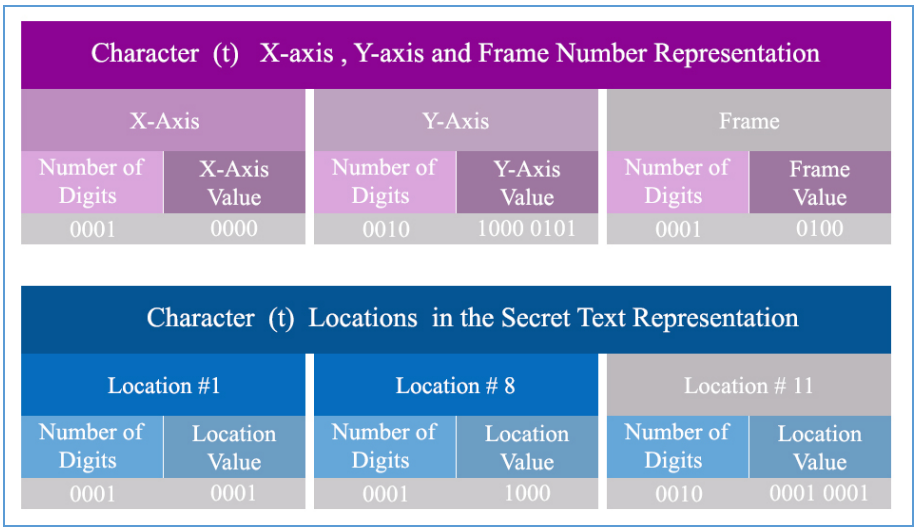

Fig. 7. The RKDD structure for "thesis test" example

In representing key values and size, EMA - RKDD uses a novel approach. When the key is generated, the system represents the number of digits for any encountered value, as shown in **Fig. 6**, starting from the left. The proposed method takes the search process output as an input to generate the Random Key - Dependent Data (RKDD) process. **Fig. 8** describes the RKDD key generation process pseudo code.

1. **Start**
2. **Input**: A list of matching pixel values (x, y), frame numbers, and a list of characters' locations in the secret text.
3. **Read**: The list of matched pixel values(x, y), frames numbers, and a list of characters' locations on secret text.
4. **Generate**: Stego-key according to the following order:
   a. X-axis number of digits & X-axis value.
   b. Y-axis number of digits & Y-axis value.
   c. Number of digits for the frame & the Frame number
   d. Character (s) and location(s) in the secret text.
   e. Repeat Steps (***a, b, c, d***) until the whole characters are represented.
5. **Compress:** The Stego-key.
6. **Encypt:** The Stego-key using Elliptic Curve Cryptography.
7. **Save**: The encypted Stego-key.
8. **Output**: Encypted Stego-key.
9. **End.**

Fig. 8. The pseudo-code for the RKDD key generation process

The key generation process (RKDD) is one of the proposed system's significant contributions. The O(mn) time complexity is considered to be the key generation algorithm.

3.3 RKDD Key Decoding Process

The process of decoding is based on the RKDD key. For the extraction process, the video selected by the sender must be taken as an input video. The system starts the RKDD decoding extraction process. The key decoding process begins with reading the first four bits representing the number of digits in the X-axis. After that, the following bits are read to recover the value of the X-axis. After the recovery process, the system's X-axis values move to the next four bits to recover the Y-axis number and the Y-axis values. The method continues with key recovery, frame number recovery, and frame-by-frame recovery of each character frame. Next, by converting the pixel values into the ASCII code, the system builds up the secret text again. The RKDD pseudo code and the process of secret decoding are defined in **Fig. 9**.

1. **Start.**
2. **Input**: The original cover-video as Stego-video and the Stego- key.
3. **Extract:** The original cover-video frames.
4. **De-Compress:** The Stego-key **.**
5. **Decrypt and decode**: The Stego-key values.
6. **Read:** The video in order to find the locations (x, y).
7. **Find:** The location of the matching pixels, which are listed in the Stego-key.
8. **Convert**: The values of these locations from decimal to ASCII.
9. **Build:** The secret message.
10. **Output**: The secret data.
11. **End.**

Fig. 9. The pseudo-code for the RKDD and the secret re-building process

4. Analysis and Experimental Results

This section discusses the proposed system implementation phase and the entire conducted experiments to evaluate EMA-RKDD technique. The Steganography system has certain features, such as capacity/payload, invisibility, robustness, and security. Steganography system features involved in the evaluation phase of the proposed method. This Section aims to elaborate on the effectiveness of EMA-RKDD Steganography system.

