

RESEARCH ARTICLE

Enhancing Speed and Imperceptibility in Watermarking Systems by Leveraging Galois Field Tables

YASMIN ALAA HASSAN^{1,2} AND ABDUL MONEM S. RAHMA³¹Department of Computer Science, College of Science, University of Baghdad, Baghdad 10071, Iraq²Department of Computer Science, University of Technology, Baghdad 10066, Iraq³Department of Computer Science, Al-Maarif University College, Al Anbar 31001, Iraq

Corresponding author: Yasmin Alaa Hassan (Yasmin.a@sc.uobaghdad.edu.iq)

ABSTRACT The use of watermarking techniques, which incorporate invisible information into multimedia files, is crucial for securing digital content. Watermarking research still faces difficulties in striking a balance between speed, robustness, and imperceptibility. This research suggests a novel method for improving the speed and imperceptibility of watermarking systems that makes use of Galois Field (GF) tables. The advantage of creating GF tables is that they implement GF operations well, which greatly accelerates the watermarking processes. It is possible to reduce computational complexity and speed up execution times by previously calculating and storing GF tables. The proposed method is ideal for applications with strict time limitations since this performance improvement allows real-time watermarking. Additionally, imperceptibility is enhanced by the use of well-chosen irreducible polynomials in GF -based watermarking algorithms. The distribution and modulation of watermark bits are influenced by the selection of irreducible polynomials, allowing for optimal embedding with the least amount of visual distortions. By using this selection technique, the host media's quality and integrity are preserved because the embedded watermark is kept transparent or undetectable to human observers. In conclusion, this study provides a novel approach that makes use of Galois Field tables and selective irreducible polynomials to improve the speed and imperceptibility of watermarking systems. GF tables are used to increase computing effectiveness and enable real-time processing, whereas irreducible polynomials are selected to maximize imperceptibility. This strategy creates new opportunities for the creation of watermarking methods that effectively safeguard digital content without reducing user experience or system performance.

INDEX TERMS Security, Galois field, irreducible polynomial, robustness, addition table, multiplication table, inverse table, image watermark.

I. INTRODUCTION

In recent years, the rapid advancements in science and technology, particularly in the realm of the Internet and multimedia technology, have revolutionized the way of exchanging and disseminating information. These technological advancements have significantly impacted daily lives. However, with the widespread use of the Internet, concerns about unauthorized alterations and illegal copying of digital content during online transmission have escalated. Consequently, safeguarding copyright and preserving information security

have become paramount [1], [2]. Digital image watermarking is a technique employed to embed information, often in the form of imperceptible marks or signatures, directly into an image [3]. The primary goal is to augment the image with additional data without significantly altering its visual appearance. This process holds immense significance in various domains, including copyright protection, authentication, and content integrity verification. The imperceptible nature of these digital marks, often invisible to the human eye, transforms them into powerful tools, serving as digital signatures or identifiers within the image. This imperceptibility enhances their efficacy, particularly in applications requiring copyright enforcement and content authentication [4]. The

The associate editor coordinating the review of this manuscript and approving it for publication was Rahim Rahmani^{id}.

significance of image watermarking becomes even more pronounced in the age of widespread digital distribution and sharing, where safeguarding digital content is of paramount importance. In the realm of image watermarking, despite notable advancements, there are persistent challenges [5]. Robustness against various attacks and the delicate balance between preserving perceptual quality and meeting stringent time constraints remain key hurdles. To address this issue, significant advancements have been made in the field of digital watermarking. Thus, digital watermarking stands as a crucial tool in the ongoing efforts to secure digital content and mitigate the risks associated with unauthorized use and copyright infringement [6]. The practice of safeguarding media content frequently involves the inconspicuous embedding of hidden marks within the host media, a technique commonly known as color image watermarking [3], [7]. Galois Field with 2^n elements, is a finite mathematical construct with distinctive properties that make it well-suited for watermarking applications. This field can be represented as a n -bit binary system and offers a structured framework for encoding and embedding information into digital media [8]. The core concept of using $GF(2^n)$ multiplication tables in watermarking lies in exploiting the mathematical relationships within this finite field for the watermarking process. This approach provides a robust and secure means of protecting digital content, allowing content owners to assert their ownership and deter unauthorized use [9]. In a related study, a blind color image watermarking system with great performance in the spatial domain has been given in [10] by blending the benefits of a spatial-domain watermarking scheme and a frequency-domain one. The colored photos in [11] on the Raspberry Pi (RPi) platform, a parallel robust watermarking approach that uses the Quaternion Legendre-Fourier Moment (QLFM) in polar coordinates is built. A binary Arnold scrambled picture is placed in the host image. The Raspberry Pi model 4B is used to implement and test the watermarking algorithm. [12] describes a brand-new digital watermarking method for color photographs that relies on the discrete cosine transform (DCT) and a triple-byte nonlinear block cipher. Further [13] proposed a new digital watermarking method for color images based on a triple-byte nonlinear block cipher and the discrete cosine transform (DCT). Based on the Galois ring (GR 23,8), a triple-byte nonlinear part of a block cipher, specifically a 24×24 substitution box (S-box), was created. Then, this encryption was used in the watermarking procedure by being divided into three bytes and applying each byte separately to the red (R), green (G), and blue (B) channels.

Critical limitations observed in current watermarking techniques include: vulnerability to common image processing attacks, speed, image quality degradation, and insufficient imperceptibility.

These limitations have impeded the practical application of watermarking in real-world scenarios. In light of these problems, in this research we developed a novel algorithm that leverages the power of GF tables for pixel value manipu-

lation, ensuring a rapid embedding process. This algorithm represents a significant enhancement in terms of speed, imperceptibility, and robustness for embedded watermarks, successfully overcoming the limitations discussed earlier. The evaluation results underscore its efficacy, with a peak signal-to-noise ratio (PSNR) reaching 56.82. Additionally, the algorithm achieves an embedding time of just 0.03 seconds. The normalized correlation (NC) of the extracted watermark is 0.995.

The remainder of this paper is organized in the following manner: Section I-A describes the finite field fundamentals and their operations. Section I-B discusses image watermarking using the Galois Field tables (GF). Section II illustrates the methodology and the proposed system. Section III presents the results and discussions. Finally, section IV presents the conclusion and future works in the field of watermarking.

