

## RESEARCH ARTICLE

# Robust and Secure Zero-Watermark Architecture With Arnold and Phase Transformation

**HSIU-CHI TSENG**  **AND KING-CHU HUNG**

College of Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 824005, Taiwan

Corresponding author: Hsiu-Chi Tseng (0415913@nkust.edu.tw)

**ABSTRACT** This paper propose a secure watermarking method based on the zero-watermarking framework, combining Arnold transformation, phase transformation, and advanced encryption techniques. The proposed method aims to enhance the concealment and security of the watermark. By applying Arnold scrambling to the images, the hiding capability of the watermark is improved, preventing unauthorized extraction. The image data is decomposed into overlapping blocks in the vertical and horizontal directions, and these blocks are extracted as the image features and transformed into the phase domain to ensure robustness against various image operations. To enhance security, the watermark embedding is encrypted using the Rivest-Shamir-Adleman (RSA) asymmetric encryption algorithm with a Golden Key. Experimental results demonstrate that the proposed method maintains excellent watermark quality and visibility under different noise attacks, while providing a high level of tamper resistance. The effectiveness of the method is evaluated using metrics such as Bit Error Rate (BER) and Normalized Correlation (NC). Overall, the proposed secure watermarking method exhibits strong performance in terms of robustness and security.


**INDEX TERMS** Zero-watermarking framework, Arnold transformation, phase transformation, Rivest-Shamir-Adleman (RSA), bit error rate (BER), normalized correlation (NC).

## I. INTRODUCTION

Watermarking plays a critical role in safeguarding the integrity and ownership of digital content [1]. With the increasing ease of unauthorized copying, distribution, and manipulation of digital media, the development of robust and secure methods for watermark embedding and detection has become paramount [2]. The utilization of watermarking techniques enables the identification and authentication of copyrighted material, ensuring its rightful ownership and preventing unauthorized usage [3]. These techniques find applications in various domains, including copyright protection, content authentication, and data integrity verification [4], [5]. By embedding unique and imperceptible signatures into digital media, watermarking allows rightful owners to assert their ownership and verify the authenticity of the content. Ensuring the resilience and security of the embedded watermark against various attacks and transformations, including noise

attacks and safeguarding against legitimate licensing, remains a major concern in watermarking techniques.

Within the sphere of digital watermarking, the process of embedding watermarks can be categorized into three types: robust, fragile, and zero-watermark. Robust watermarks are typically intertwined with transform domain embedding techniques, wherein the essence of digital media is subtly transformed using coefficients and sophisticated algorithms to embed the watermark [6]. This strategic approach imbues the digital image with fortified resilience against an array of potential attacks, making it a staple for safeguarding copyright interests [7], [8]. In contrast, the delicate nature of fragile watermarks finds expression through direct embedding methodologies [9]. Here, watermarks are seamlessly interwoven into the very fabric of pixel values or datasets within the digital image. Such interventions might encompass nuanced pixel value manipulations or orchestrated bit position alterations. The trade-off, however, is a somewhat reduced armor against attacks, rendering this methodology particularly suited for unearthing telltale signs of unauthorized tampering [10], [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Kashif Saleem .

The innovative realm of zero-watermarking introduces a distinctive paradigm. It involves extracting proprietary feature values from digital media through meticulously designed algorithms [12], [13]. These extracted features are artfully correlated with the watermark, employing an array of techniques [14]. In the domain of zero-watermarking, the application involves the extraction of intricate image features, coupled with the astute design of algorithms for the seamless embedding of watermarks. This multifaceted process entails the discerning extraction of image attributes, culminating in the ingenious implementation of algorithms that facilitate the optimal integration of the watermark. This methodology harmonizes the pursuit of copyright protection with the preservation of the unblemished integrity of the original digital image [15], [16], [17].

Drawing upon the merits inherent in the zero-watermarking approach, the present paper ingeniously employs a zero-watermark framework as the cornerstone of our watermarking algorithm.

The concept of the zero-watermarking framework was initially proposed by Wen et al. [18]. This framework eliminates the need for the original unmarked image during the watermark embedding process and focuses on directly embedding the watermark into the extracted features or attributes from the host image. Due to its robustness and security, the zero-watermarking framework has gained significant attention and has been widely applied in various watermarking methods. For example, researchers Wang et al. [19], Chen et al. [20], Wang et al. et al. [21], and Huang et al. [22] presented a watermarking method based on the zero-watermarking framework. However, despite the many advantages of the zero-watermarking framework, it also has certain limitations and drawbacks. The watermark embedding process, relying on the features of the host image, may introduce inevitable distortions or a decrease in visibility for some images. Additionally, zero-watermarking methods may be more vulnerable to certain forms of noise attacks, and there are also some controversies regarding the security of ownership authentication.

A digital watermarking method is typically evaluated based on several essential attributes, including imperceptibility, robustness, security, and payload [23], [24]. However, zero-watermarking stands apart by not requiring the embedding of data directly into the original image, ensuring that the visual quality of the image remains unaltered, in line with imperceptibility requirements. Recent advancements in zero-watermarking technology have enhanced its resistance against various attacks, showcasing robustness. When combined with well-crafted combinations of features and algorithms, it can also contribute to enhancing security. Notably, in terms of payload, zero-watermarking outperforms other watermarking frameworks. Since zero-watermarking doesn't introduce destructive changes to the original image, it can embed a substantial amount of watermarking data through diverse algorithmic designs. Additionally, it's noteworthy that zero-watermarking enables blind extraction of the watermark without requiring access to the original data. The choice of

the zero-watermarking framework in this paper signifies the inheritance of the advantageous characteristics inherent in this architecture. Furthermore, a blind watermark extraction approach has been devised.

This paper presents a novel method for securely watermarking Arnold transformed images within a zero-watermarking framework using phase transformation. The proposed method aims to address the limitations of existing approaches and enhance the robustness and security of the watermarking process. We improve the stability of the feature values by applying the Arnold scrambling on the images and utilizing phase transformation. Additionally, we introduce the application of RSA asymmetric key encryption. This is primarily done to achieve two objectives: first, to enhance ownership authentication, and second, to improve the security of embedded watermarks.

