Journal of Positive Sciences (JPS) eISSN: 2582-9351, https://positive-sciences.com JPS. 2023 July; 3(4): 8-21

Article info
Received on: 03.07.2023
Accepted on: 29.07.2023
Published on: 31.07.2023
doi: https://doi.org/10.52688/ASP525038

Research Article

An encapsulation technique for cryptography based on steganography

Muthanna Jabbar Abdulredhi [1, *] , Ali Munther Abdulrahman [2], Mohammed A. Fadhel [3]

[1, 2] Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
[3] College of Computer Science and Information Technology, University of Sumer, Thi Qar, Iraq

[*] muthanna.jabbar@uoitc.edu.iq

ABSTRACT

In cryptography, plain data is transformed into an incomprehensible form. In this paper, we propose an integrated approach to integrated cryptography and steganography for RGB images. A steganography method for concealing secure information using another cover form, such as images, voice, or videos. The Elliptic Curve Cryptography (ECC) encryption method and Pixel-Value Differencing (PVD) method are used here to achieve that extremely high security level. As a result, the robust protection algorithm uses higher bit sizes within the Red, Green, and Blue channels of the cover image, yielding very high-level security levels when transferring the secure information over unsecure communication channels.

Keywords: Cryptography, Steganography, ECC, PVD, Security

INTRODUCTION

Due to the rapid rise in information technology developments and the rapid changes in human life, there is a need to maintain the security of clandestine data when it is sent across the internet, since the pace of life is constantly changing as well as the dramatic increase in information technology development. Before clandestine digital data is sent, it is generally prepared for transmission. As a result of this digital data preparation mutating the data into a different appearance, only a commissioned person is able to carry out the invertible operation on the altered data after it has been altered as described before. The clandestine transmission of digital data is protected by a variety of data protection methods. Cryptography is one of the most commonly used methods of data protection. Although data encryption creates senseless codes for communication, it may be exploited by prying parties to alter messages intentionally or to gain access to them through cryptographic attacks, despite the fact that it generates senseless codes for communication. One of the best ways to secure data is to use a method known as Steganography, which prevents any change in its form from affecting it and making it meaningless to prying eyes. It involves concealing data in any unexpected cover media, such as images, audios, and videos, creating a symbolic message that is known as stego-media. As a result of the hidden data contained in stego-media and authentic cover media, it is very difficult for a human to distinguish between these two types of media. Cryptographic and steganographic-based security systems are used today in a variety of applications, including military telecommunication, Internet of Things (IoT), banking, social media, and healthcare, but the latter requires more datasets for training the network [1].

A malicious user or attacker becomes increasingly adept at manipulating information systems and gaining access to clandestine data (disclose, modify, deform, or utilize it) by attacking it. Developing steganography or cryptography separately has im-proved data security, but they still have some vulnerabilities. So, integrating ste-ganography and cryptography is necessary to solve this problem. In order to ensure a very high-security level when sending and receiving data, this paper uses a combina-tion of Elliptic Curve Cryptography (ECC) and Pixel–value Differencing (PVD) [2].

RELATED WORK

There are more developed methods for securing data when it is transferred over the internet, because data transfer requires a higher level of security. The best way to address this requirement is to integrate cryptography and steganography. This integration hides the encrypted message (unreadable message) in the cover image (stego image).

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

Various works combining steganography and cryptography are reviewed in this section. By combining neural networks and visual cryptography (image-based data protection), Seethalakshmi et al. suggest ways of improving image steganography security. The message is encrypted using the AES algorithm, then the cover image is split into blocks and the strength coefficients for each block are determined by analysing the Integer Wavelet Transform (IWT). It uses the neural network approach to embed encrypted data at the perfect location, followed by LSB steganography to embed encrypted data.

Bukhari et al. used the LSB method to insert an image message into the cover image to secure it over an unsecured wireless channel. Each block of the stego image is then multiplied by the first random phase. By using a double random phase (converted to white stationary noise), the stego image is divided into eight 8 by 8 blocks. Next, the multiplied image is converted to a frequency domain using the Fourier transform. Finally, the image is convolved with a second phase mask[3]

A proposal by R. Das and I. Das [4] is that the use of steganography and cryptography during data transfers between IoT devices could be used systematically in order to provide a secure platform for data transfer. In order to protect the information of the IoT device, it is ciphered and inserted into the cover image along with a summary of what has been gathered. After the information has been sent to the local server, it is extracted, compared to the original message summary, and authenticated.

It was proposed by Joshi and Yadav [5] to combine a new method of cryptography with gray-level steganography in a new procedure. After the Vernam cipher algorithm was employed to encrypt the clandestine message, the authors claim that it increased the hiding capacity to 100% after it was inserted into the cover image with Shift LSB Steganography, which enabled the message to be hidden to a high degree.

A space domain steganography method has been proposed by Patel and Meena [6]. The secret image is separated from the carrier image, while the carrier image is of the same size. It is then determined which of these channels will be used for the two images, generating a pseudorandom noise sequence using that key. Each image is split into R, G, and B channels, and then one of these channels is selected for each image. A carrier image (CI) is split into 16 pixels blocks, and then each block is selected the same way it appears on the screen. The second key allows the secret image to be divided into 16 pixel blocks based on row and column sequences, followed by the chosen pixel ciphers, and then finally inserted into the carrier image.

An AES cryptographic algorithm and LSB steganography algorithm are integrated by Suchitra and Madona [7] to create a high-level security platform, where the AES algorithm ciphers a clandestine message to generate the ciphertext. Pixel Value Differencing (PVD) with K-bit LSB is then used to insert the ciphertext into RGB images. In the proposed method, the image quality is high, the embedded capacity is high, and errors are minimal..

According to Sahu et al. [8], clandestine messages can be hidden by flipping the 7th and 8th bits of each block of pixels (bit flipping method), and then using three methods, either inserting 3 bits of the clandestine message into these blocks or inserting 4 bits of the clandestine message into these blocks. Compared to the existing bit flipping method, the proposed approaches show better performance regarding peak signal-to-noise ratios and hidden capacities. According to the proposed approaches, PSNR and HC have values of 47.38 dB, respectively, and 524,288 and 393,216 bits, respectively.