A. AN OVERVIEW OF THE FINITE FIELD

Finite fields, also known as Galois fields (GF), are mathematical structures that have important properties and applications in various fields, including cryptography, error correction codes, and digital signal processing [13]. Finite fields play a crucial role in modern cryptography algorithms. Public key cryptography algorithms like elliptic curve cryptography (ECC) and RSA utilize finite fields for key generation, encryption, and decryption operations. The algebraic properties of finite fields contribute to the security and efficiency of these cryptographic schemes [14]. $GF(p^n)$ is a common abbreviation for the finite field of order p^n . In honor of the mathematician who first explored finite fields, GFs are so termed. The set of numbers $(0, 1, \dots, p-1)$ and the arithmetic operations modulo p are known as the finite field of order p , or $GF(p)$. As a result, arithmetic mod p can be used to define the finite fields of order p . When n is greater than 1, arithmetic over polynomials can be used to define the finite fields of order p^n [15]. Employing larger GF tables leads to an increase in the number of possible combinations or elements in the field. This leads to the creation of a larger key space, thereby making it more challenging for attackers to break the encryption or uncover the original data; this implies enhanced security.

Further, the use of larger GF tables also improves resistance to attacks [16]. These tables add complexity and non-linearity to cryptographic operations. This enhances resistance attacks, such as differential and linear cryptanalysis, algebraic attacks, and brute-force attacks [17]. GF arithmetic enables efficient operations such as addition, subtraction, multiplication, and division. This property leads to efficient arithmetic operations, as it enables the selection of appropriate GF tables. Moreover, cryptographic algorithms can be optimized to achieve faster computations while maintaining security [18].

A set of elements with the binary operations addition and multiplication make up a field F , sometimes written as " $F, +, *$ " [19]. If operations such as addition, subtraction,

multiplication, and division can be performed without deserting the set, then the field is a set [1].

When prime $(p) = 2$, the finite field with GF elements is known as $GF(p^n)$ and is also referred to as the GF . It is common to write the elements of $GF(2^n)$ as binary numbers [20].

Standard integer arithmetic and arithmetic in a finite field are two separate things. The finite field has a finite number of elements, and every action carried out there produces an element of that field [1].

1) ADDITION

The coefficients of the corresponding powers in the polynomial representations of the field elements are added to accomplish addition in a finite field. This addition takes place in $GF(2)$, which is also known as modulo 2, where the sum of 1 and 1 equals 0. Thus, addition and subtraction operations in a finite field $GF(2^n)$ can be equivalently represented as exclusive-or operations on the n -bit representations of field elements [10].

2) MULTIPLICATION

In a finite field, multiplication is more difficult than addition since it includes multiplying the two polynomials that represent the elements and adding like powers of x to the outcome. It is reduced modulo an irreducible polynomial $m(x)$ of degree n if the multiplication produces a polynomial with a degree higher than $n-1$. In this reduction, the polynomial is divided by $m(x)$, and the remainder is kept. An irreducible polynomial $m(x)$ over a field F cannot be expressed as the product of two polynomials, both over F , and both of lower degrees than $m(x)$ (neither of which has a degree of zero) [11].

B. IMAGE WATERMARKING USING GALOIS FIELDS

Image watermarking using GF s is a technique that enables the embedding of hidden information, known as a watermark, into digital images. GF provides a mathematical structure that enables secure and efficient data manipulation within a finite set of elements [12], [15]. In image watermarking, the GF is used to divide the watermark data into smaller elements, which are then embedded into the image pixels. A variety of operations, including addition, multiplication, and inversion, can be carried out on the image and watermark data by modeling it as elements of the GF [21]. Computational efficiency in Galois field arithmetic arises from reduced operand size, simplified operations within a restricted set, utilization of bitwise operations, potential for parallelism, and resource optimization. The finite field structure allows for faster and more compact computations, particularly beneficial in watermarking applications where speed and resource efficiency are paramount. Quantitative assessment involves comparing execution times of key operations in Galois field arithmetic against standard arithmetic, providing insights into efficiency gains [15], [22], [23]. Using GF s, image watermarking entails adjusting image pixels to include a watermark while balancing resilience and imperceptibility. In order to

achieve this balance, GF operations are essential. It has been used for tamper detection in digital images, content authentication, and copyright protection [24]. It is important to note that the specific implementation and techniques used in image watermarking using GF may vary depending on the irreducible polynomial used to construct the tables and algorithms employed. Researchers and practitioners often tailor the watermarking process to meet specific requirements such as robustness, imperceptibility, capacity, security and optimize performance [19].

II. METHODOLOGY

In this section, a comprehensive account of the research methodology is presented. Subsections include the selection of the dataset, the application of evaluation metrics, and a thorough exploration of the proposed watermarking system.

A. DATASET

For our experiments, well-established standard test images widely recognized within the field of image processing have been selected. Specifically, Lena, Pepper, and Parrot, which have a history of use in evaluating various image processing techniques and they are widely used in this field to be as benchmark for comparison purpose. These images, the cover image (1000×1000 resolution) were chosen for their diverse content, making them valuable benchmarks for assessing watermarking algorithms, and two watermark images (64×64 resolution) and (225×225 resolution), of type (.png). The watermark images were black-and-white and colored image. The inclusion of these standard images is vital as it allows us to gauge our algorithm's performance against established references.

Additionally, the utilization of the Break Our Watermarking System (BOWS) dataset, comprising 10,000 high-resolution cover images. The choice of this dataset for evaluation aligns with our research's focus on real-world applicability, allowing us to assess the watermarking process under conditions relevant to practical scenarios.

B. EVALUATION METRICS

Our selection of evaluation metrics, including Peak Signal-to-Noise Ratio (PSNR) and Normalized Correlation (NC), is based on their relevance in quantifying the quality of watermarked images. PSNR, expressed in decibels (dB), provides a measure of the fidelity of watermarked images in comparison to their original counterparts. It quantifies the ratio between the maximum pixel value and the noise level introduced during watermarking, providing insight into the image quality. The PSNR formula is defined as:

$$PSNR = 10 \times \log_{10} \left(\frac{MAX^2}{MSE} \right) \quad (1)$$

where:

- MAX represents the maximum pixel value (e.g., 255 for 8-bit images).

- MSE stands for Mean Squared Error, which quantifies the difference between the original and watermarked images pixel by pixel.

Furthermore, Normalized Correlation (NC) is employed to assess the degree of similarity or correlation between the original and watermarked images. With NC values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation) and 0 indicating no correlation, it serves as a valuable metric for evaluating watermarking quality.

The formula for NC calculation is as follows:

$$NC = \frac{\sum (I_1 - I'_1)(I_2 - I'_2)}{\sqrt{(\sum (I_1 - I'_1)^2) \sum (I_2 - I'_2)^2}} \quad (2)$$

where:

- I_1 and I_2 are the two signals or images being compared.
- I'_1 and I'_2 are the means (averages) of I_1 and I_2 respectively.