## II. PRELIMINARIES

This section introduces four key techniques: Arnold, Phase, Shuffle, and RSA, which will be applied within the proposed framework for watermark embedding. The utilization of Arnold ensures the confidentiality of the watermark, while Phase is employed to enhance the robustness of key feature values. Shuffle is utilized to introduce variations in the embedded watermark, thereby enhancing security. Finally, the RSA asymmetric encryption is employed to achieve irreversible encrypted data and provide ownership authentication. In summary, the application of Arnold, Phase, Shuffle, and RSA techniques enhances the confidentiality of the watermark, stability of feature values, security through variations, and ownership authentication provided by irreversible encryption. The integration of these techniques elevates the security level of the watermarking process.

### A. ARNOLD TRANSFORMATION

Chaotic mapping is a mathematical transformation technique that finds wide applications in fields such as image processing, cryptography, and data encryption. These mappings exhibit sensitivity to initial conditions and are characterized by their ergodicity, irregular trajectories, and high entropy. Chaotic mappings play a vital role in generating pseudo-random numbers, achieving secure communication, and performing digital image transformations.

The Arnold transformation, proposed by Vladimir I [25], [26]. Arnold, is a specific chaotic mapping method used to shuffle the positions of image pixels. It is particularly suitable for applications such as visual encryption, digital watermarking, and steganography. This algorithm is based on the concept of dynamical systems and employs an iterative process of nonlinear mapping to induce chaotic behavior in the pixel distribution of an image.

The Arnold transformation is applicable to grayscale or color images of size  $N \times N$ , where each pixel is represented by its spatial coordinates  $(x, y)$ . The transformation proceeds through a series of  $T$  iterations, with each iteration leading to a rearrangement of the pixel positions. Specifically, the steps

of the Arnold transformation involve dividing the image into four equal quadrants. For each pixel  $(x, y)$ , a new coordinate  $(x', y')$  is assigned based on its quadrant. The mathematical representation of the Arnold transformation is as follows:

$$\begin{cases} x' = (2x + y) \bmod N \\ y' = (x + y) \bmod N \end{cases} \quad (1)$$

The Arnold transformation possesses several notable characteristics that make it suitable for cryptographic applications. Firstly, it is a reversible transformation, allowing the original image to be recovered by applying the inverse operation of the transformation. Secondly, the transformation exhibits strong mixing and diffusion properties, ensuring that even small changes in the input image result in significant output variations. Finally, the increased number of iterations, denoted by  $T$ , provides a larger key space, enhancing resistance against brute-force attacks.

In summary, the Arnold transformation is a nonlinear mapping algorithm based on chaotic theory and dynamical systems. It serves as a versatile tool for image encryption, digital watermarking, and data hiding, finding extensive applications in secure communication and multimedia protection. By iteratively rearranging the pixel mappings, the Arnold transformation generates visually captivating and complex transformation effects while ensuring the integrity and confidentiality of digital content.

### B. 1-D NON-REDUNDANT DISCRETE WAVELET TRANSFORM

The 1-D Non-Recursive Discrete Periodized Wavelet Transform (1-D NRDPWT) is a method used for signal processing and data compression [27]. Unlike the traditional recursive wavelet transform, the NRDPWT computes the full-order frequency sub-band coefficients in parallel without the need for iterative signal decomposition [28].

In the 1-D NRDPWT, a set of channel filters is used to decompose the signal. The coefficients of these filters are organized into a matrix  $A$ , where each column vector represents a filter. The construction of matrix  $A$  ensures specific relationships between the filters at different levels.

Specifically, the construction of matrix  $A$  is based on the scale and frequency relationships at each level. For the  $L$ -th level,  $A$  is composed of a matrix  $A_L$  with dimensions  $N$  by  $2^{-L}$ . In  $A_L$  the filter coefficients of adjacent columns have a vertical offset relationship of  $2^{(L-J)}$ . This construction ensures the corresponding relationships between filters at different levels in terms of scale and frequency. Mathematically, matrix  $A$  can be represented as:

$$A = [B_0, A_0, A_1, A_2, \dots, A_{(J+1)}] \quad (2)$$

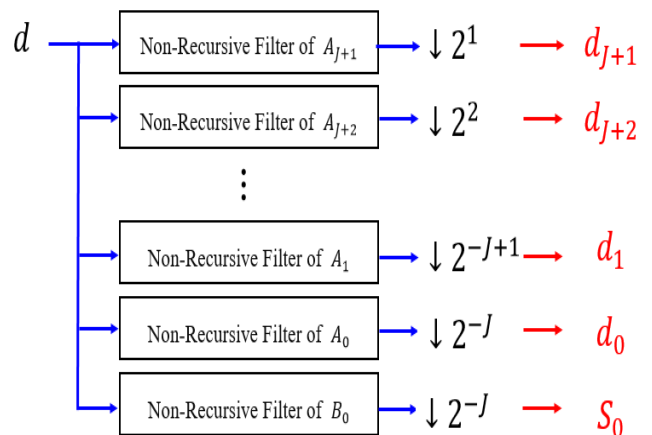
where  $B_0$  is a column vector composed of constant elements and can be seen as a bias vector.  $A_0, A_1, A_2, \dots, A_{(J+1)}$ , are matrices with dimensions  $N$  by  $2^{-L}$ , corresponding to the wavelet coefficients at different levels. In  $A_L$ , the filter coefficients of adjacent columns have a vertical offset relationship of  $2^{(L-J)}$ .

During the signal decomposition process, the input signal  $S$  is multiplied by the matrix  $A$  to obtain a vector  $d$  that contains the scale coefficients and wavelet coefficients. The scale coefficients represent the low-frequency part of the signal, while the wavelet coefficients represent the high-frequency details. The 1-D NRDPWT can be defined as (3) and as shown in Fig. 1:

$$\begin{aligned} d &= S_j A \\ &= [s_0, d_0, d_1, d_2, \dots, d_{(J+1)}] \end{aligned} \quad (3)$$

In the above definition, the construction of matrix  $A$  ensures the specific vertical offset relationship between filters at different levels. Through this construction, the 1-D NRDPWT can perform multi-level scale transformation and frequency decomposition of the signal, extracting its feature information.