Sahu and Swain [9] propose overlapped pixel-value differencing with modulus function, and overlapped pixel-value differencing (OPVD) to improve the pixel-value differencing (PVD) method to acquire an HC for the frame-image and a PSNR for the cover image by dividing it into 1*5 pixels non-overlapping blocks. For PVD, the OPVDMF finds a difference between the four pixels and the 5th one; otherwise, it divides the block into 4 subblocks (1st, 5th) - (5th, 2nd) - (3rd, 5th) - (4th, 5th). According to the proposed methods, peak signal-to-noise ratios (PSNR) for OPVDMF and OPVD are 36.03 dB and 37.01 dB, respectively, and hidden capacity (HC) is 825,730 bits for OPVDMF and 786,741 bits for OPVD. Comparing PSNR and HC between the proposed methods and existing modern techniques, the proposed methods demonstrate outstanding performance.

Aditya Kumar Sahu and Gandharba Swain [10] propose two methods that enhance PSNR and HC by using pixel value differencing with modulus function (PVDMF). Using adaptive two tables, both methods determine whether sequential pixels are non-overlapping, then hide messages. A peak signal-to-noise ratio (PSNR) of 42.04 for OPVDMF 1 and 38.36 for OPVDMF 2 is determined by the proposed methods. The hidden capacity (HC) of OPVDMF 1 is 577,832 bits, and the hidden capacity (HC) of OPVDMF 2 is 786,741 bits. PVDMF 1 increases PSNR, while PVDMF 2 increases HC.

By integrating three technologies: remainder replacement (RR), adaptive quotient value differencing (AQVD), and quotient value correlation (QVC), Reshma Sonar & Gandharba Swain [11] propose a new technology to address inclusion process problems using PVD and APVD technologies. There was a split of three parts in this block, a part for content and a part for rest (RR), the rest part was masked with two bits and replaced with a decimal, and the data was hidden at the corners using AQVD, while the secret bits were hidden using QVC, resulting in a masking rate of 3.12 bits and a signal to noise ratio of 35.27dB.

BACKGROUND

In this section, cryptography and stenography are discussed in briefly as follows:

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

CRYPTOGRAPHY

Cryptography involves converting a plaintext message into a ciphertext message through encryption or ciphering algorithms to protect sensitive information. Cryptography algorithms are divided into two categories: symmetric algorithms that use a single, private or secret key[11], and asymmetric or public-key algorithms that use distinct key pairs[12]. Elliptic Curve Cryptography (ECC) is a type of public key cryptosystem that utilizes implicit elliptic curves with different levels of security and faster key generation than RSA (ECC uses smaller encryption/decryption keys than RSA approximately 160–256 bits for ECC compared to 1024–3072 bits for RSA)[11]. Cryptography is commonly used in various applications such as connecting to Wi-Fi, mobile networks, social media, emails, e-banking, shopping, etc.

STEGANOGRAPHY

Steganography is another method used to protect sensitive information by embedding clandestine data into images, audio files, or videos (cover media) to create stego media. There are generally four types of steganography methods: Spatial Domain Methods, Transform Domain Technique or frequency domain methods, deformation methods, and masking and filtering methods. Spatial domain methods involve using hidden bits to change the values of the pixels of the cover image. Pixel-value differencing (PVD) is a widely used approach for data concealment. Steganography techniques have limitations and must preserve the statistical characteristics of an image when using image steganography [13].

In conclusion, both cryptography and steganography are important methods of protecting sensitive information. Cryptography involves converting plaintext messages into ciphertext using encryption algorithms or ciphering algorithms, while steganography involves interjecting clandestine data into images, audio files, or videos to create stego media. Both methods have various techniques and algorithms that can be used, and the choice of technique or algorithm depends on the specific application and the level of security required. For more inserted messages, this paper uses the RGB color image PVD method where the message is inserted into the RGB, B, and G channels.

PROPOSED TECHNIQUE

To conceal data, we use cryptography and steganography in this paper to achieve two levels of security. Our first step is to encrypt the message using the hybrid form of elliptic curve cryptography (ECC). In addition, the ciphertext is inserted into the cover image using Pixel Value Difference (PVD) stenographical techniques on RGB channels with a higher concealing capacity and satisfactory visible of the stego-image using this method. Fig. 1. Describes the paper's proposed technique.

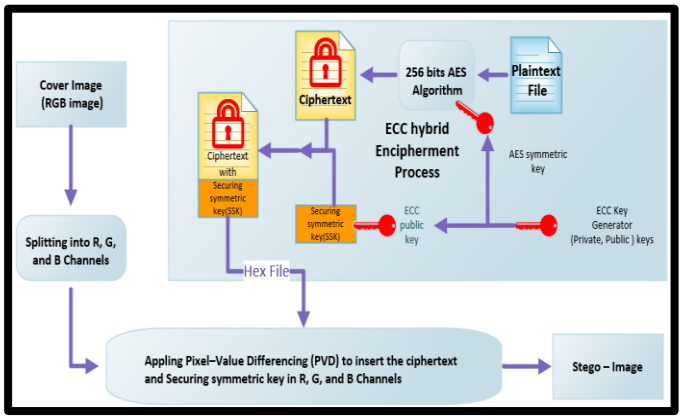

Figure 1: The proposed technique's pipeline

PRINCIPLE OF ELLIPTIC CURVE ENCRYPTION (ECC)

It is usually the case that public and private keys are used directly in the cipher/decipher process in asymmetric encrypting systems, but ECC does not work; instead, ECC encrypts using the ECC algorithm. As opposed to this, Elliptic Curve Diffie–Hellman (ECDH) is a symmetric encryption algorithm and key exchange. The curve is made up of points (x, y) that can be described in terms of a general equation. [14]:

$$y2 + cxy + dy = x3 + ax2 + ex + b \quad (1)$$

As shown in the equations, the encipherment systems use the simplified Weierstrass elliptic curve (c = d = e = 0):