The Structural Similarity Index (SSIM) is a metric used to measure the similarity between two images. It is designed to assess the perceived quality of the images by taking into account various aspects of human visual perception. SSIM considers three main components: luminance, contrast, and structure. The resulting index ranges from -1 to 1, where 1 indicates perfect similarity.

C. THE PROPOSED SYSTEM

To calculate the addition of two numbers in finite fields, specific addition tables must be used, and to calculate the product of two numbers using the multiplication function of polynomials—which differ from algebraic multiplication operation—the irreducible polynomial of specified $GF(2^n)$ must be used.

First, the addition table is constructed by performing addition operations on all possible combinations of elements in $GF(2^n)$. Each element in the field is represented by a polynomial with coefficients in $GF(2)$, and the addition operation is implemented by adding the coefficients for the corresponding powers of the polynomials. The resulting addition table showcases the outcomes of these additional operations.

Similarly, the multiplication table is generated by multiplying the polynomials representing the elements in $GF(2^4)$. The outcome of the multiplication operation is reduced modulo a degree four irreducible polynomial by grouping like powers of the polynomial together. As a result, the degree of the resulting polynomial is guaranteed to stay within the limits of the field. The results of these multiplication operations are explained by the multiplication table. There are numerous irreducible polynomials used as modulus values and each one produces a different table. This may thwart the attempts of unauthorized user attacks and increase the level of security. The generated multiplication table for multiplying values in the range of 1 to $n-1$ exclude zero (the first row and first column in the generated multiplication table). The main reason for using the addition and multiplication tables is to reduce the time of execution. The algorithms of addition and multiplication using $GF(2^n)$ are provided below.

D. ADDITION TABLE CONSTRUCTION IN $GF(2^n)$

Algorithm: Addition Table Construction on $GF(2^n)$

Input: n (degree of GF), field_size (2^n)

Output: Addition table in $GF(2^n)$ of size field_size x field_size

1. Begin

2. Initialize an empty 2D array “addition_table” of size field_size x field_size.

3. Initialize an array “element” with values [0, 1, 2, ..., field_size-1].

4. Create an array “bit_masks” to represent all possible bit masks of size n .

5. For each element x in elements:

5.1. For each bit_mask in bit_masks:

5.1.1. Compute the sum result as x XOR bit_mask.

5.1.2. Assign the result to addition_table[x][result].

6. Return the addition_table.

7. End.

E. MULTIPLICATION TABLE CONSTRUCTION IN $GF(2^n)$

Algorithm: Multiplication Table Construction on $GF(2^n)$

Input: n (degree of GF), field_size (2^n)

Output: Multiplication table in $GF(2^n)$ of size field_size x field_size

1. Begin

2. Initialize an empty 2D array “multiplication_table” of size field_size x field_size.

3. Initialize an array “elements” with values [0, 1, 2, ..., field_size-1].

4. For each element x in elements:

4.1. For each element y in elements:

4.1.1. Compute the product result as the product of x and y in $GF(2^n)$ using the GF multiplication operation, which involves polynomial multiplication.

4.1.2. Perform a modulo operation with respect to an irreducible polynomial $g(x)$ specific to the $GF(2^n)$ field. This ensures that the result stays within the field and conforms to its properties.

4.1.3. Assign the result to multiplication_table[x][y].

5. Return the multiplication_table.

6. End.

F. ADDITION AND MULTIPLICATIVE INVERSE TABLE IN $GF(2^n)$

The function of addition and multiplicative inverse tables in GF s, more specifically in the field of $GF(2^n)$, is crucial. The addition and multiplicative inverse operations inside the GF are fully represented in these tables. The watermarking process is made more robust against attacks such as geometric changes, cropping, filtering, and compression thanks to the addition and multiplicative inverse tables, which increase complexity and non-linearity. The procedures carried out utilizing the tables make sure that the watermark is securely embedded and that it is challenging to remove or tamper with without causing obvious artifacts. The additive inverse of a GF is shown as zeroes in the addition table. This suggests that the result will be 0 if you combine an element in the field with its equivalent additive inverse. Given that subtracting from the GF is identical to adding with the additive inverse, this characteristic is essential. Similarly, the multiplicative inverse of a GF is shown as one in the multiplicative table. The identity element, which is generally represented as one, is obtained by multiplying each element in the table by its corresponding multiplicative inverse. Due to the fact that division is the same as multiplication with the multiplicative inverse, this characteristic permit division within the GF . By utilizing the zeroes in the addition table and the ones in the multiplicative table, operations within the GF can be efficiently performed, thereby enabling various applications, such as error correction codes, cryptography, and image watermarking. These applications make it easier to implement operations within GF s and to do calculations efficiently, which helps to create safe and reliable systems. The following is a general algorithm for computing the addition and multiplicative inverses in different GF s ($GF(2^3)$, $GF(2^4)$, $GF(2^5)$, and $GF(2^6)$) tables.

G. THE WATERMARK EMBEDDING PROCESS USING THE $GF(2^4)$ MULTIPLICATION TABLE

The watermark can withstand common image alterations thanks to the use of $GF(2^n)$ as a transform, and it is imperceptible because the watermarked image maintains its visual quality. However, it is important to consider the trade-off between the strength of the watermark and its visibility in order to maintain a balance between effective protection and preserving the integrity of the image. The created $GF(2^4)$ field is additionally used to put a watermark on an image. The watermarking approach divides the watermark into three bytes, each of which corresponds to the red (R), green (G), and blue (B) channels of the image. This division is made possible by the features of $GF(2^4)$ and effectively insert the watermark data within the image.

The outcomes of these operations show how $GF(2^4)$ can be used to build tables, carry out arithmetic calculations, and insert watermarks in digital images, among other cryptographic and data manipulation activities. The findings analysis offers important insights into the capabilities and

Algorithm: Compute the addition and multiplicative inverse tables in different Galois fields

Input: Addition and multiplication tables in $GF(2^3)$, $GF(2^4)$, $GF(2^5)$, and $GF(2^6)$

Output: addition and multiplicative inverse tables in $GF(2^3)$, $GF(2^4)$, $GF(2^5)$, and $GF(2^6)$

1. Begin

2. Define the Galois field tables for addition and multiplication.

3. Create a function to compute the multiplicative inverse table:

a. Initialize an empty dictionary for the multiplicative inverse table.

b. Iterate each element in the multiplication table.

c. Find the index of the element's multiplicative inverse (i.e., the element multiplied by its inverse equals 1).

d. Add the element and its inverse to the multiplicative inverse table dictionary.

e. Return the multiplicative inverse table.