4.1 Implementation

The proposed UN-Substituted Video Steganography system (EMA-RKDD) is implemented in Windows 7 of a 64-bit version. In practice, it is implemented on a machine that has a processor of 2.10 GHz Intel Core i3 CPU, with a memory of 4 GB 1067 MHz DDR3. The EMA - RKDD system is implemented using VB.NET.

4.2 The Evaluation of the EMA-RKDD System against Steganography System Features

This section provides a thorough appraisal of the EMA-RKDD system against Steganography system core features (capacity, invisibility, and robustness). On the other hand, the security analysis is presented, analyzed, and highlighted. Other performance measures are assessed such as searching time, key and secret text size, and recovery time and accuracy. Finally, a

comparison between EMA-RKDD and current video Steganography techniques is constructed and discussed.

A. Capacity Feature

The main benefit of EMA-RKDD is based on its high payload capacity. The system is tested by using distinctive video and secret sizes. The conducted experiment aims to demonstrate that the proposed Stenography system can find a different match number for every single character of the secret text in the covering video frames. The primary test relies on the ability of EMA-RKDD to find out at least a match for each character in the covering video frames.

A nominated video called "birds.avi" with a size of 1.42 MB is used to experiment. For this investigation, only 10 extracted frames are used. The selected secret message consists of 1,000 characters. **Fig. 10** shows the results of this assessment.

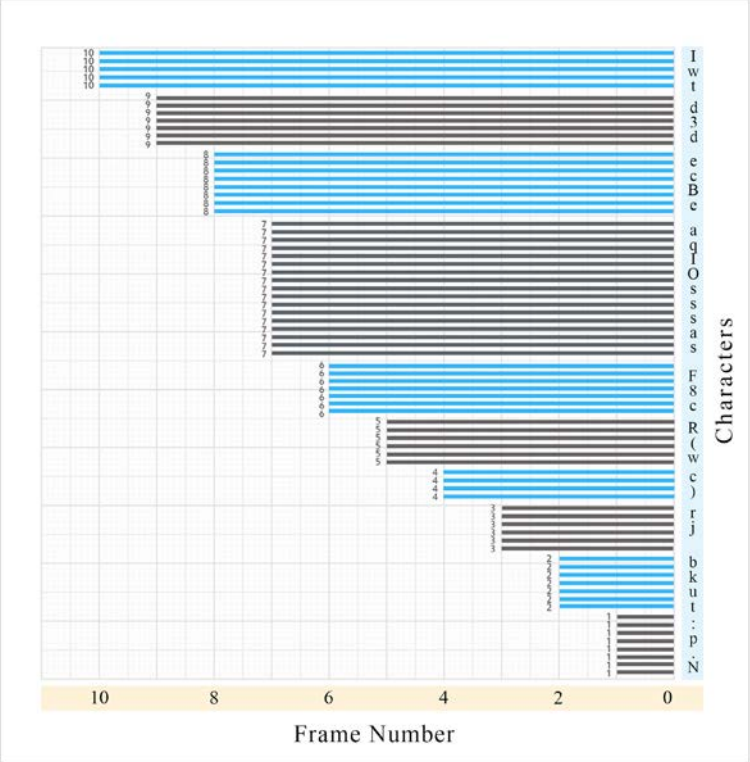

Fig. 10. Results for a sample of 1000 characters secret messages matched in 10 frames

Characters as (W, h, I, L, y, -, t) are found in the ten frames. Other characters such as (h, d, 3, H) are located in only nine frames. Notice at the bottom of **Fig. 10** that some characters are found merely in one frame.

In conclusion, the whole secret text character is often found in several frames of the selected video. Another observation is that most of the secret message characters exist or matching in the first four frames. There is no need to choose a large number of frames unless the user desires to spread out the secret message over a large number of frames.