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

$y^2 = x^3 + ax^2 + b$ (2)

In ECC, equation (2) is specified by using a finite field (Galois field) named Fp with prime numbers p>3 and F2m with prime numbers 2m and 2m, where x and y are integer values within the matrix, and when addition, doubling and multiplication are performed on this matrix, another point(s) within this matrix is generated. The equation becomes:

$y^2 = x^3 + ax^2 + b \ (mod\ p) \quad where\ 4a^3 + 27b^2 \neq 0$ (3)

Where the mod or module is the operation that gets the remainder of the division and a, b ∈ Fp, and the key range in ECC is points {x ∈ [0 ➔ p-1], y ∈ [0➔ p-1]} ∈ Fp.

One crucial property elliptic curve in a finite field is the multiplication of pre-determined point (generator point G) on the curve by an integer value, which will generate another point on the curve.

$Gnew = k * G \text{ and } G, Gnew \in Fp$ (4)

By using a vast random number and a predetermined point G, ECC generates a huge private key and a new public key (new point) compute by:

*Pubic key = private key * G* (5)

An elliptic curve is selected based on the time it takes to compute the public key (log2 (private key), which is calculated by adding points to it repeatedly (double-and-add method). With 256-bit curves (i.e., p bits length = 256 bits), adding points to only a few hundred elliptic curves was sufficient. The second parameter is security level that computed by √(private key) . For example, the 256-bit ECC is approximately a 128-bit security level [14]. A large number of standard elliptic curves are available in the ECC, such as the 192-bit curve (secp192r1), the 233-bit curve (sect233k1), the 224-bit curve (secp224k1), the 256-bit curve (sect256k1, sect256r1, and Curve25519), the 283-bit curve (sect283k1), and the 384-bit curve (curves p384 and secp384r1). Cryptographers thoroughly examine the standard elliptic curves to ensure their security levels.

Encryption(Encipherment) Algorithms:

As mentioned previously, this paper uses the elliptic curve cryptography (ECC) as a hybrid encipherment blueprint for AES symmetric data encryption (encipherment) and decryption (decipherment) processes to produce a shared secret key with the ECDH (Elliptic Curve Diffie–Hellman) key exchange blueprint. We use the 256-bit curve "secp256r1" as private and public keys generation, suppose we have elliptic curve over Fp, and base point G(x, y) in the curve as shown in algorithms[1,2,3]:

Algorithm1: Compute Encryption (Encipherment) key

Input: 256-bit curve "secp256r1"

Output: sender public key (sePubKey), and sender private key (sePriKey)

Begin

Step 1. Curve, G(x, y)➔ Get Curve from "secp256r1"

Step 2. Generate sender private key (sePriKey) ➔ new random (private key)

Step 3. Compute sender public key (sePubKey) ➔ sePriKey * G(x, y)

Step 4. Return (sePriKey, sePubKey, curve, G(x, y))

End

Algorithm 2: AES symmetric key cipher

Input: plaintext (message), AES shared key (aesShrKey) 256bits

Output: AES ciphertext (aesCiphertxt)

***Corresponding author**
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

Begin

Step 1. Divide plaintext into 128 bits blocks

Step 2. Divide each block into 4 by 4 square matrix as name blocki

Step 3. Initialize round for keys to 14 rounds as Keyi ➔ aesShrKey

Step 4. Add round key ➔ XOR (blocki , Keyi)

Step 5. While rounds between 1 to 14 repeat step 6 to step 9

Step 6. Cry out Bytes substation (blocki)

Step 7. Cry out shift rows (blocki)

Step 8. Cry out Mix column (blocki)

Step 9. Add round key ➔ XOR (blocki , Keyi)

Step 10. Return aesCiphertxt ➔ blocki

End

Algorithm 3: Elliptic curve cipher

Input: plaintext (message)

Output: AES ciphertext (aesCiphertxt), receiver public key (rePubKey)

Begin

Step 1. Get curve, G(x ,y) ➔ call Algorithm 1: compute encryption key

Step 2. Generate receiver private key (rePriKey) ➔ new random (private key)

Step 3. Compute receiver public key (rePubKey) ➔ rePriKey * G(x ,y)

Step 4. Compute ECC shared key (eccShrKey) ➔ rePubKey * sePubKey

Step 5. Compute AES shared key (aesShrKey) ➔ convert to 256bits key(eccShrKey)

Step 6. Get AES ciphertext (aesCiphertxt) ➔ Call Algorithm2 AES symmetric key cipher (message, aesShrKey)

Step 7. Return (aesCiphertxt, rePubKey)

End

PIXEL-VALUE DIFFERENCING (PVD) FUNDAMENTALS

In the PVD technique [16, 17], a grayscale cover image is used, and clandestine message bit series with a variable length are inserted into the cover image. The smaller clandestine message bit series are inserted on the flats instead of the more bits on the brims. The cover image is divided into non-overlapping blocks and is then horizontally scanned. Assume PXi and PXi+1 are two

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

sequential pixels within (ith) block inside the cover image, then calculate the absolute difference value by di =| PXi – PXi+1 |, if the difference is slight, then it will flatten region, while bigger value represents brim region. The di values occur between 0 and 255 because of the grayscale that consists of 256 level values, which divide into six regions as shown in Table 1 [16, 17].