In the inverse transformation process, the original signal can be recovered by multiplying the vector  $d$  of scale coefficients and wavelet coefficients by the inverse matrix of  $A$ . The inverse matrix  $A^{-1}$  is the transpose matrix  $A'$  since  $A$  is a unitary matrix, meaning its determinant is equal to 1. Through the inverse transformation, the scale coefficients and wavelet coefficients can be recombined to reconstruct the original signal.



**FIGURE 1.** The filter architecture diagram of NRDPWT. This architecture utilizes parallel computation to simultaneously compute the wavelet coefficients across all scales in one step.

The 1-D NRDPWT is a non-recursive method that achieves signal decomposition and reconstruction using a specifically constructed filter matrix  $A$  and scale transformation operation  $S_j$  [27], [28]. It finds broad applications in signal processing and data compression, effectively extracting feature information from signals while reducing computational costs. The offset relationships in the construction of matrix  $A$  further enhance the properties of the 1-D NRDPWT.

### C. FISHER-YATES ALGORITHM

In the domain of watermarking research, the Fisher-Yates Algorithm is a widely used method for generating random permutations. This algorithm was initially proposed by

Ronald Fisher and Frank Yates [29]. It offers an efficient and straightforward approach to rearrange a sequence of  $n$  elements into a random order.

The fundamental concept behind the Fisher-Yates Algorithm involves traversing the sequence and starting from the last element. At each iteration, a random position is selected, and the element at that position is swapped with the element at the current traversal position. Through the progression of the traversal, each element has an equal chance of being selected, thereby achieving a random permutation [30].

The Fisher-Yates Algorithm can be represented by the following mathematical formulation:

$$X_i = X_{j=Random(n)}, \quad \begin{cases} i \in \{1, 2, 3, 4, \dots, n\} \\ j \in \{1, 2, 3, 4, \dots, n\}, \end{cases} \quad (4)$$

where  $n$  denotes the length of the sequence,  $i$  represents the current traversal position, and  $j$  is a randomly generated integer between 1 and  $n$ . By traversing the sequence and performing the swapping operation.

The Fisher-Yates Algorithm finds wide application in various domains, including data analysis, simulation [31], and encryption [32], [33], among others. Its inherent randomness makes it a crucial tool in watermarking techniques, ensuring the unpredictability and security of watermarks.

#### D. RSA ENCRYPT/DECRYPT ALGORITHM

In the modern field of information security, Rivest-Shamir-Adleman (RSA) is a widely adopted asymmetric key encryption algorithm. It was jointly proposed by Ron Rivest, Adi Shamir, and Leonard Adleman [34], with the name derived from the initials of its inventors. The core idea behind RSA is based on the mathematical problem of integer factorization in number theory, which involves decomposing a large integer into the product of its prime factors.

In RSA, a pair of keys, namely the public key and the private key, are utilized. The public key is used for encrypting data, while the private key is used for decrypting data. This characteristic of asymmetric encryption allows two communicating parties to securely exchange encrypted information without the need for pre-shared secret keys.

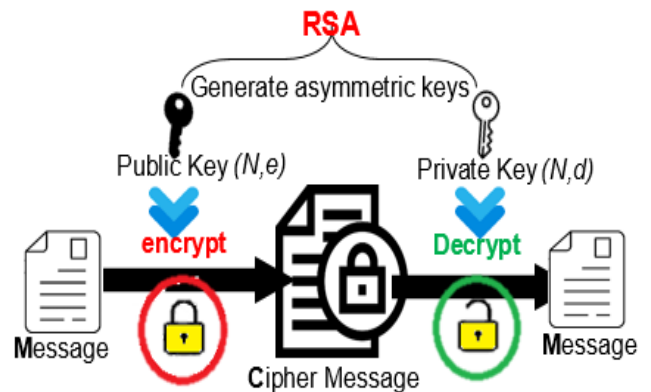
PKCS 8 format serves as a key storage format that supports various encryption algorithms [35]. When using RSA with PKCS 8 for encrypting and decrypting documents, it typically involves generating a pair of asymmetric keys, consisting of a public key and a private key. The public key  $(N, e)$  is used for encryption (5), while the private key  $(N, d)$  is used for decryption (6), as illustrated in Fig. 2. This enables individuals with the private key to access the encrypted documents, indicating that the private key holder has the authority and ownership over the documents. This further extends to the concept of ownership rights.

$$C = m^e \bmod N \quad (5)$$

$$M = C^d \bmod N \quad (6)$$

To encrypt a plaintext message  $M$  into an integer  $m$ , satisfying  $0 \leq m < N$ , using the public key  $(N, e)$ , and decrypt the resulting ciphertext  $C$  back into the plaintext message  $M$  using the private key  $(N, d)$

Utilizing RSA for encryption and decryption requires complex mathematical operations and key management. Therefore, in practical applications, it is crucial to carefully design and implement appropriate security measures to ensure the security and reliability of the encryption and decryption processes, thus protecting the rights of access ownership.



**FIGURE 2.** RSA Encryption and Decryption Process. Using a Public Key generated from an asymmetric key pair, the Message is employed as the encryption key, while the Private Key serves as the decryption key. This approach ensures both security and authentication.

### III. PROPOSED ZERO-WATERMARKING SCHEME

#### A. EMBEDDING WATERMARK

The size of the host image is denoted as  $N \times N$ , and the embedded watermark is a binary watermark of size  $M \times M$ . The embedding process of the zero watermark is illustrated in Fig. 3, and the embedding steps are described as follows:

##### Step 1: Scrambling the Host Image

The core concept of the Arnold algorithm is to perform mathematical transformations on the pixel positions within an image (1). By altering the positions of individual pixels in the image, it enhances security and increases the difficulty of watermark attacks. Therefore, in the initial stage of watermark embedding in this paper, we utilize an advanced version of the Arnold algorithm by introducing coefficients  $a$  and  $b$ , as well as varying the number of iterations denoted as  $T$ . This approach aims to enhance the security against decryption and provide uniqueness to each image signal in the host images we seek to protect. The scrambling formula is as follows:

$$\begin{aligned} x' &= (ax + b) \bmod N, \\ y' &= (ay + x) \bmod N \end{aligned} \quad (7)$$