Another experiment was conducted on EMA-RKD. EMA-RKD had been tested on different video sizes with different secret message sizes. Three different videos were selected with only three extracted frames from each video. The candidate secret message sizes are (4, 10, 30, 50, 100, 1000, 2000, and 3000) characters. **Table 2** clarifies the results.

Table 2. Results from testing EMA-RKD with different videos and secret text sizes

#	Secret Message Characters Number	Text Matches Number	Selected Video Name
1	4	17,427	MVI_3572.avi
2	10	38,247	
3	30	189,153	
4	50	138,507	
5	100	339,027	
6	500	53,953	
7	1000	255,015	
8	2000	230,789	
9	3000	563,472	
10	4	34,087	H264_test2_Talkinghead_avi
11	10	19,909	
12	30	88,388	
13	50	134,515	
14	100	157,626	
15	500	65,321	
16	1000	70,952	
17	2000	80,289	
18	3000	36,593	
19	4	25,757	H264_test3_Talkingheadclipped.avi
20	10	48,201	
21	30	233,347	
22	50	205,764	
23	100	248,326	
24	1000	162,983	
25	2000	155,539	
26	3000	573,492	

The test results demonstrate that EMA-RKDD system can find a large number of matches for the secret message. By studying the results, we inferred that the number of matches is based on the selected videos. The scenes in the candidate video, background, and animation all determine the matches' number. The quality of the video, which determines the number of matches and the time of the search, is another significant factor to be taken into account. According to the conducted experiments, the testing results show that the proposed system can find matches for each character in any secret message of any size.

An additional test to evaluate EMA - RKDD's capacity or payload is performed for further details. In this test, the size of the secret message increases each time randomly to assess the ability of the system in finding the matches, as is evident in **Fig. 11**. The secret message size ranges from 15 to 707,382 characters, where this is nearly equivalent to a document containing around (1460) pages.

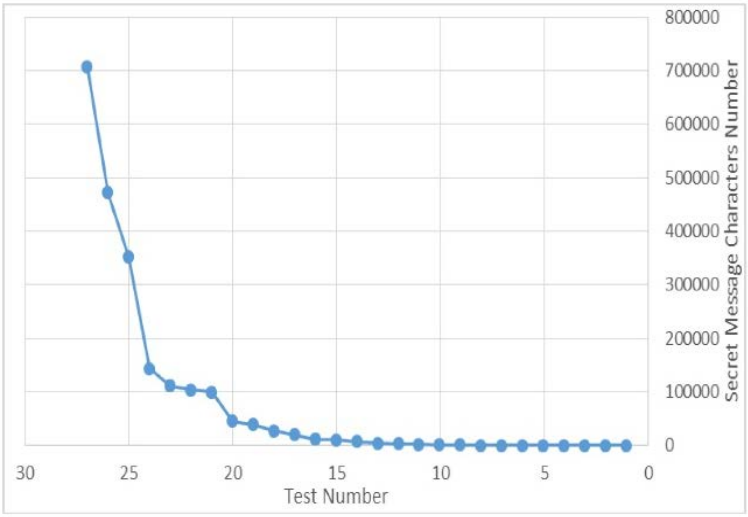

Fig. 11. Matched characters of different secret messages

B. Imperceptibility Feature

Imperceptibility implies that the hidden message must be perceptually and statistically undetectable. The cornerstone of the Steganography system is to hide the existence of the secret message. The EMA-RKDD system finds the matches, generates the random data-dependent key, and then discards the cover-video. Fidelity and quality are the two perceptibility metric values that can evaluate Steganography systems. Fidelity is the perceptual similarity between the cover-video and the Stego-video. In the proposed method, it is proved that the similarity is 100% since the cover-video is processed then wholly abandoned. This idea makes EMA-RKDD be a sole system, unlike other methods that modify the cover–file to conceal the secret message inside it. Quality is the measure of the integrity of the cover file. As mentioned previously, the quality of the cover video preserves. It is unchanged, and there is no added distortion to the frames.