Table 1. Mapped ranges

	R1	R2	R3	R4	R5	R6
Range	8 (3bits)	8(3bits)	16(4bits)	32(5bits)	64(6bits)	128(7bits)
	0-7	8-15	16-31	32-63	64-127	128-255

The di map to one of the regions, where the lower and upper bound of each Ri is represented by [LOi, UPi], the number of inserted bits (tbits) can be calculated by tbits = floor($\log_2$(UPi – LO_{i+1})), then transform tbits to its corresponding decimal value (tdec), after that calculate new difference value (d_{new}) where $d_{new} = t_{dec} + LO_i$, then calculate m = | d_{new} – d_i |, finally finding the new pixels pair. The new pixels values PXinew,PXinew+1 will substitute instead of the old value PX_i and PX_{i+1}. The main concept is explained by algorithm [4]:

Algorithm 4: Embedding (Inserting) Ciphertext

Input: ECC ciphertext (aesCiphertxt, rePubKey) as Hex File, Cover Image(covImg)

Output: inserted Ciphertext inside Sego–Image(sgImage)

Begin

Step1: Dividing the cover Image into R, G, and B Channels and saving them into three vectors, Rchan, Gchan, and BChan, respectively.

Step2: Convert the Hexadecimal file to binary vectors, then determine the length of the binary vector.

Step3: TempChan= Rchan

Step4: For TempChan, create sub-vectors from two non-overlapping adjacent pixels scanned horizontally [PXi, PXi+1]

Step5: Create two Vectors of upper UP and lower LO bounds

Step6: Compute di ➔ abs(PXi – PXi+1)

Step7: According to table 1, determine which region (Ri [UPi, LOi]) that di belong to, then compute the number of bits that will be inserted (tbits = floor(UPi - LOi + 1)), cut these bits from the ciphertext, and convert it to decimal value tdec.

Step8: Compute dnew tdec + LOi , and m➔ abs(dnew – di).

Step9: Check (PXi, PXi+1) and (dnew – di) then calculate new pixels PXinew, PXinew+1

Step10: Repeat Steps 3 through 8 until TempChan is complete, then calculate total inserted bits in TempChan,

Step11:Calculate the remaining binary vector if it is empty stop, else repeat steps 4 through 10 for TempChan =Bchan and TempChan = Gchan

End

For Example, if we consider the plaintext message is This is the Plain Text' and after implementing ECC using python, and the PVD implements using MATLAB, we obtain (0x mean in hexadecimal form):

The Plaintext: This is the Plain Text;

The Ciphertext:

0x44e0d17d127277590bc66af2aed228318025e17be9fc525dd958240dbbdd6ae4f2e6e115fbf8cecefa32e713652580e3223754db2f012c

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

The sending private key: 0x4d3ac0f0ece848240d9030e956f0e68c73151ccbf1b75f842cdf95675d4cf9c0;

The ECC public Key point (x, y)=

(0xad41d7beea8c901f06708ff5afa6264a677f2ccea6da461793fbd92ce93e364,
0xf6ec1c46d0bbd093c4b29abebafe0c70ea907b9aed56747ae2127a13e88b8c37);

Length Ciphertext= 440 bits;

Length of Private Key: 256 bits;

Length of sending ECC Public key{x}: 256 bits;

Length of sending ECC Public key{y}: 256 bits

And the Binary vector:

'0100010011100000110100010111110100010010011100100111011101011001000010111100011001101010111100101010111011010010001010000011000110000000000100101111000010111101111101001111111000101001001011101110110010101100000100100000001101101110111101110101101010111001001111001011100110111000010001010111111011111110001100111011001110111110100011001011100111000100110110010100100101100000000111000110010001000110111010101001101101100101111000000000100101100';

Assume that the two sequential pixels inside cover image is:

[PXi, PXi+1] = [73, 54]; then di = abs(73– 54)= 19;

From Table 1 obtain: Ri [UPi, LOi] = [31, 16], and tbits = floor (31 – 16 + 1)= 4 bits;

The Cutting bits are: ‘**0100**’; tdec =4; dnew=4 + 16 =20; m = |20 – 19| = 1,

The New Pixels are:

(PXinew,PXinew+1)= 73+ceil(1/2),54-floor(1/2)**; 73≥54 and 20>19**;

The new inserted pixels in stego image are [PXinew, PXinew+1] = [**74, 54**].

For stenographical security levels measurements that describe the effectiveness of inserting cipher message into stego-images quality (or how the inserted message effect image distortion), the MSE and PSNR. Where Cov(i, j) is the cover image and Steg(i, j) is the stego–image with m rows and n columns. The lower value of MSE or higher value of PSNR means less distortion or insensible original image changing detected by human eyes (excellent hidden data) [16,17].

EXTRACTION (DATA EXTRACTION) AND DECRYPTION (DECIPHERMENT) ALGORITHMS

Data extraction begins with the stego–image after the ciphertext is determined using the data extraction algorithm. To obtain the plaintext or the original message, the deciphering algorithm is applied to the ciphertext. Figure 2 illustrates the process of obtaining the plaintext and deciphering the message.

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

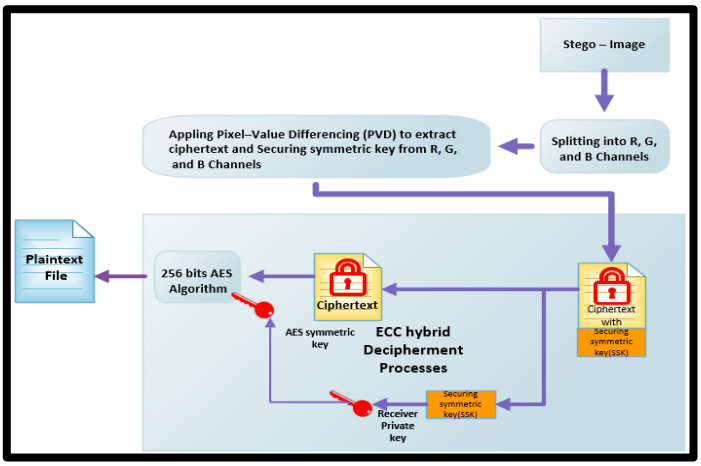

Figure 2: Processes for extracting and deciphering data

Algorithms [5,6] explains the main steps of Data Extraction, Decryption, AES decryption, and ECC decryption.