4. Create a function to compute the addition inverse table:

a. Initialize an empty dictionary for the addition inverse table.

b. Iterate over each element in the addition table.

c. Find the index of the element's addition inverse (i.e., the element added to its inverse equals 0).

d. Add the element and its inverse to the addition inverse table dictionary.

e. Return the addition inverse table.

5. Call the function in step (3) for each multiplication Galois field table to get the multiplicative inverse tables.

6. Call the function in step (4) for the addition Galois field table to get the addition inverse table.

7. Print the computed inverses for each Galois field.

8. End.

performance of $GF(2^4)$ in applications for image watermarking.

The watermarking algorithm used in the proposed scheme is based on Galois Field ($GF(2^n)$) arithmetic. This algorithm converts a watermark image into values corresponding to the elements of the $GF(2^n)$ table and embeds these values into a cover image using XOR operations. This approach is often used in watermarking to enhance the security and robustness of the embedded watermark. A $GF(2^4)$ was utilized in this study as an example of transformation to insert a watermark in the corners of the cover image while maintaining the original cover image's quality. Bitwise XOR operations are used to combine the watermark's matching bits with the pixel values to include it as a copyright indicator or unique identifier. Because the watermark is integrated into the image during this process, it cannot be removed or altered. The $GF(2^4)$ table can be used to map each pixel to a numerical value within the field, which can then be converted to

binary. This binary representation is utilized for performing watermarking operations via XOR operation. This enables the seamless integration of a watermark, typically in the form of a binary sequence, into the image while minimizing visual distortion.

Algorithm: Watermark Embedding Process

Input: multiplication table of $GF(2^4)$, cover image, watermark image

Output: Watermarked image

1. Begin

2. Load the cover image and watermark image.
 3. Define the $GF(2^4)$ table containing the values for the desired mapping.
 4. Create a copy of the cover image as the watermarked image.
 5. Resize the watermark image to match the desired size if needed.
 6. Convert the pixel values of the watermark image to their corresponding values in the $GF(2^4)$ table:
 - Iterate over the pixels of the watermark image.
 - Extract the lower 4 bits (y) and upper 4 bits (x) of each pixel value.
 - Map the values x and y to their corresponding values in the $GF(2^4)$ table using $gf_table[x][y]$.
 - Update the corresponding pixels in the converted watermark image with the mapped values.
 7. Embed the converted watermark in the corners of the cover image:
 - Iterate over the pixels of the converted watermark image.
 - For each pixel in the converted watermark image:
 - Retrieve the corresponding pixel in the watermarked image.
 - Perform a blending operation using XOR between the watermarked image pixel and the converted watermark pixel using the $GF(2^4)$ table.
 - Update the corresponding pixel in the watermarked image with the blended value.
 8. The watermarked image now contains the embedded watermark in the corners.
 9. Display or save the watermarked image.
 12. **End.**
-

This algorithm ensures that the watermark is embedded in a manner that minimizes its visibility to the human eye while utilizing the $GF(2^4)$ table for robustness against image processing operations. A $GF(2^4)$ multiplication table has been employed in this investigation along with the irreducible polynomial $(x^4 + x + 1)$. Another multiplication table will be produced when this irreducible polynomial is changed to $(x^4 + x^3 + 1)$, which will improve the outcomes.

H. THE WATERMARK EXTRACTION PROCESS USING THE $GF(2^4)$ MULTIPLICATIVE INVERSE TABLE

In this section, the process of extracting watermarks embedded within an image using the Galois Field $GF(2^4)$ multiplicative inverse table has been presented. The use of Galois field arithmetic provides a structured mathematical framework, and the inverse table becomes a crucial element in efficiently reversing the watermark embedding process. The input is the watermarked image and the multiplicative inverse table, to obtain the extracted watermark image.

Algorithm: Watermark Extraction Process

Input: Watermarked Image, Cover Image, $GF(2^4)$ Multiplicative Inverse Table

Output: Extracted Watermark Image

1- Begin

- 2- load the watermarked image and the cover image
- 2- XOR the two images to obtain the watermark (the values after converting to $GF(2^4)$ multiplication table)
- 3- Convert the resulted extracted watermark to $GF(2^4)$ multiplicative inverse table
- 4- Shift the values to the left 4 bits (as opposite as the converting to GF multiplication table)
- 5- Save the resulted watermark image

6-End

In summary, the algorithm focuses on reversing the operations applied during the embedding process, including XOR operations, conversion to $GF(2^4)$ table values, and bit-shifting. The final result is the extracted watermark.

III. RESULTS AND DISCUSSIONS

The following account presents the results obtained from constructing the addition and multiplication tables in the GF , specifically $GF(2^4)$, and utilizing this field for embedding a watermark on an image. The other addition and multiplication tables are in the appendix section. The GF arithmetic enables secure and efficient data manipulation within a finite field structure. The experiments were conducted using Python version 3.8 on a Windows 10 system with a Core i7 CPU and 16 GB of RAM. This setup was chosen to ensure efficiency and accuracy in our experimentation process.

Table 1 displays the multiplication tables for $GF(2^4)$. To enhance accessibility and readability, addition and multiplication tables for $GF(2^3)$, $GF(2^5)$, and $GF(2^6)$ are provided in the appendix section.

A. MULTIPLICATION TABLE IN $GF(2^4)$ WITH IRREDUCIBLE POLYNOMIAL $x^4 + x + 1$

See Table 1.







B. IMAGE WATERMARK USING $GF(2^4)$ MULTIPLICATION TABLE

An invisible mark or signal is embedded into digital images using the image watermarking process in order to protect

TABLE 1. Multiplication table in $GF(2^4)$.

[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 5],
[2, 4, 6, 8, 10, 12, 14, 3, 1, 7, 5, 11, 9, 15, 13],
[3, 6, 5, 12, 15, 10, 9, 11, 8, 13, 14, 7, 4, 1, 2],
[4, 8, 12, 3, 7, 11, 15, 6, 2, 14, 10, 5, 1, 13, 9],
[5, 10, 15, 7, 2, 13, 8, 14, 11, 4, 1, 9, 12, 3, 6],
[6, 12, 10, 11, 13, 7, 1, 5, 3, 9, 15, 14, 8, 2, 4],
[7, 14, 9, 15, 8, 1, 6, 13, 10, 3, 4, 2, 5, 12, 11],
[8, 3, 11, 6, 14, 5, 13, 12, 4, 15, 7, 10, 2, 9, 1],
[9, 1, 8, 2, 11, 3, 10, 4, 13, 5, 12, 6, 15, 7, 14],
[10, 7, 13, 14, 4, 9, 3, 15, 5, 8, 2, 1, 11, 6, 12],
[11, 5, 14, 10, 1, 15, 4, 7, 12, 2, 9, 13, 6, 8, 3],
[12, 11, 7, 5, 9, 14, 2, 10, 6, 1, 13, 15, 3, 4, 8],
[13, 9, 4, 1, 12, 8, 5, 2, 15, 11, 6, 3, 14, 10, 7],
[14, 15, 1, 13, 3, 2, 12, 9, 7, 6, 8, 4, 10, 11, 5],
[15, 13, 2, 9, 6, 4, 11, 1, 14, 12, 3, 8, 7, 5, 10]]

TABLE 2. The PSNR, SSIM, and execution time with irreducible polynomial $x^4 + x + 1$ and black-and-white watermark image.