The fidelity matrix represents both; the Peak Signal to Noise Ratio (PSNR) and the Mean Error Square (MES) [23] (please refer to **Table 6**). The PSNR measures the similarity between the cover-video and Stego-video. Both are similar and indistinguishable to each other if the PSNR value catches to be high. The MSE is the statistical difference of the pixel values between the original video and the Stego-video [24]. EMA-RKDD never changes any values of any pixels at all. The best quality is observed when the MSE value is small or close to zero. Calculated MSE value is close to zero in EMA-RKDD.

C. Robustness Feature

Robustness is an important feature that is characterized and addressed by any Steganography system. Robustness is against visual attacks and statistical attacks [23]. Visual attacks inspect the Stego-file visually to find any difference between the file prior and after the embedding process. The EMA-RKDD system does not perform any modifications to the cover media where this denotes that the cover media will not make attention, or suspicion to any hacker. The secret message invisibility is considered the most significant property of this system. Statistical attacks apply mathematical calculations by using computer-aided tools to find some changes to the cover- In short, EMA-RKDD is entirely undetectable and invisible by using visual and statistical attacks.

4.3 Security Analysis

The Stego-key is encrypted by applying the encryption algorithm Elliptic Curve (EC), which is based on the Elliptic Curve's discrete mathematical hard problem logarithm. The proposed steganography is also based on the matching values between the secret value and the video pixel selected, producing unchanged stego-video (original 100 percent). Since the results of the experiment showed that there are many matches based on values of(x, y) for each character. The proposed scheme was found to be secure against brute-force attacks from the security perspective.

We conclude that one significant advantage of EMA - RKDD is its improved security against highly qualified hackers. The security aspects are summarized in particular as follows:

1. No changes or modifications on the cover carrier (video).
2. The particular frame extraction method is used to extract the frames.
3. The random selection of the frames is involved in the virtual embedding process.
4. The secret text is divided into chunks based on a number of selected frames.
5. In the matching selection, there are always many matches for each character where only one of them is selected based on the values of (*x, y*).
6. The Stego-key is generated randomly based on the locations of matches of the characters in the secret text.
7. The Stego-key is encrypted by applying the Elliptic Curve (EC) encryption algorithm.

4.4 Searching Time Analysis

The searching speed is an essential concern that is evaluated in the testing phase. During the enhancements which had been performed on EMA-RKDD processes, two-optimization activities are conducted to enhance the system searching time in finding the matches between the secret message and RGB values of the video frames.

The system has been modified to skip any repetitive matched character during the search to overcome any of the factors that may affect the performance in the search. Second, the optimization step divides the secret text into chunks and assigns each chunk to a single frame. In **Table 3**, the searching time before the optimization process is highlighted. The check is conducted using two different videos with different sizes and different secret message sizes.

Table 3. Searching time before optimization

#	Secret Message Size (Bytes)	Time (Seconds)	Selected Video Name
1	10	14.36	MVI_3572
2	30	16.5	
3	50	17.30	
4	100	25.46	
5	500	31.6	
6	1000	35.23	
7	2000	38.8	
8	3000	43.78	
9	10	6.96	H264_test2_Talkinghead_avi
10	30	18.45	
11	50	20.34	
12	100	21.45	

#	Secret Message Size (Bytes)	Time (Seconds)	Selected Video Name
13	500	19.56	
14	1000	21.67	
15	2000	22.82	
16	3000	27.58	

Searching time is reevaluated after the optimization process (please refer to **Table 4**).

Table 4. Searching time after optimization

#	Number of Char.	Time (Sec.)
1	15	0.85
2	25	0.66
3	45	0.13
4	85	0.171
5	105	0.191
6	154	0.134
7	254	0.173
8	457	0.192
9	814	0.52
10	1462	0.73
11	2636	0.82
12	3061	0.87
13	4733	0.9
14	7624	1.21
15	11527	1.78
16	20499	2.25
17	39500	2.55
18	111366	2.85
19	143461	3.34
20	351634	3.94
21	472630	4.13
22	707382	5.22

The optimization processes improve the searching time. Before the optimization process, the searching time for a secret text consisting of (10) characters reached to (14.36) seconds, while the searching time after the optimization process for a secret text of (15) characters reached to (0.85) seconds.