Here,  $x'$  and  $y'$  represent the new pixel positions, while  $x$  and  $y$  represent the original pixel positions. Coefficients  $a$  and  $b$  are two parameters of the Arnold algorithm that determine the transformation of pixel positions. The selection of these coefficients can influence the rearranged image features, such

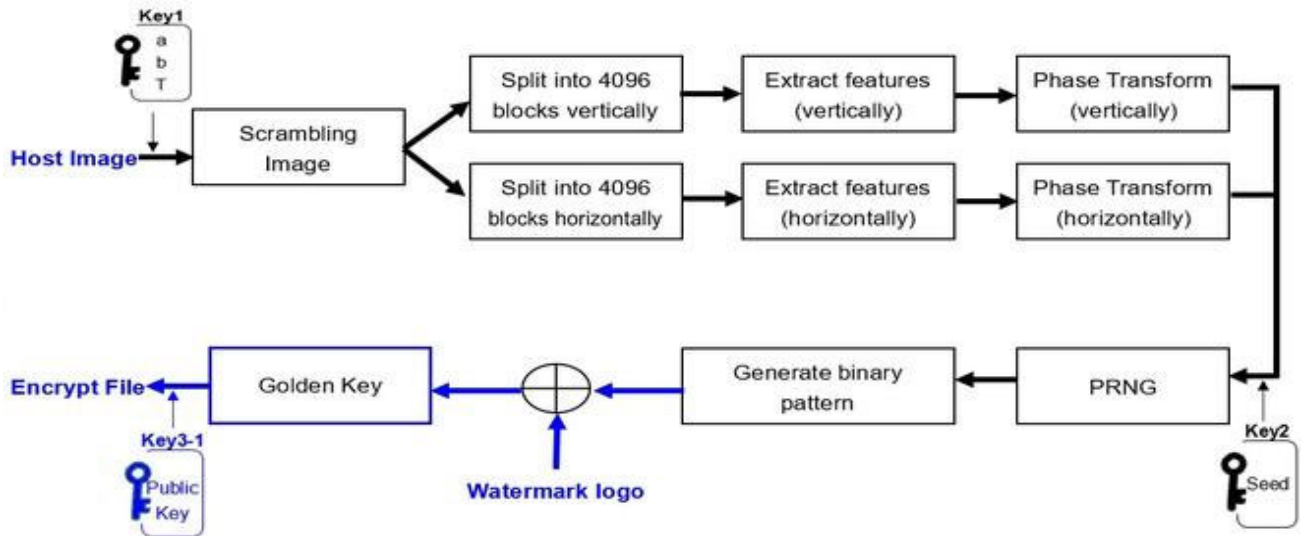


FIGURE 3. Embedded watermarking architecture. The input data for the embedding architecture comprises the original image and Key1. Key1 incorporates the three coefficients essential for the execution of the “Scrambling the host image” step, specifically a, b, and T.

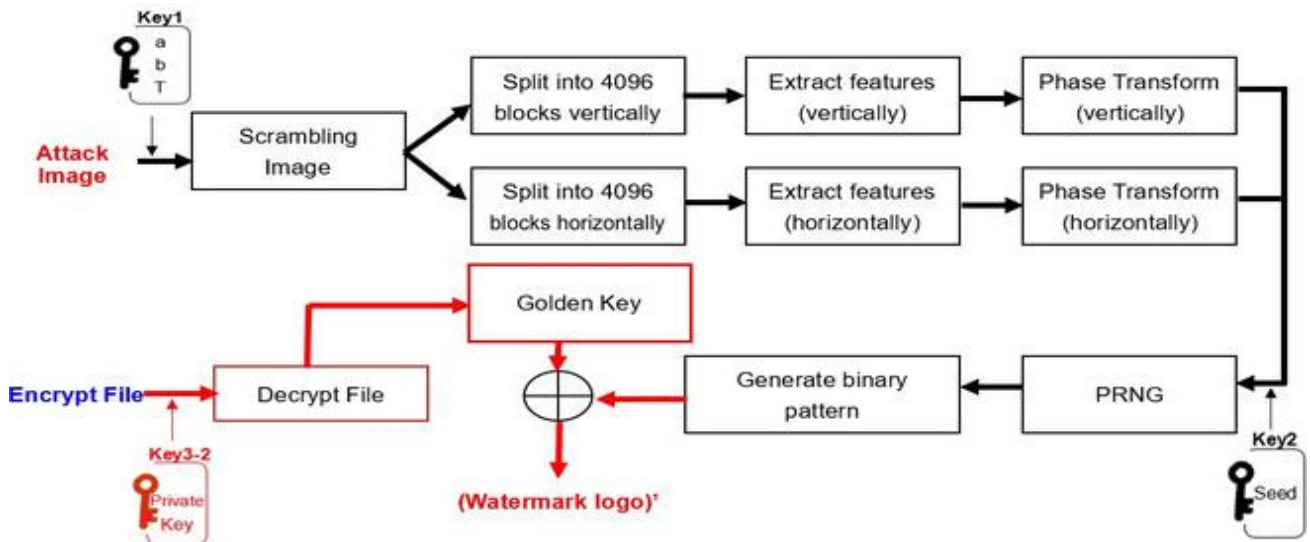


FIGURE 4. Extracted watermarking architecture. The input data for the extraction architecture consists of the image subjected to noise attacks and Key1. The watermarking framework employed in this paper is designed for blind extraction, requiring no reliance on the original image. It directly inputs the image affected by noise attacks for watermark extraction. Key1 encompasses the three coefficients necessary for executing the “Scrambling the host image” step, which are a, b, and T.

as pixel position distribution and relative relationships. Different coefficient values may lead to varying effects and performance outcomes.  $N$  refers to the width and height of the image. The iterative step (7) is repeated  $T$  rounds to obtain the final pixel positions.

Step 2: Extract Features

From the scrambled image, we partition the image into  $(\frac{N}{r}) \times (\frac{N}{r})$  blocks through horizontal and vertical segmentation, as illustrated in Fig. 5. To offer a practical example, let’s consider a digital image with dimensions of  $512 \times 512$ , pixels, where the parameter  $n$  corresponds to the image’s

equal length and height of 512. Given a pixel displacement value of 8 pixels, i.e.,  $r = 8$ , the image can be divided into  $(\frac{512}{8}) \times (\frac{512}{8}) = 4096$  distinct blocks.