Cover image	Watermark image	PSNR	SSIM	Execution time
		50.7292	0.954	0.0338
		50.8468	0.985	0.0313
		50.7944	0.940	0.0360

intellectual property, authenticate users, and identify the content of the image. Utilizing $GF(2^4)$, one of the mathematical characteristics of GFs , as a transformation mechanism is one way to accomplish this. The $GF(2^4)$ table with the irreducible polynomial $x^4 + x + 1$ has been used in this experiment.

The application of the proposed method to a colored watermark is feasible, and the corresponding outcomes are delineated in Table 3.

TABLE 3. The PSNR, SSIM, and execution time with irreducible polynomial $x^4 + x + 1$ and colored watermark image.







Cover image	Watermark image	PSNR	SSIM	Execution time
		56.4511	0.952	0.0433
		56.7354	0.979	0.0436
		56.9309	0.938	0.0457

TABLE 4. Multiplication table in $GF(2^4)$.

[[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15],
[2, 4, 6, 8, 10, 12, 14, 9, 11, 13, 15, 1, 3, 5, 7],
[3, 6, 5, 12, 15, 10, 9, 1, 2, 7, 4, 13, 14, 11, 8],
[4, 8, 12, 9, 13, 1, 5, 11, 15, 3, 7, 2, 6, 10, 14],
[5, 10, 15, 13, 8, 7, 2, 3, 6, 9, 12, 14, 11, 4, 1],
[6, 12, 10, 1, 7, 13, 11, 2, 4, 14, 8, 3, 5, 15, 9],
[7, 14, 9, 5, 2, 11, 12, 10, 13, 4, 3, 15, 8, 1, 6],
[8, 9, 1, 11, 3, 2, 10, 15, 7, 6, 14, 4, 12, 13, 5],
[9, 11, 2, 15, 6, 4, 13, 7, 14, 12, 5, 8, 1, 3, 10],
[10, 13, 7, 3, 9, 14, 4, 6, 12, 11, 1, 5, 15, 8, 2],
[11, 15, 4, 7, 12, 8, 3, 14, 5, 1, 10, 9, 2, 6, 13],
[12, 1, 13, 2, 14, 3, 15, 4, 8, 5, 9, 6, 10, 7, 11],
[13, 3, 14, 6, 11, 5, 8, 12, 1, 15, 2, 10, 7, 9, 4],
[14, 5, 11, 10, 4, 15, 1, 13, 3, 8, 6, 7, 9, 2, 12],
[15, 7, 8, 14, 1, 9, 6, 5, 10, 2, 13, 11, 4, 12, 3]]

Table 4 illustrates the results of image watermarking when using multiplication table of $GF(2^4)$ observed by table 4 with the second irreducible polynomial ($x^4 + x^3 + 1$)

C. MULTIPLICATION TABLE IN $GF(2^4)$ WITH IRREDUCIBLE POLYNOMIAL $x^4 + x^3 + 1$

The results obtained with colored images as watermarks demonstrate superior performance compared to black and white counterparts. This improvement is attributed to the enhanced complexity and information content present in

TABLE 5. the PSNR, SSIM, and execution time with irreducible polynomial $x^4 + x^3 + 1$ and black and white watermark image.



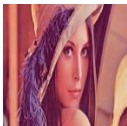









Cover image	Watermark image	PSNR	SSIM	Execution time
		58.5504	0.974	0.0320
		58.8200	0.985	0.0304
		58.6782	0.977	0.0312

TABLE 6. The PSNR, SSIM, and execution time with irreducible polynomial $x^4 + x^3 + 1$ and colored watermark image.

Cover image	Watermark image	PSNR	SSIM	Execution time
		58.6379	0.971	0.0389
		58.8635	0.982	0.0414
		58.8975	0.973	0.0351

colored images, contributing to a more robust and effective watermarking process.

Table 2 demonstrates that when employing the irreducible polynomial $x^4 + x + 1$, the watermarking algorithm yielded its most notable PSNR result, reaching 50.8, while maintaining an efficient execution time of 0.0313 seconds. This achievement was observed specifically for the standard “Lena” image. In Table 4, the utilization of the irreducible polynomial $x^4 + x^3 + 1$ to construct a multiplication table has led to notable improvements in both execution speed and

image quality. Specifically, for the “Lena” image, the PSNR has significantly increased to 58.8, while the execution time has been reduced to 0.0304 seconds.

To compute the improvement percentage, the following formula has been used:

$$Improvement\ Percentage = \frac{New_{value} - Old_{value}}{old_{value}} \times 100\% \tag{3}$$

[25]

As a result, the performance is improved when using the second irreducible polynomial $x^4 + x^3 + 1$, with the PSNR increasing by 15.75%, and the execution time improving by 2.88% according to the formula mentioned as (3).

D. WATERMARK EXTRACTION USING MULTIPLICATIVE INVERSE TABLE

The extraction process involves recovering a hidden watermark from a watermarked image by utilizing the information from the original cover image. Initially, the watermarked image and the cover image are loaded into the system. Through a bitwise XOR operation between these two images, the embedded watermark is revealed. This extracted watermark, represented in $GF(2^4)$ multiplicative inverse table values, undergoes a transformation where the values are shifted to the left by 4 bits. This step is crucial as it counteracts the previous conversion to the $GF(2^4)$ multiplication table during the embedding process. The resulting values are then saved, constituting the extracted watermark image. This intricate process ensures the retrieval of the original watermark, allowing for authentication or verification purposes in various applications such as digital watermarking and image integrity verification. Fig1. illustrates the NC of extracted watermark image.