4.5 RKDD Key Compression Results

A Huffman compression technique is used to compress Stego Key (RKDD). It is clear that the key size (RKDD) is large and must be reduced. There are two advantages to the compression process. First, a new security feature is added. Secondly, it has a significant decrease in size. The system searches for the pattern that has the heights in the key and gives this pattern the lowest weight. The compression process divides the key into blocks (b). To achieve the maximum compression ratio, the Block size (k) is selected. For example, depending on the key size,3-bits,4-bits,8-bits,16-bits, or 32-bits. No reminder is allowed when dividing the key into blocks (b), and all blocks should be of the same size. The process is demonstrated by Equation (3).

Number of Blocks (n) = Overall key Length / Block Size (k) (3)

Where,
n: denotes to the number of blocks.
k: indicates the block size.

The EMA - RKDD system performs a complete optimization on the Stego key. The initial optimization step is carried out to minimize the key size, summarized as follows:

1. Represent each integer from (0 to 9) by 4-bits instead of 8-bits.
2. Repeat characters take only one matching location (*x*, *y*) for the character to represent this character in the whole text.
3. All matches from the frame are grouped to represent the frame number one time only.

Compression is performed, and the key is generated. **Fig. 12** shows the Stego-key size before the compression process related to the secret text size.

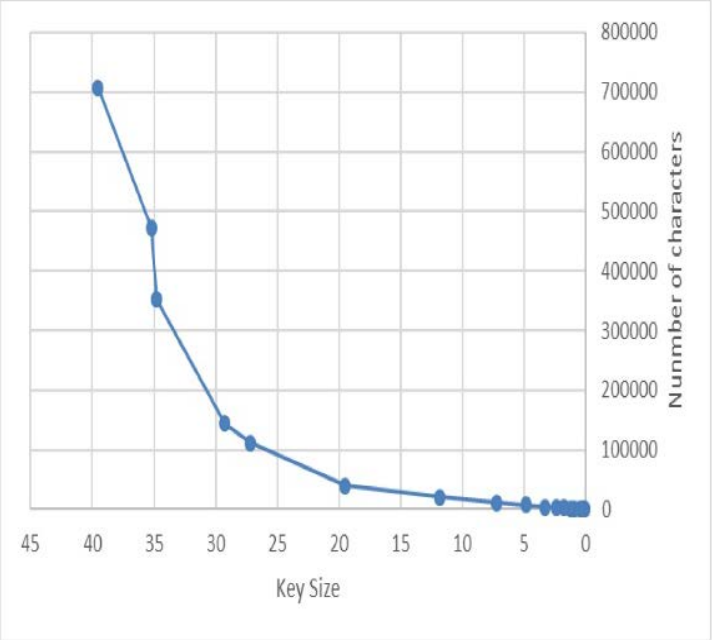

Fig. 12. Key size before the compression process related to secret text size

RKDD key size is lowered after using the Huffman Encoding Technique. The decrease is large. **Fig. 13** illustrates a comparison of the key size with different secret text sizes before and after the compression process.

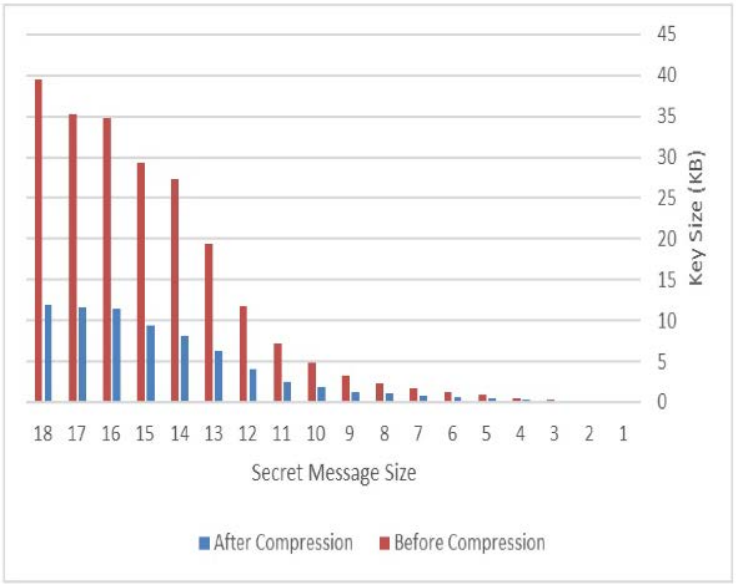

Fig. 13. Comparisons between the key size before and after the compression process

4.6 Secret Text Recovery Time

The accuracy assessment is performed on EMA-RKDD. The testing outcomes of recovery process related to textual accuracy are clear in **Table 5**.