Algorithm 5: Extraction Algorithm (Data Extraction)

Input: stego–image

Output: Ciphertext hexadecimal file

Begin

Step 1. Dividing the cover Image into R, G, and B Channels and saving them into three vectors, Rchan, Gchan, and BChan, respectively.

Step 2. For Rchan, create sub-vectors from two non-overlapping adjacent pixels scanned horizontally [PX_i, PX_{i+1}]

Step 3. Compute d_i □ abs($PX_i - PX_{i+1}$).

Step 4. According to table 1, determine which region (R_i [UP_i, LO_i]) that d_i belong to, then compute the number of bits that will be inserted ($\mathbf{t_{bits}}$ **= floor(**$UP_i - LO_i + 1$)), after that compute the decimal value of inserted message by $\mathbf{t_{dec}}$ □ abs($d_i - LO_i$).

Step 5. Convert $\mathbf{t_{dec}}$ to binary number this binary number insert to binary vector.

Step 6. Repeat steps 2 through 5 until Rchan is complete.

Step 7. For Bchan and Gchan, repeat steps 2 through 6.

Step 8. Convert binary vector to Hexadecimal vector and save it to Ciphertext hexadecimal file.

End

Algorithm 6: AES decryption (decipherment) algorithm

Input: AES ciphertext (*aesCiphertxt*), *AES decipherment key*(*aesDecKey*)

Output: plaintext

Begin

Step 1. Divide ciphertext into 128 bits blocks

Step 2. Divide each block into 4 by 4 square matrix as name $block_i$

***Corresponding author**
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

Step 3. *Initialize round for keys to 14 rounds as* Key_i □ *aesShrKey*

Step 4. *Add round key* □ *XOR* ($block_i$, Key_i)

Step 5. *While rounds between 1 to 14 repeat step 6 to step 9*

Step 6. *Cry out Inverse Bytes substation* ($block_i$)

Step 7. *Cry out Inverse shift rows* ($block_i$)

Step 8. *Cry out inverse Mix column* ($block_i$)

Step 9. *Add round key* □ *XOR* ($block_i$, Key_i)

Step 10. *Return* ***plaintext*** □ $block_i$

End

Algorithm 7: ECC decryption (decipherment) algorithm

Input: Ciphertext hexadecimal file contains AES ciphertext (*aesCiphertxt*), *receiver public key* (*rePubKey*)

Output: Plaintext

Begin

Step 1. *Split receiver public key (rePubKey) and AES ciphertext (aesCiphertxt) into 2 string vectors.*

Step 2. *Compute ECC share key*(*eccShrKey*) □ *rePubKey * sender private Key* (*sePrivKey*).

Step 3. *Compute AES decipherment key*(*aesDecKey*) □ *convert to 256bits key*(*eccShrKey*)

Step 4. *Obtain* ***plaintext*** □ *call algorithm 6:AES decipherment algorithm*(*aesCiphertxt, aesDecKey*)

Step 5. *Return* ***plaintext.***

End

When the received pixels that inserted into stego image are [PXinew, PXinew+1] = [74, 54] then di = |74 -54| = 20 From Table 1 obtain: Ri [UPi, LOi] = [31, 16], and tbits = floor (31 – 16 + 1)= 4 bits, then tdec = |20 – 16| = 4, i.e. the inserted message in binary is $(0100)_2$, this value saved in hexadecimal in ciphertext file.

RESULTS AND DISCUSSION

In the proposed technique, Python 3.7 is used to generate the ciphertext hexadecimal file. MATLAB 2019 can combine the features of both systems by calling the Python file in MATLAB 2019 and then performing PVD steganography on the ciphertext hexadecimal file. To test the proposed technique, four RGB images covering various features were used on a Dell laptop with an Intel Core i7 @2.5Gz and 8GB of RAM.

The ciphertext message with the size of 1,384,434 hexadecimal values (i.e., 5,537,736 bits) was inserted into the selected RGB images (real test sending the file, not generating random messages during the inserting process). There are two levels of evaluation for the proposed system. Using Elliptic Curve Cryptography(ECC) reduces key length and processing speed at the encipherment level, which is the first level. 256-bit elliptic curve cryptography is used [17] while 3072-bit non-elliptic curve cryptography is used. Elliptic curve algorithms use smaller keys than non-elliptic curve algorithms.

The figures (3–5) represent a 1024 × 1024 RGB cover–images and stego–images that result after ciphertext inserted into the cover–image and (i) to (iii) for each figure represent corresponding histograms.