E. ROBUSTNESS MEASUREMENTS

The robustness of a watermarking scheme, which is essential for evaluating its performance, is quantified by its ability to withstand deliberate attacks. A benchmark known as the Normalized Cross-Correlation (NC) has been introduced to facilitate the assessment of the robustness of the presented watermarking technique in a fair and effective manner. With a range spanning from 0 to 1, the NC value is employed to provide a clear indication of the watermarking system’s robustness. Importantly, a stronger resistance to attacks is indicated by a higher NC value, thereby demonstrating the robustness of the approach. Conversely, a reduced ability to withstand attacks is implied by a lower NC value, highlighting the vulnerability of the watermarking algorithm.

The NC value ranges from -1 (perfect negative correlation) to 1 (perfect positive correlation). A value of 0 indicates no correlation between the two signals or images.

In the context of watermarking, the NC is often used to measure the similarity or correlation between the original watermark and the extracted watermark from a watermarked

TABLE 7. The applied attacks with extracted watermark.

Watermarked image				Extracted Watermark After Attack
Compression QF=70				
Gaussian Noise NV=0.01				
Salt & Pepper Noise p1=0.01, p2=0.01				
Rotation angle=30				

*QF=Quality Factor, NV=Noise Variance, P1, p2=Probability1 and Probability2.

TABLE 8. The NC values after applying different attacks for robustness measurements.

	Compression	Gaussian Noise	Salt & Pepper Noise	Rotation
Pepper	0.995	0.952	0.933	0.811
Lena	0.998	0.948	0.939	0.795
Parrot	0.987	0.943	0.940	0.836

image. Higher NC values indicate a stronger correlation and better watermark retrieval performance.

TABLE 5 offers a concise overview of the watermarking system’s performance across different attacks. Specifically, it assesses the system’s robustness in the face of common challenges. The “Compression” column indicates the system’s ability to maintain watermark integrity after compression, with higher values signifying better performance. Similarly, the “Gaussian Noise” and “Salt & Pepper Noise” columns gauge its resilience against noise, where higher scores like 0.952 for Pepper indicate stronger noise resistance. Lastly, the “Rotation” column evaluates how well the watermark withstands image rotation, with values such as 0.811 for Pepper reflecting its performance.

A comprehensive comparison of the proposed scheme with other related works is presented in Table 6, showcasing the superiority of the proposed method in terms of watermarked image quality, imperceptibility, and execution time. Notably, a substantial 35% improvement in image quality is

TABLE 9. Comparison of the proposed method with other schemes.

Scheme	Tested image	PSNR	Execution time (second)
[1]	Lena	44.4928	1.2634
[20]	Lena	45.88	3.63
[22]	Lena	46.57	1.9154
	Pepper	46.48	
[26]	Lena	35.48	0.7123
	Pepper	35.22	
Proposed method	Lena	58.8200	0.0304
	Pepper	58.5504	

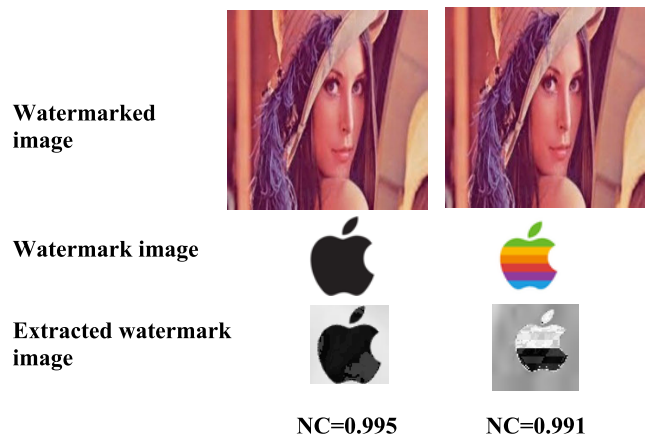


FIGURE 1. The NC of extracted Watermark image.

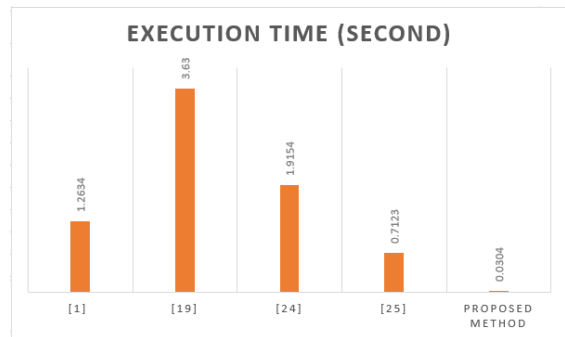


FIGURE 2. PSNR comparison of the proposed method with other schemes.

demonstrated when compared to the studies included in the comparison. Furthermore, the execution time for the watermark embedding process has been significantly enhanced, revealing 98% improvement. These findings underscore the remarkable advancements achieved by the proposed method, positioning it as an effective solution in the field of watermarking. Figures 1 and 2 visually depict the comparison with alternative watermarking techniques.

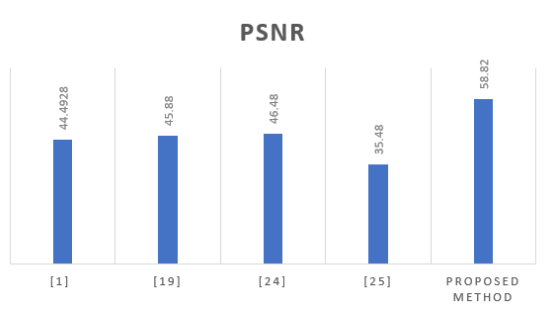


FIGURE 3. Execution time comparison of the proposed method with other schemes.

TABLE 10. PSNR values when utilizing the bows dataset.

Irreducible polynomial	Watermark1 (64x64)	PSNR	Watermark2 (225x225)	PSNR
x^4+x+1		42.9492		42.30
x^4+x^3+1		49.8230		48.05

TABLE 11. Addition table of $GF(2^3)$.

[0 1 2 3 4 5 6 7]
[1 0 3 2 5 4 7 6]
[2 3 0 1 6 7 4 5]
[3 2 1 0 7 6 5 4]
[4 5 6 7 0 1 2 3]
[5 4 7 6 1 0 3 2]
[6 7 4 5 2 3 0 1]
[7 6 5 4 3 2 1 0]

TABLE 12. Addition table of $GF(2^4)$.