Table 5. Recovery time and accuracy of EMA-RKDD

Secret Text Size (Char.)	Retrieve Time (Sec.)	Accuracy (%)
20	0.40	100%
30	0.41	100%
90	0.55	100%
130	0.58	100%
210	0.66	100%
300	0.75	100%
450	0.84	100%
600	1.1	100%
725	1.32	100%
829	1.45	100%
971	1.60	100%
4733	2.35	100%
7624	3.15	100%

4.7 Comparing EMA-RKDD to Other Steganography Systems

This Section compares EMA-RKDD system with other existing methods. **Table 6** demonstrates the comparison between the proposed method and four previous studies. The comparison is based on the PSNR and MSE fidelity performance measures.

Table 6. Comparison between EMA-RDD and previous systems

#	Author	Technique Used For Data Hiding	PSNR	MSE	Embedding Capacity
1	Sheng and Kin (2011)	Use at least 4 Significant Bits of each frame of the hosting video.	29 dB	0.015	55%
2	Sudeepa et al., (2014)	Data encryption is performed by dividing it up into equal parts (*m*) and by XORing it with the secret key. The result value is hidden in the randomly selected frames by using the LSB method	Undefined	Undefined	90,000 Char.
3	Elbyoumy Et Al. (2014).	Perform a bit by bit insertion (LSB).	42.9 dB	3.54	66.666%
4	Kaur Et Al. (2016)	A combination of the 4LSB substitution method and 2-bit identical match (ID).	66 dB	0.01838	50%
5	Khulood, Mohammed, Maher, and Eman (2018)	UN-Substituted Video Steganography	Infinite	Zero	100,000,000 Char.

The above table shows that the proposed system achieves better results than another existing system in comparison with the current systems. EMA-RKDD is providing higher capacity. The similarity between the cover-video and the original video is 100%, which indicates that there is a high (infinite) obtained PSNR value. Also, the MSE is zero since it measures the distance between the pixel values before and after the embedding process.

For the proposed Video Steganography, it is computationally impossible to mount an attack on the Stego-key secret. Since it is based on the discrete logarithm of Elliptic Curve property NP-hard problem (non-deterministic polynomial mathematical hard problem), which in this case uses compact, efficient key size (128-bit) which should give 2128 possibilities for every value of the key that is being attacked with brute force. However, the discrete logarithm problem on elliptic curve cryptosystem is more complicated than the other mathematical problem.

5. Conclusion

In this paper, we presented an enhanced UN-Substituted Video Steganography approach based on an exact matching algorithm and a random key-dependent data technique. We targeted to extract the exact matches between the secret text and the selected video frames. These matches are used randomly to generate Stego-key. The key is encrypted by applying the Elliptic Curve (EC) encryption algorithm. EMA-RKDD is characterized by its high embedding capacity and robustness against visual and statistical attacks.

The EMA - RKDD scheme is tested against video steganography benchmarks. In terms of capacity/payload, we found that our method can find the matches of any secret message with unlimited sizes. The assessment process reached a secret text size of (707,382) characters, which is approximately equal to a document of 250 pages. In terms of invisibility, the steganography scheme is considered invisible since the selected video is unmodified to be

notified by attackers. From the performance perspective, the average recovery time of every single character of the secret text is found to be about 4.6 milliseconds.

The evaluation results of the proposed system are considered extremely encouraging and are using a unique approach that has never been used by any of the existing Steganography systems, previously. Furthermore, the security of the proposed method is enhanced. The proposed system should be adopted by security agencies, military, IoT, and other interested parties. This paper provides a new understanding of the relationships between the secret text and the cover-objects, such as the images and the videos. Accordingly, a future implementation can be performed on other cover-objects, such as text files, HTML, Audio, and protocols. In the future, instead of randomizing, we can study approximation algorithms one more closely.