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

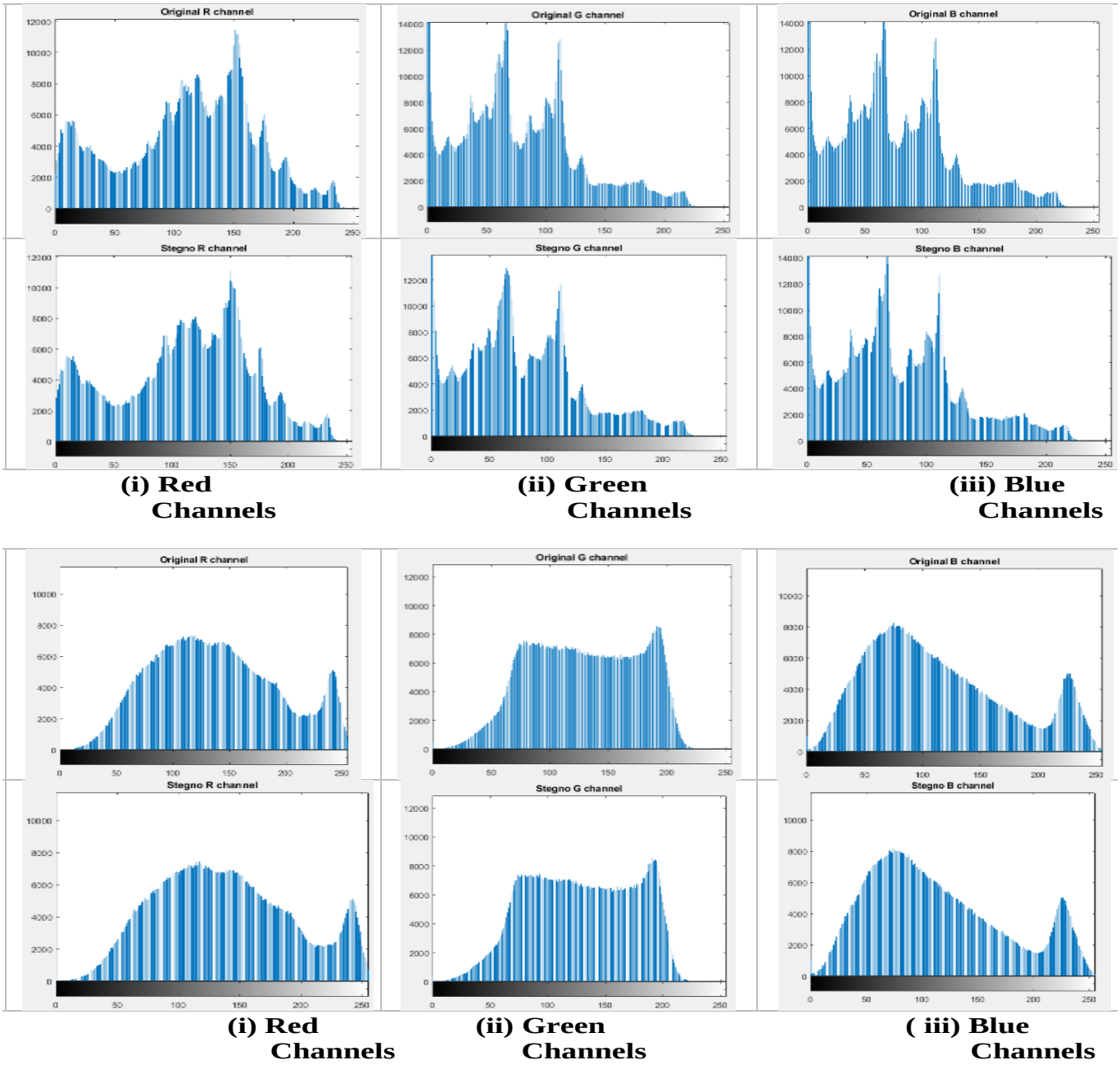

Figure 3: outcomes from the first experiment.

***Corresponding author**
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

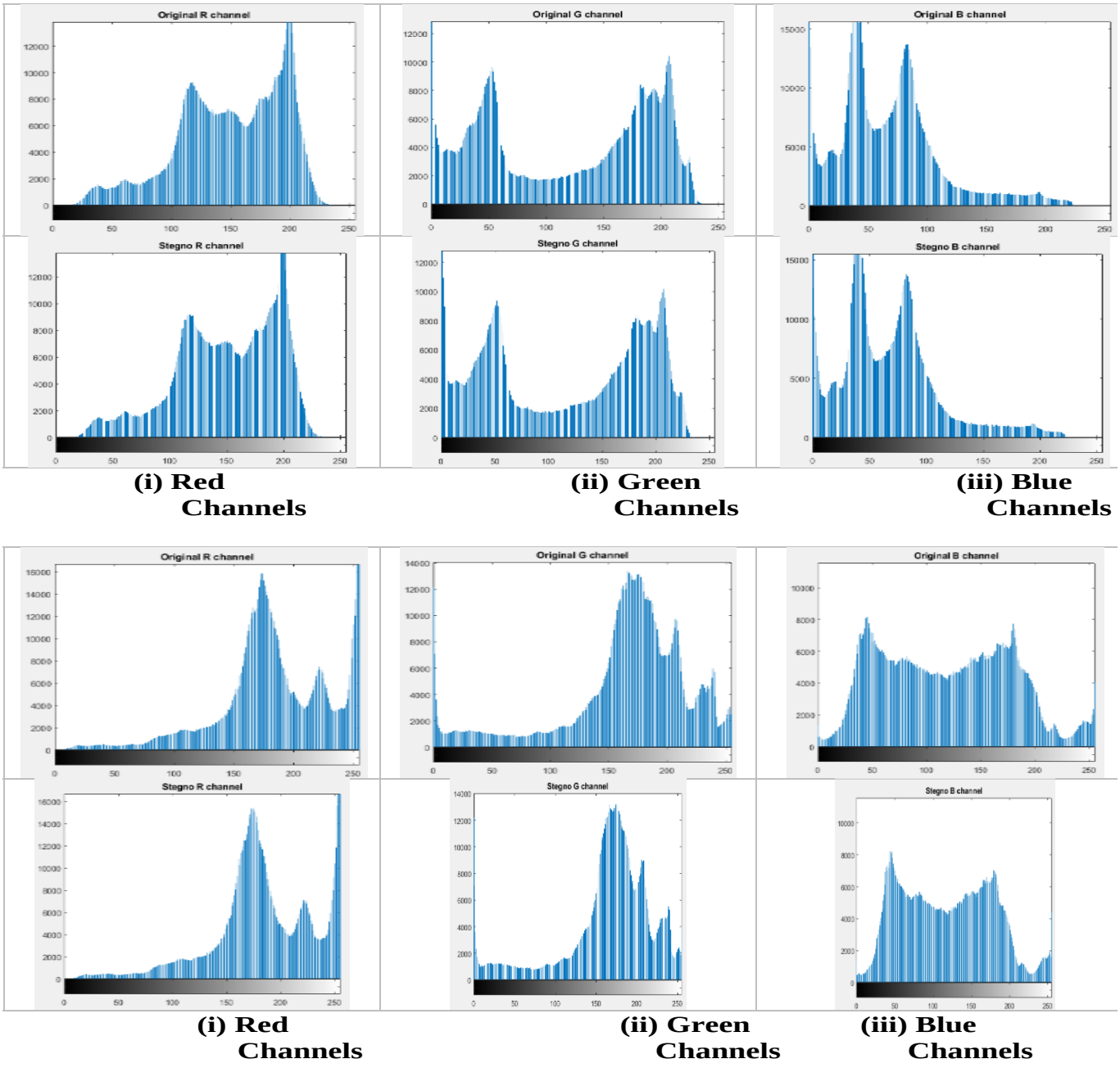

Figure 4: outcomes from the second experiment.