Here,  $r$  denotes the pixel displacement value assigned to each block, and the physical size of each block extends by  $r$  pixels on both sides, resulting in an extension of  $r + r = 2r$  pixels from the central pixel. Consequently, the dimensions of each block, both in length and height, are defined as  $r + 2r = 3r$ . Irrespective of whether the segmentation is performed horizontally or vertically, we undertake the computation of the average pixel value within each individual block. These

computed values ( $F_V$  and  $F_H$ ) serve as the designated feature descriptors for the image, as detailed in (8).

However, due to the vulnerability of image edges to potential geometric attacks, we implement a measure to enhance the robustness of these feature descriptors. This entails replacing the values of blocks located at the four edges with the average values of their neighboring blocks. This approach effectively fortifies the strength and stability of the feature descriptors, mitigating susceptibility to geometric interference.

$$\begin{aligned}
 F_V[w, h] &= \frac{1}{(3r)^2} \sum_{x'=r \times (w-1)}^{r \times (w+1)} \sum_{y'=r \times (h-1)}^{r \times (h+1)} \text{Img}(x', y') \\
 F_H[w, h] &= \frac{1}{(3r)^2} \sum_{x'=r \times (w-1)}^{r \times (w+1)} \sum_{y'=r \times (h-1)}^{r \times (h+1)} \text{Img}(y', x'),
 \end{aligned}$$

where

$$\begin{aligned}
 h &= 3, 4, 5, 6, \dots, \left(\frac{N}{r}\right) - 2 \\
 w &= 3, 4, 5, 6, \dots, \left(\frac{N}{r}\right) - 2
 \end{aligned} \tag{8}$$

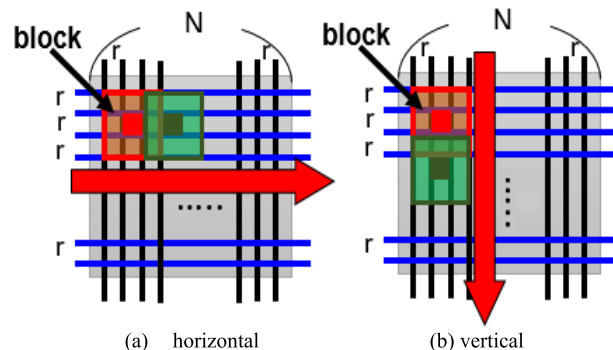
where the vertical displacement value of a block is defined as  $h$ , while the horizontal displacement value is denoted as  $w$ . Originally, both  $h$  and  $w$  were intended to initiate from 1. However, given the potential impact of attacks on the boundaries, the initial displacement for feature value extraction now commences from 3 and extends up to  $\left(\frac{N}{r}\right) - 2$ . The initiation values for the blocks, nonetheless, still range from 1 to  $\left(\frac{N}{r}\right)$ . For both  $h$  and  $w$  were values 1 and 2, as well as  $\left(\frac{N}{r}\right) - 1$  and  $\left(\frac{N}{r}\right) - 2$ , the feature values are substituted with the averages of their neighboring blocks, considering the implications of potential interference. By way of illustration, consider the following:  $F_V[1, 1] = (F_V[1 + 2, 1 + 2] + F_V[1 + 3, 1 + 3]) / 2$ . However, within the context of  $\text{Img}, x'$  and  $y'$  represent the pixel values at the coordinates of the image.

**Step 3: Phase Transformation**

In order to facilitate signal conversion, we employed an expanded computational approach based on the 1-D NRDPWT (2). Our choice involved utilizing a matrix, denoted as  $A_0$ , in which each element is derived from the  $g$ -th column and the  $k$ -th row as  $a_{(g,k)}$ . Notably, the first row of  $A_0$  is defined by the element value  $a_{(g,1)} = [a_{(1,1)}, a_{(2,1)}, a_{(3,1)}, \dots]$ . Therefore, we define  $p[k]$  as  $\sum_{g=1}^{2^{-J}} a_{(g,k)}$ , which means that  $p[1]$  equals  $\sum_{g=1}^{2^{-J}} a_{(g,1)}$ . Hence, the transformation formula phase ( $P$ ) is defined as follows:

$$P = [p[1], p[1], p[1], \dots, \dots, p[2^{-J}]]^T \tag{9}$$

Next, we utilize equation (9) to perform the phase transformation (10) on the extracted feature values obtained in the previous stage. By repeatedly applying (10) with a shift of 1 to the positions of the feature values and continuing this



**FIGURE 5. Horizontal and Vertical Block Segmentation Diagram.**  $N$  represents the pixel size of the image's width and height.  $r$  denotes the magnitude of each displacement. The red and green boxes depict blocks segmented in sequential order. The area of intersection between them represents their overlap.

process until they return to their initial positions, we halt the transformation.

$$\begin{aligned}
 P_V &= \cos^{-1}(F_V \odot P), \\
 P_H &= \cos^{-1}(F_H \odot P)
 \end{aligned} \tag{10}$$

This methodology capitalizes on the underlying principles of 1-D NRDPWT theory to enhance the stability of feature values during the signal conversion process, ultimately striving for optimal outcomes.