References

[1] Z. Khan et al., "Threshold-based steganography: A novel technique for improved payload and SNR," *The International Arab Journal of Information Technology*, vol. 13, no. 4, pp. 380-386, 2016.

[2] A. K. Pal, K. Naik, and R. Agarwal, "A Steganography Scheme on JPEG Compressed Cover Image with High Embedding Capacity," T*he International Arab Journal of Information Technology*, vol. 16, no. 1, pp. 116-124, 2019.

[3] S. Sun, "Image Steganography Based on Hamming Code and Edge Detection," *The International Arab Journal of Information Technology*, vol. 15, no. 5, pp. 875-881, 2018.

[4] M. A. A. Pujari and M. S. S. Shinde, "Data Security using Cryptography and Steganography," *IOSR J. Comput. Eng.*, vol. 18, no. 04, pp. 130-139, 2016. Article (CrossRef Link).

[5] M. Raggo and C. Hosmer, "Data Hiding: Exposing Concealed Data in Multimedia, Operating Systems, Mobile Devices, and Network Protocols," 2012.

[6] T. P. T. Gebreslassie, "Information Security Using Image Based Steganography," *International Research Journal of Engineering and Technology (IRJET)*, Vol. 03, Issue 06, pp. 2839-2844, 2016.

[7] S. Imaculate Rosaline and M. Ashok Raj, "Adaptive Pixel Pair Matching based Steganography for audio files," in *Proc. of 2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT)*, pp. 1–5, 2013. Article (CrossRef Link).

[8] A. Sakthivel and R. Nedunchezhian, "Analyzing the point multiplication operation of elliptic curve cryptosystem over prime field for parallel processing," *The International Arab Journal of Information Technology*, vol. 11, no. 4, pp. 322-328, 2014.

[9] S. Abed, M. Al-Mutairi, A. Al-Watyan, O. Al-Mutairi, W. Alenizy, and A. Al-Noori, "An Automated Security Approach of Video Steganography Based LSB Using FPGA Implementation," vol. 28, no. 5, 2019. Article (CrossRef Link).

[10] S. Venkatraman, A. Abraham, and M. Paprzycki, "Significance of steganography on data security," in *Proc. of International Conference on Information Technology: Coding Computing, ITCC*, 2004. Article (CrossRef Link).

[11] M. G.R and A. Danti, "A Novel Hash Based Least Significant Bit (2-3-3) Image Steganography In Spatial Domain," *Int. J. Secur. Priv. Trust Manag.*, vol. 4, no. 1, pp. 11-20, 2015. Article (CrossRef Link).

[12] R. Gupta, S. Gupta, and A. Singhal, "Importance and Techniques of Information Hiding : A Review," *Int. J. Comput. Trends Technol.*, vol. 9, no. 5, 2014. Article (CrossRef Link).

[13] S. D. Hu and K. T. U, "A novel video steganography based on non-uniform rectangular partition," in *Proc. of - 14th IEEE Int. Conf. on Computational Science and Engineering, CSE 2011 and 11th Int. Symp. on Pervasive Systems, Algorithms, and Networks, I-SPA 2011 and 10th IEEE Int. Conf. on IUCC 2011*, 2011. Article (CrossRef Link).

[14] M. El-Bayoumy, M. El-Mogy, A. Abou El-Fetouh, and R. El-Hadary, “A Proposed Technique for Hiding Encrypted Data in Video Files,” *Int. J. Comput. Appl.*, vol. 79, no. 10, pp. 38–42, 2013. Article (CrossRef Link).

[15] K. B. Sudeepa, K. Raju, H. S. Ranjan Kumar, and G. Aithal, “A New Approach for Video Steganography Based on Randomization and Parallelization,” *Phys. Procedia*, vol. 78, pp. 483–490, 2016. Article (CrossRef Link).