[0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15]
[1 0 3 2 5 4 7 6 9 8 11 10 13 12 15 14]
[2 3 0 1 6 7 4 5 10 11 8 9 14 15 12 13]
[3 2 1 0 7 6 5 4 11 10 9 8 15 14 13 12]
[4 5 6 7 0 1 2 3 12 13 14 15 8 9 10 11]
[5 4 7 6 1 0 3 2 13 12 15 14 9 8 11 10]
[6 7 4 5 2 3 0 1 14 15 12 13 10 11 8 9]
[7 6 5 4 3 2 1 0 15 14 13 12 11 10 9 8]
[8 9 10 11 12 13 14 15 0 1 2 3 4 5 6 7]
[9 8 11 10 13 12 15 14 1 0 3 2 5 4 7 6]
[10 11 8 9 14 15 12 13 2 3 0 1 6 7 4 5]
[11 10 9 8 15 14 13 12 3 2 1 0 7 6 5 4]
[12 13 14 15 8 9 10 11 4 5 6 7 0 1 2 3]
[13 12 15 14 9 8 11 10 5 4 7 6 1 0 3 2]
[14 15 12 13 10 11 8 9 6 7 4 5 2 3 0 1]
[15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0]

When utilizing the BOWS dataset [26], comprising 10,000 standard images, with watermark1 (64 × 64) and watermark2 (225 × 225), the following results were obtained:

This comparison suggests that the choice of the irreducible polynomial has a substantial impact on the PSNR, which is an indicator of image quality in watermarking. The second

TABLE 13. Addition table of $GF(2^5)$.

[0 1 2 ... 29 30 31]
[1 0 3 ... 28 31 30]
[2 3 0 ... 31 28 29]
...
[29 28 31 ... 0 3 2]
[30 31 28 ... 3 0 1]
[31 30 29 ... 2 1 0]

TABLE 14. Addition table of $GF(2^6)$.

[0 1 2 ... 61 62 63]
[1 0 3 ... 60 63 62]
[2 3 0 ... 63 60 61]
...
[61 60 63 ... 0 3 2]
[62 63 60 ... 3 0 1]
[63 62 61 ... 2 1 0]

TABLE 15. Multiplication table in $GF(2^3)$.

[1, 2, 3, 4, 5, 6, 7],
[2, 4, 6, 3, 1, 7, 5],
[3, 6, 5, 7, 4, 1, 2],
[4, 3, 7, 6, 2, 5, 1],
[5, 1, 4, 2, 7, 3, 6],
[6, 7, 1, 5, 3, 2, 4],
[7, 5, 2, 1, 6, 4, 3]]

TABLE 16. Multiplication table in $GF(2^5)$.

[1, 2, 3, 4, 5, 6, ... 26, 27, 28, 29, 30, 31],
[2, 4, 6, 8, 10, 12, ... 17, 19, 29, 31, 25, 27],
[3, 6, 5, 12, 15, 10, ... 11, 8, 1, 2, 7, 4],
...
[29, 31, 2, 27, 6, 4, ... 15, 18, 11, 22, 20, 9],
[30, 25, 7, 23, 9, 14, ... 4, 26, 10, 20, 19, 13],
[31, 27, 4, 19, 12, 8, ... 30, 1, 22, 9, 13, 18]]

polynomial, $x^4 + x^3 + 1$, yielded a notably higher PSNR, indicating better image quality and imperceptibility compared to the first polynomial, $x^4 + x + 1$.

These results demonstrate the importance of carefully selecting the irreducible polynomial when implementing watermarking algorithms, as it can significantly affect the quality and robustness of the watermarked images. The second polynomial, $x^4 + x^3 + 1$, appears to be a more suitable choice for this particular watermarking task based on the higher PSNR achieved. The method exhibits a bal-

TABLE 17. Multiplication table in GF(2⁶).

[[1, 2, 3, ... 61, 62, 63],
[2, 4, 6, ... 57, 63, 61],
[3, 6, 5, ... 4, 1, 2],
...
[61, 57, 4, ... 46, 42, 23],
[62, 63, 1, ... 42, 43, 21],
[63, 61, 2, ... 23, 21, 42]]

TABLE 18. GF(2³).

W	-W	W ⁻¹
0	0	-
1	1	1
2	2	5
3	3	6
4	4	7
5	5	2
6	6	3
7	7	4

TABLE 19. GF(2⁴).

W	-W	W ⁻¹
0	0	-
1	1	1
2	2	9
3	3	14
4	4	13
....
12	12	10
13	13	4
14	14	3
15	15	8

anced performance across different watermark sizes, offering versatility and adaptability for various applications with varying requirements for payload size and perceptual quality. Researchers and practitioners can leverage this flexibility based on the specific demands of their use cases.

IV. CONCLUSION

In conclusion, the research was motivated by the goal of advancing watermarking techniques for the protection of digital content. Through the experiments conducted, it was demonstrated that the choice of irreducible polynomials significantly influenced watermarking quality. Specifically, the

TABLE 20. GF(2⁵).

W	-W	W ⁻¹
0	0	-
1	1	1
2	2	18
3	3	28
4	4	9
5	5	23
6	6	14
...
26	26	21
27	27	31
28	28	3
29	29	19
30	30	20
31	31	27

TABLE 21. GF(2⁶).

W	-W	W ⁻¹
0	0	-
1	1	1
2	2	33
3	3	62
4	4	49
5	5	43
6	6	31
...
58	58	59
59	59	58
60	60	55
61	61	16
62	62	3
63	63	32

use of the irreducible polynomial $x^4 + x^3 + 1$ resulted in a 15% improvement when compared to $x^4 + x + 1$.

Furthermore, the method excelled in comparison to other studies, showcasing a remarkable 35% enhancement in image quality and a 98% reduction in execution time. These findings underscore the substantial progress achieved in watermarking technology. However, it should be noted that the scalability of the approach may vary depending on multimedia file size and complexity.

Regarding future work, efforts will be made to investigate techniques aimed at further optimizing watermarking sys-

tems, with a particular emphasis on real-time applications such as live streaming and video conferencing.

APPENDIX

A. ADDITION TABLE OF GF (2³)

See Table 11.

B. ADDITION TABLE OF GF (2⁴)

See Table 12.

C. ADDITION TABLE OF GF (2⁵)

See Table 13.

D. ADDITION TABLE OF GF (2⁶)

See Table 14.

E. MULTIPLICATION TABLE IN GF (2³)

See Table 15.

F. MULTIPLICATION TABLE IN GF (2⁵)

See Table 16.

G. MULTIPLICATION TABLE IN GF (2⁶)

See Table 17.

H. ADDITION AND MULTIPLICATIVE INVERSE TABLE IN GF (2ⁿ)

See Tables 18–21.