**Step 4: Generate Binary Pattern**

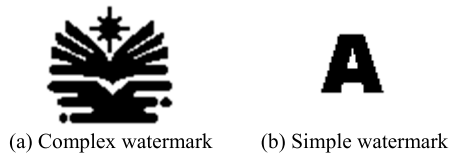
In this step, we input our unique  $key2$  value and utilize a pseudorandom number generator (PRNG) to generate two sequences of random numbers,  $R_V$  and  $R_H$ , each containing  $\left(\frac{N}{r}\right) \times \left(\frac{N}{r}\right)$  numbers ranging from 1 to  $\left(\frac{N}{r}\right) \times \left(\frac{N}{r}\right)$  in (11). From  $R_V$ , we extract the phase values for  $F_V$ , and from  $R_H$ , we extract the phase values for  $F_H$ . By comparing these two sets of values (12), we derive a binary pattern template  $PT$  through a binarization process.

$$\begin{cases}
 R_V \\
 = \left[ \text{key1}(\text{seed1}), \text{random}\left(1, \left(\frac{N}{r}\right) \times \left(\frac{N}{r}\right)\right) \right] \\
 R_H \\
 = \left[ \text{key1}(\text{seed2}), \text{random}\left(1, \left(\frac{N}{r}\right) \times \left(\frac{N}{r}\right)\right) \right],
 \end{cases} \tag{11}$$

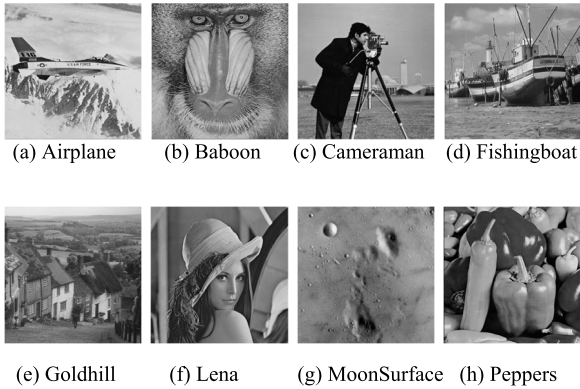
$$\begin{aligned}
 PT[n] &= \begin{cases} 1, & \text{if } (F_V[R_V[n]] > F_H[R_H[n]]) \\ 0, & \text{otherwise,} \end{cases} \\
 n &= 1, 2, 3, 4, \dots, \left(\frac{N}{r}\right) \times \left(\frac{N}{r}\right)
 \end{aligned} \tag{12}$$

**Step 5: Embedding Watermark**

Through the aforementioned transformations, convert them into PT. Then, utilizing the binary watermark logo, we perform XOR operation to achieve watermark embedding. The approach described in this paper involves transforming the digital image through a series of methods before embedding the watermark. This is accomplished through domain transformation.



**FIGURE 6. Binary Watermark Logo. (a) Represents a more complex binary watermark logo example. (b) Represents a simpler binary watermark logo example. Both (a) and (b) are sized at  $64 \times 64$ .**



**FIGURE 7. Host images. All eight experimental digital images (a) to (h) are grayscale and have a size of  $512 \times 512$  pixels.**

Let  $W$  be the original binary watermark logo. The original binary watermark is XOR with a binary pattern template  $PT$  to obtain the watermark key value Golden Key ( $GK$ ), as represented by the following mathematical expression:

$$GK = PT \oplus W \quad (13)$$

#### Step 6: Encrypt Golden Key

The generation of key3 using the RSA asymmetric encryption algorithm ensures the creation of distinct public key (refer to key 3-1 in Fig. 3) and private key (refer to key 3-2 in Fig. 3) for each individual image. The public key is employed for encryption purposes, while the private key is exclusively held by the owner of the watermarking system, serving as the decryption key for the files. Beyond its primary function, the most significant application of this approach lies in the authentication of image ownership and the assertion of image copyright. Therefore, utilizing the (5), the value ( $GK$ ) is encrypted using the encryption function  $E$ , resulting in the encrypted value (Cipher Text,  $CT$ ).

$$CT = E(\text{key3} - 1, GK) \quad (14)$$

## B. EXTRACTING WATERMARK

The architecture of zero-watermarking does not involve embedding the watermark directly into the original image, thus mitigating potential damage to the original image. As a result, the process of watermark extraction does not require reversible retrieval of the watermark from the image, distinguishing it from traditional watermarking techniques. Zero-watermarking is a nonreversible technique. Consequently,

within the framework we propose, the steps for extracting the watermark align with steps one to four of the watermark embedding framework described in this paper, with the sole distinction lying in the verification image. Hence, redundant elaboration is omitted. Thus, we proceed directly to step five, elucidating the process of watermark extraction.

#### Step 5: Decrypt Cipher Text ( $CT$ )

In this step, the decryption process is performed using the private key from asymmetric key cryptography. When the correct private key (refer to key 3-2 in Fig. 4) is applied, successful decryption takes place, allowing for the extraction of the original encrypted data, denoted as Golden Key ( $GK$ ). The decryption operation relies on decryption technique (6) and utilizes function  $D$  to execute the decryption algorithm, which can be expressed by the following formula:

$$GK = D(\text{key3} - 2, CT) \quad (15)$$

#### Step 6: Extracting Watermark

First, let us define the binary pattern template, which is generated during the extraction process of the fourth stage of watermarking, as  $PT'$ . It is crucial to emphasize that this pattern is not originated from the original image. Subsequently, by executing an XOR operation between the watermark key value  $GK$  and  $PT'$ , we can successfully extract the watermark  $W'$  from the verification image. The formula can be expressed as follows:

$$W' = PT' \oplus GK \quad (16)$$

where  $W'$  denotes the watermark extracted from verification images that have been subjected to various forms of attacks. By successfully completing the aforementioned steps, the extraction of the watermark can be achieved in a secure and efficient manner.

## IV. EXPERIMENTAL ANALYSIS AND RESULTS

To evaluate the efficacy of the proposed zero-watermarking scheme in combating noise, we conducted experiments using two binary watermarks of size  $64 \times 64$ , as depicted in Fig. 6. These watermarks encompass both a complex and a simple design. As for the original images used in the experiments, we selected a diverse set of eight  $512 \times 512$  images for analysis and empirical assessment, as shown in Fig. 7. To comprehensively evaluate the scheme, we subjected the test images to a total of 18 different attacks classified into six major categories, as presented in Table 1. The attacks were carefully designed to represent varying levels of severity.

The average Peak Signal-to-Noise Ratio (PSNR) values after the attacks were applied are reported in Table 1, which serves as a quantitative measure of the degree of interference experienced by the test images. PSNR is a widely used evaluation metric for quantifying the level of distortion between an original image and a modified or processed image.