[16] R. Kaur and S. Kaur, “XOR-EDGE based Video Steganography and Testing against Chi-Square Steganalysis,” *Int. J. Image, Graph. Signal Process.*, vol. 8, no. 9, pp. 31-39, 2016. Article (CrossRef Link).

[17] S. Manisha and T. S. Sharmila, “A two-level secure data hiding algorithm for video steganography,” *Multidimens. Syst. Signal Process.*, vol. 30, pp. 529-542, 2019. Article (CrossRef Link).

[18] M. A. Alia, “Cryptosystems Based on Chaos Theory," *Erçetin Ş., Banerjee S. (eds) Chaos, Complexity and Leadership 2013, Springer Proceedings in Complexity. Springer, Cham*, pp. 129-145, 2015. Article (CrossRef Link).

[19] A. Nasreen and G. Shobha, “Key Frame Extraction from Videos – A Survey,” *International Journal of Computer Science and Computer Networks (IJCSCN)*, vol. 3, issue 3, pp..194-198, 2013.

[20] A. A. Maaita, and H. A. A. Al Sewadi, "Deterministic Random Number Generator Algorithm for Cryptosystem Keys," *International Journal of Computer and Information Engineering*, vol. 9, no. 4, pp. 972-977, 2015. Article (CrossRef Link).

[21] N. State, "Fibonacci Random Number Generator using Lehmer ' s Algorithm," *Mathematical Theory and Modeling*, vol. 3, no. 14, pp. 56-63, 2013.

[22] A. Mohammad, O. Saleh, and R. A. Abdeen, "Occurrences Algorithm for String Searching Based on Brute-force Algorithm," *J. Comput. Sci.*, vol. 2, no. 1, pp. 82-85, 2006. Article (CrossRef Link).

[23] M. A. Alsarayreh, M. A. Alia, and K. A. Maria, "A Novel Image Steganographic System Based on Exact Matching Algorithm and Key-Dependent Data Technique," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 5, 2017.

[24] E. Cole, “Hiding in Plain Sight: Steganography and the Art of Covert Communication,” *Wiley*, 2003.

Khulood Abu Maria is an Assistant professor at the Computer Information Systems department, Faculty of Science Computer and Information Technology, Al Zaytoonah University of Jordan. She received the B.Sc. degree in Computer Science from the Mutah University, Jordan, in 1992. She obtained her Ph.D. degree in Computer Information System from University Science of Arab Academy for Banking and Financial Sciences, in 2008. During 1992 until 2006 she worked in Petra Engineering Industries Co. as Programmer, Analyst, Network Administrator, and IT Manager. During 2008 until 2009, she worked at Al-Isra University as a part-time instructor of Management Information System. During 2009 until now, she worked at Al-Zaytoonah University of Jordan as an instructor of Computer Sciences and Information Technology-CIS Department. Her research interests are in the fields of Network Security, Artificial Intelligent, Agent-Based System, Information System, and Software Engineering.

Mohammad Alia is a Professor at the computer information systems department, Faculty of Science and information technology ,Al Zaytoonah University of Jordan. He received the B.Sc. degree in Computer Science from the Al Zaytoonah University, Jordan, in 2000. He obtained his Ph.D. degree in Computer Science from University Science of Malaysia, in 2008. During 2000 until 2004, he worked at Al Zaytoonah University of Jordan as an instructor of Computer Sciences and Information Technology. Then, he worked as a lecturer at Al-Quds University in Saudi Arabia from 2004 - 2005. Currently, he is working as a Faculty of Science & IT deputy dean at the Al Zaytoonah University of Jordan. His research interests are in the field of Cryptography and Network security.

Maher A. Alsarayreh received his B.Sc. degree from Arab Open University in 2013, and his M.Sc. in Computer Science from the Al Zaytonah University of Jordan in 2017. His research interests are in the field of Image Processing and Steganography.

Eman Abu Maria is an instructor at the Computer Information Systems department, Faculty of Science Computer and Information Technology, Al Zaytoonah University of Jordan. She received the B.Sc. degree in Computer Science from the Al Zaytoonah University, Jordan, in 2001, and her M.Sc. from the Arab Academy for Management, Banking and Financial Sciences in 2005. Her research interests are in the fields of Network security and Agent-Based System.