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

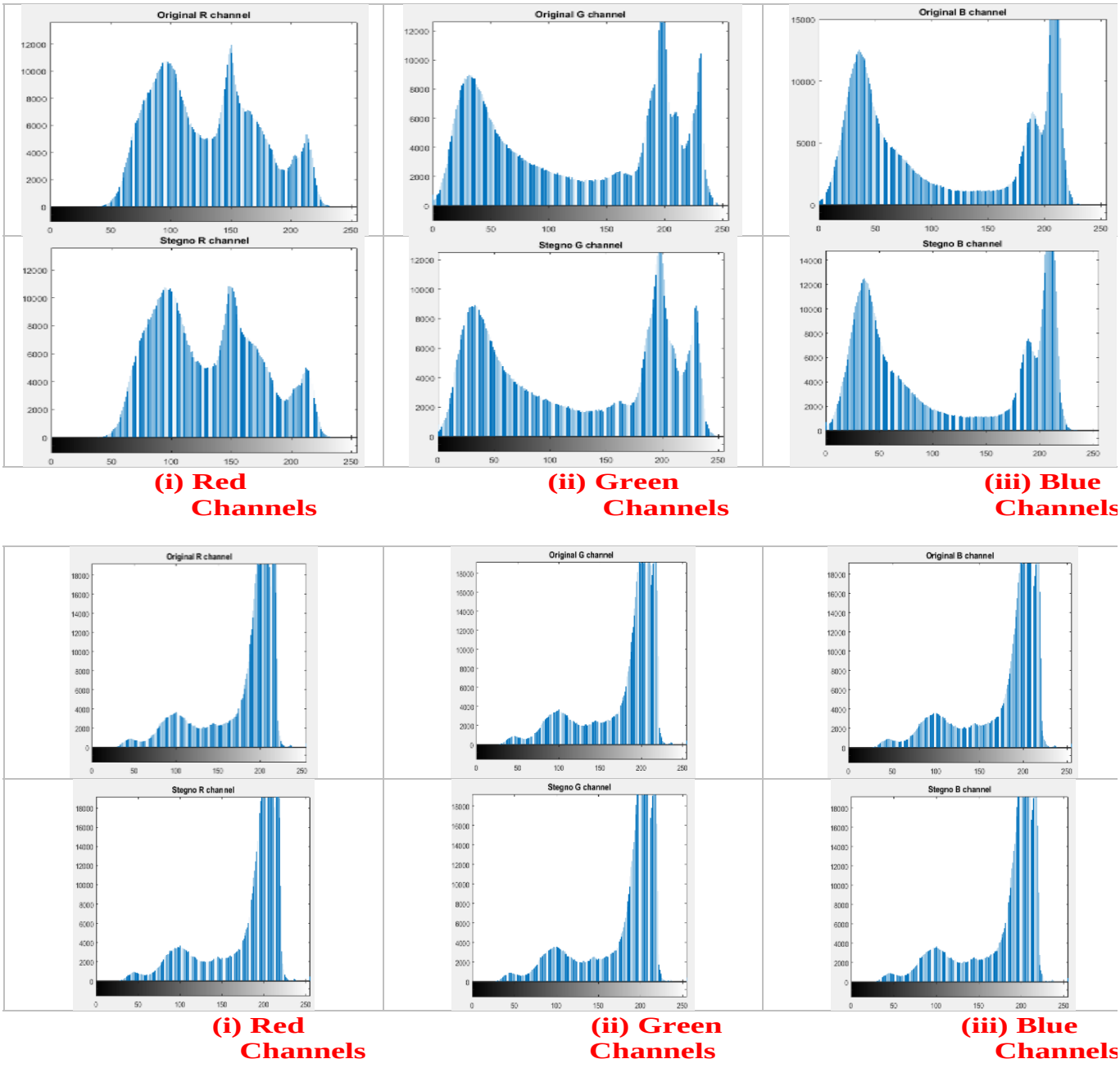

Figure 5: outcomes from the third experiment.

As a second level, embedding process measurements are measured, the most popular measurements being peak signal-to-noise ratio (PSNR) and the number of messages inserted (HC). This table compares the PSNR (dB) and total inserted bits of the proposed technique with that of four previous works using Pixel Value Differencing (PVD) with random generation of plain text for five standard images commonly used in steganography. According to table 2, the proposed technique produces a PSNR of 54.1 dB, while Wu D-C and Tsai W-H have a PSNR of 54.1 dB. According to Yang & Wang [19], Swain G [20, 21], Mandal & Das [22], respectively, the average noise level is 39.71dB, 41.09dB, 47.04dB, and 39.54dB. For the proposed method, there are 5537736 bits inserted, while the average PSNR for Wu D-C, Tsai W-H is . In [18], Yang & Wang [19], Swain G [20, 21], Mandal & Das [22], the bits are 1278313 bits, 196608 bits, 1374809 bits, and 1,278,313 bits, respectively. Therefore, the proposed method is higher by factor varied from 4.03 to 28.17, and has the greatest inserted (hidden) capacity when compared to other methods. Based on previous results, 1024 1024 3 RGB provided a substantial increase in clandestine message size with negligible effects on visual quality. The measured value of elapsed time for all encipherment/decipherment processes and also inserting/extracting processes. These results were tested on two Dell laptops with CPU Intel Core i7 @2.5Ghz and RAM 8GB (for sending and receiving) connected through the internet. The result shows that the processes that include the encipherment, inserting, extracting, and finally decipherment take between 200.9115 sec. (3minutes and 21 sec.) to 202.06 (3minutes and 23 sec) with an average of 201.43 sec (3 minutes 21.43 sec.) only. This means the proposed technique is fast processing time in spite of large image size (1024×1024×3) RGB images. With neglecting the time required to upload and download the sending and receiving stego–image

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

file. The results obtained show the proposed technique effective technique to get a very high combination of cryptographical and stenographical techniques with preserving the reasonable visual quality of stego-images where an attacker could not detect the inserted information.