The PSNR calculation formula is as follows:

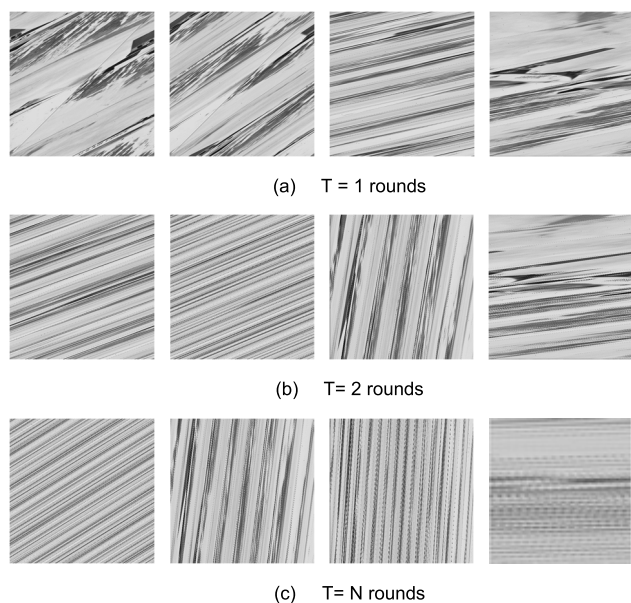
$$PSNR = 10 \log_{10} \left( \frac{1}{M \times N} \times \frac{(max)^2}{\sum_{x=1}^M \sum_{y=1}^N (Img_{x,y} - Img'_{x,y})^2} \right) \quad (17)$$

where  $max$  is the highest scale value of the 8-bits grayscale [22], [35]. And,  $M$  and  $N$  represent the width and height of the image ( $Img$ ), respectively.  $Img_{x,y}$  represents the pixel value of the original image, while  $Img'_{x,y}$  represents the pixel value of the modified or processed image. PSNR values are typically expressed in decibels (dB). Higher PSNR values indicate lower distortion and greater similarity to the original image. In general, higher PSNR values correspond to better image quality.

## A. EXPERIMENTAL ANALYSIS

### 1) ARNOLD TRANSFORMATION ANALYSIS

In this stage of analysis, we employ the Arnold algorithm to scramble the image, aiming to enhance the security and intricacy of watermark embedding. Specifically, we select Fig. 6(a) as the experimental input and vary the  $Key1$  values, representing the parameters  $a$ ,  $b$ , and  $T$ . By doing so, we obtain distinct patterns of scrambled images under different parameter settings, as demonstrated in Fig. 8.



**FIGURE 8.** The “Airplane” image undergoes different geometric transformations under varying Arnold scrambling parameters,  $a$ ,  $b$ , and  $T$ , resulting in different visual distortions. Refer to Figure 6 (a) for the original digital image of the airplane.

The primary objective of this experimental analysis is to explore the impact of varying parameters on the Arnold algorithm and further strengthen the robustness of watermarking. Through a thorough examination of the results in

Fig. 8, it becomes evident that the image scrambling effects exhibit variations with different parameter configurations. This observation underscores the critical role of adjusting input parameters within the Arnold algorithm in significantly augmenting the security of watermarking, rendering it more resilient against decryption or tampering attempts. Consequently, our findings contribute to the development of a robust algorithm for safeguarding the copyright and integrity of digital content.

To sum up, in this analysis stage underscores the application of the Arnold algorithm in advancing the field of watermarking and showcases the diverse scrambling effects achieved by manipulating different parameters.

### 2) FEATURE STABILITY ANALYSIS

The stability of image feature extraction under various noise attacks is a critical factor closely related to the quality of watermark extraction in the zero-watermarking system architecture. The objective of analyzing the stability of image features is to evaluate the robustness and consistency of feature extraction algorithms under different input images, using different types and intensities of attack conditions.

To conduct the stability analysis, we have chosen the airplane image shown in Fig. 7 (a) as the primary representative for experimental analysis and explanation in this stage. We have analyzed this image under various attack types listed in Table 1. By comparing the extracted feature values with the feature values obtained from the original image, we have proposed a feature extraction method that captures feature values from both the horizontal and vertical directions. Fig. 9 demonstrates the results of the vertical analysis ( $F_V$ ), and Fig. 10 demonstrates the results of horizontal analysis ( $F_H$ ). This analysis includes seven categories of noise attacks, each with three different attack intensities. Therefore, a total of  $7 \times 3 = 21$  different noise attacks have been analyzed and compared with the original image. The simplicity of colors indicates a closer resemblance to the original image, reflecting higher stability of the feature values. Conversely, larger deviations and more chaotic color patterns in the graph represent lower stability under the respective attack. The original image is represented by black points, blue color points represent Type 1 for the corresponding attack category, gray represents Type 2, and green represents Type 3. From the graph, it is evident that, except for minor influences observed in Gaussian noise and JPEG compression attacks, the remaining feature values exhibit significant fluctuations relative to the original image.

### 3) PHASE TRANSFORMATION STABILITY ANALYSIS

Based on the experimental analysis conducted in the previous stage, it was observed that there are certain instabilities in feature extraction, which may lead to a decrease in the reliability of the watermark. To address this issue, this paper proposes the phase method (10) as a transformation of the feature values to enhance the stability of the method.



The phase Transformation plays a crucial role in achieving stability in the extracted feature values. By applying this transformation, the instability of the feature values is minimized. The experimental analysis in this stage follows the results presented in Fig. 11 and Fig. 12, where the extracted feature values are subjected to seven types of attacks, each with three different intensities as shown in Table 1. The results of the analysis after applying the phase transformation for the airplane feature values are presented in Fig. 11 ( $P_V$ ) and Fig. 12 ( $P_H$ ). In the Figure (Fig. 11 and Fig. 12),  $P_V$  and  $P_H$  represent the vertical and horizontal transformed phase, respectively. If the colors in the image exhibit chaos, it indicates instability after the phase transformation. However, our experimental results demonstrate color consistency and simplification. This research confirms that our method exhibits optimal stability and reliability under various noise attacks. The original image and the representative colors for Type 1, Type 2, and Type 3 are consistent with those used in Fig. 9 and Fig. 10.

**B. EXPERIMENTAL RESULT**

1) EXPERIMENTAL RESULTS WITH DIFFERENT IMAGES AND WATERMARKS

This paper conducted extensive experimental analysis on the zero-watermark framework to evaluate the effectiveness and robustness of the proposed method. For this purpose, a diverse set of eight images, referred to as the host images, was carefully selected as shown in Fig. 7. Each image was subjected to seven different attack methods, with three different intensity types (such as Type 1, Type 2, Type 3) for each attack method, as described in Table 1. This experimental design enabled a comprehensive assessment of the method’s robustness under various noise attack conditions.