REFERENCES

- Z. Yuan, Q. Su, D. Liu, and X. Zhang, "A blind image watermarking scheme combining spatial domain and frequency domain," *Vis. Comput.*, vol. 37, no. 7, pp. 1867–1881, Jul. 2021.
- W. Wan, J. Wang, Y. Zhang, J. Li, H. Yu, and J. Sun, "A comprehensive survey on robust image watermarking," *Neurocomputing*, vol. 488, pp. 226–247, Jun. 2022.
- D. K. Mahto and A. K. Singh, "A survey of color image watermarking: State-of-the-art and research directions," *Comput. Electr. Eng.*, vol. 93, Jul. 2021, Art. no. 107255.
- R. Sinhal, I. A. Ansari, and C. W. Ahn, "Blind image watermarking for localization and restoration of color images," *IEEE Access*, vol. 8, pp. 200157–200169, 2020.
- T. N. Hummadia and N. F. Hassanb, "Survey of recent video watermarking techniques," *Eng. Technol. J.*, vol. 39, no. 1B, pp. 165–174, Mar. 2021.
- G. U. Onoja, M. A. Bagiwa, and A. M. Kufena, "Robust watermarking techniques for the authentication and copyright protection of digital images: A survey," *Sule Lamido Univ. J. Sci. Technol.*, vol. 6, nos. 1–2, pp. 100–112, Mar. 2023. [Online]. Available: <https://doi.org/10.56471/slujst.v6i.366>
- Y. A. Hassan and A. M. S. Rahmah, "An overview of robust video watermarking techniques," *Iraqi J. Sci.*, vol. 64, no. 7, pp. 4513–4524, Jul. 2023.
- Y. Li, L. Zhang, H. Wang, and X. Wang, "Multiple security protection algorithm for GF-2 images based on commutative encryption and watermarking," in *Proc. Int. Conf. Spatial Data Intell.*, Hangzhou, China. Cham, Switzerland: Springer, Apr. 2021, pp. 141–147.
- G. Tong, Z. Liang, F. Xiao, and N. Xiong, "A residual chaotic system for image security and digital video watermarking," *IEEE Access*, vol. 9, pp. 121154–121166, 2021.
- S. R. Pillutla and L. Boppana, "An area-efficient bit-serial sequential polynomial basis finite field GF(2) multiplier," *AEU-Int. J. Electron. Commun.*, vol. 114, Feb. 2020, Art. no. 153017.
- K.-W. Kim, "Low-latency semi-systolic architecture for multiplication over finite fields," *IEICE Electron. Exp.*, vol. 16, no. 10, 2019, Art. no. 20190080.
- A. Sahin and I. Guler, "A survey of digital image watermarking techniques based on discrete cosine transform," *Int. J. Inf. Secur. Sci.*, vol. 10, no. 3, pp. 99–110, 2021.
- D. K. Matrassulova, Y. S. Vitulyova, S. V. Konshin, and I. E. Suleimenov, "Algebraic fields and rings as a digital signal processing tool," *Indonesian J. Electr. Eng. Comput. Sci.*, vol. 29, no. 1, p. 206, Jan. 2022.
- I. Khalid, T. Shah, K. A. Almarhabi, D. Shah, M. Asif, and M. U. Ashraf, "The SPN network for digital audio data based on elliptic curve over a finite field," *IEEE Access*, vol. 10, pp. 127939–127955, 2022.
- M. M. Hazzazi, S. Attuluri, Z. Bassfar, and K. Joshi, "A novel cipher-based data encryption with Galois field theory," *Sensors*, vol. 23, no. 6, p. 3287, Mar. 2023.
- A. M. Awaludin, H. T. Larasati, and H. Kim, "High-speed and unified ECC processor for generic Weierstrass curves over GF(p) on FPGA," *Sensors*, vol. 21, no. 4, p. 1451, Feb. 2021.
- S. Hussain, M. Asif, T. Shah, A. Mahboob, and S. M. Eldin, "Redesigning the serpent algorithm by PA-loop and its image encryption application," *IEEE Access*, vol. 11, pp. 29698–29710, 2023.
- Y. Wang, H. Xie, and R. Wang, "Digital signature scheme to match generalized Reed–Solomon code over GF(q)," *Proc. Cyberspace Saf. Secur., 14th Int. Symp. (CSS)*, Xi'an, China. Cham, Switzerland: Springer, Oct. 2022, pp. 32–47.
- X. Zhang and Y. Lao, "On the construction of composite finite fields for hardware obfuscation," *IEEE Trans. Comput.*, vol. 68, no. 9, pp. 1353–1364, Sep. 2019.
- K. M. Hosny, A. Magdi, N. A. Lashin, O. El-Komy, and A. Salah, "Robust color image watermarking using multi-core raspberry Pi cluster," *Multimedia Tools Appl.*, vol. 81, no. 12, pp. 17185–17204, May 2022.
- A. Broumandnia, "Image encryption algorithm based on the finite fields in chaotic maps," *J. Inf. Secur. Appl.*, vol. 54, Oct. 2020, Art. no. 102553.
- S. S. Jamal, T. Shah, S. Farwa, and M. U. Khan, "A new technique of frequency domain watermarking based on a local ring," *Wireless Netw.*, vol. 25, no. 4, pp. 1491–1503, May 2019.
- P. Amrit, K. N. Singh, N. Baranwal, A. K. Singh, J. P. Singh, and H. Zhou, "Deep learning-based segmentation for medical data hiding with Galois field," *Neural Comput. Appl.*, pp. 1–16, Nov. 2022. [Online]. Available: <https://doi.org/10.1007/s00521-023-09151-2>
- F. N. Al-Wesabi, "A hybrid intelligent approach for content authentication and tampering detection of Arabic text transmitted via internet," *Comput. Mater. Continua*, vol. 66, no. 1, pp. 195–211, 2020.
- S. Phanyaem, "Explicit formulas and numerical integral equation of ARL for SARX (Pr)_L model based on CUSUM chart," *Math. Statist.*, vol. 10, no. 1, pp. 88–99, 2022.
- F. Zhang, T. Luo, G. Jiang, M. Yu, H. Xu, and W. Zhou, "A novel robust color image watermarking method using RGB correlations," *Multimedia Tools Appl.*, vol. 78, no. 14, pp. 20133–20155, Jul. 2019.



YASMIN ALAA HASSAN received the B.Sc. and M.Sc. degrees in computer science from the University of Baghdad, where she is currently pursuing the Ph.D. degree with the Department of Computer Science, College of Science. She is a Lecturer Assistant with the Department of Computer Science, College of Science, University of Baghdad. Her research interests include security, watermarking, video, and image processing.



ABDUL MONEM S. RAHMA received the B.Sc. degree in computer science from the Military College of Engineering, University of Baghdad, the M.Sc. degree in computer science from Brunel University London, and the Ph.D. degree from the Loughborough University of Technology, U.K. He is currently a Professor with the Department of Computer Science, College of Science, Al-Maarif University College. His research interests include image processing, biometrics, computer security, and graphics.