CONCLUSION

The proposing technique shows a high-security encipherment algorithm using higher inserted bits size inside the Red, Green, and Blue channels of cover–image that results in very high-security levels for transferring the securing information over unsecured communication environments. By using Elliptic Curve (EC) algorithms that utilize smaller key length than their non-elliptic curve alternative to give a hexadecimal ciphertext that inserted to RGB cover – image three channels using Pixel Value Difference (PVD) with very high peak signal-to-noise ratio (PSNR) that result from insensible changes in the original image that make it very difficult attack with any prying.

REFERENCES

[l] Barakat, M., Eder, C., & Hanke, T. (2018). An Introduction to Cryptography. Timo Hanke at RWTH Aachen University, 1-145.
[2] Sateesh, G., Lakshmi, E.S., Ramanamma, M., Jairam, K., & Yeswanth, A. (2016). Assured data communication using cryptography and steganography. Int. J. Latest Technol. Eng. Manage. Appl. Sci, 5(3), 545-551.
[3] Bukhari, S., Arif, M.S., Anjum, M. R., & Dilbar, S. (2016). Enhancing security of images by Steganography and Cryptography techniques. In 2016 Sixth International Conference on Innovative Computing Technology (INTECH). Dublin, Ireland: IEEE. 531-534.
[4] Das, R., & Das, I. (2016). Secure data transfer in IoT environment: Adopting both cryptography and steganography techniques. In 2016 Second International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). Kolkata, India: IEEE. 296-301.
[5] Joshi, K., & Yadav, R. (2015). A new LSB-S image steganography method blend with Cryptography for secret communication. In 2015 Third International Conference on Image Information Processing (ICIIP). Waknaghat, India: IEEE, 86-90.
[6] Patel, N., & Meena, S. (2016). LSB based image steganography using dynamic key cryptography. In 2016 International Conference on Emerging Trends in Communication Technologies (ETCT) Dehradun, India: IEEE, 1-5.
[7] Eyssa, A.A., Abdelsamie, F.E., & Abdelnaiem, A.E. (2020). An Efficient Image Steganography Approach over Wireless Communication System. Wireless Personal Communications, 110(1), 321-337.
[8] Aditya Kumar Sahu, Gandharba Swain, E. Suresh Babu (2018). Digital Image Steganography Using Bit Flipping. CYBERNETICS AND INFORMATION TECHNOLOGIES. Volume 18, No.1. Sofia. Print ISSN: 1311-9702; Online ISSN: 1314-4081
[9] Aditya Kumar Sahu &Gandharba Swain (2018). Pixel Overlapping Image Steganography Using PVD and Modulus Function. 3D Research Center, Kwangwoon University and Springer-Verlag GmbH Germany, part of Springer Nature. https://doi.org/10.1007/s13319-018-0188-5.
[10] Aditya Kumar Sahu . Gandharba Swain (2019). An Optimal Information Hiding Approach Based on Pixel Value Differencing and Modulus Function. Springer Science + Business Media, LLC, part of Springer Nature. Wireless Personal Communications 108:159–174.
[l1] Reshma Sonar & Gandharba Swain (2022). A hybrid steganography technique based on RR, AQVD, and QVC, Information Security Journal: A Global Perspective, 31:4, 479-498, DOI: 10.1080/19393555.2021.1912219
[12] Stallings, W., & Tahiliani, M.P. (2014). Cryptography and network security: principles and practice (6 th ed.), Prentice Hall.
[13] Hussain, M., & Hussain, M. (2013). A survey of image steganography techniques. International. Journal of Advanced Science and Technology, 54.
[14] Hankerson, D., Menezes, A. J., & Vanstone, S. (2006). Guide to elliptic curve cryptography. Springer Science & Business Media.
[15] Prasad, S., & Pal, A.K. (2017). An RGB colour image steganography scheme using overlapping block-based pixel-value differencing. Royal Society open science, 4(4), 161066.
[16] Seethalakshmi, K.S., Usha, B.A., & Sangeetha, K.N. (2016). Security enhancement in image steganography using neural networks and visual cryptography. In 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS) Bangalore, India: IEEE, 396-403.
[17] Hitchcock, Y., Montague, P., Carter, G., & Dawson, E. (2004). The efficiency of solving multiple discrete logarithm problems and the implications for the security of fixed elliptic curves. International Journal of Information Security, 3(2), 86-98.
[18] Wu, D.C., & Tsai, W.H. (2003). A steganographic method for images by pixel-value differencing. Pattern Recognition Letters, 24(9-10), 1613-1626.
[19] Yang, C. Y., & Wang, W. F. (2015). Block-based colour image steganography using smart pixel-adjustment. In Genetic and Evolutionary Computing. Cham, India: Springer, 145-154.
[20] Swain, G. (2016). Adaptive pixel value differencing steganography using both vertical and horizontal edges. Multimedia Tools and Applications, 75(21), 13541-13556.

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq

[21] Swain G. (2019). Very high capacity image steganography technique using quotient value differencing and LSB substitution. Arabian journal for science and engineering, 44(4), 2995-3004.
[22] Mandal JK, Das D. 2012 Steganography using adaptive pixel value differencing (APVD) of gray images through exclusion of overflow/underflow. In 2nd Int. Conf. on Computer Science, Engineering and Applications (CCSEA-2012), Delhi, India.

*Corresponding author
Muthanna Jabbar Abdulredhi,
Collage of Business Informatics, University of Information Technology and Communications, Baghdad, Iraq
e-mail: muthanna.jabbar@uoitc.edu.iq