During the experimental phase, two distinct types of binary watermarks were employed to investigate their potential impact on watermark quality during embedding. Figure 6(a) represents the application of a complex binary watermark, while Figure 6(b) represents the application of a simple binary watermark. Through these experiments utilizing different watermarks, our objective was to ascertain whether the proposed method could maintain the quality of various watermarks under diverse noise attacks.

To quantify the accuracy of watermark extraction, we utilized Bit Error Rate (BER) and Normalized Correlation (NC) as reliable evaluation metrics. These metrics were chosen to gauge the accuracy of watermark extraction. The formulas for calculating BER and NC are provided as (18) and (19) respectively:

$$BER(W, W') = \frac{1}{i \times j} \sum_{i=1}^{Hight} \sum_{j=1}^{Width} (W_{i,j} \oplus W'_{i,j}) \quad (18)$$

where *Hight* and *Width* represent the sizes of the watermarks. Then,  $W$  is the original watermark and  $W'$  is the extracted watermark for comparison. The error rate can be calculated using the following formula. The smaller the BER value

**TABLE 1. Common types of noise attacks and corresponding abbreviation for three different levels of intensity.**

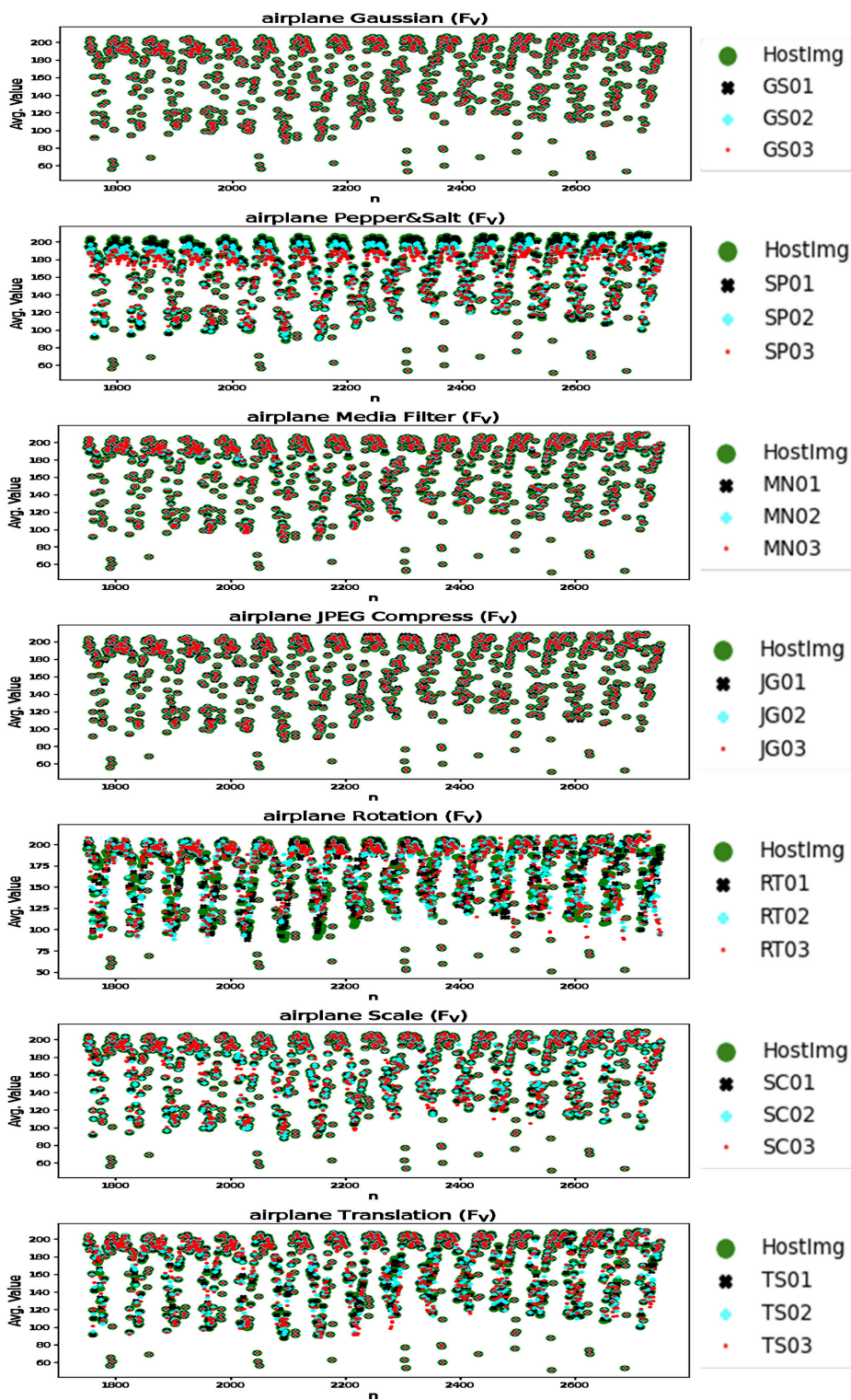
Attack Type	Strength of Attack	Abbr..	Avg PSNR(db)
Gaussian Noise	Type1:var= 1%	GS01	29.2245
	Type2:var= 5%	GS02	27.6098
	Type3:var=10%	GS03	24.7449
Pepper-and-Salt Noise	Type1:var= 1%	SP01	22.2530
	Type2:var= 5%	SP02	15.4123
	Type3:var=10%	SP03	12.6202
Median Filter	Type1:3 × 3	MN01	29.4245
	Type2:5 × 5	MN02	26.8480
	Type3:7 × 7	MN03	25.5086
JPEG Compression	Type1:Q=5%	JG01	38.0696
	Type2:Q=50%	JG02	32.1754
	Type3:Q=90%	JG03	27.6217
Rightward Rotation	Type1:angle = 5°	RT01	13.7854
	Type2:angle = 15°	RT02	11.0739
	Type3:angle = 20°	RT03	10.5650
Scaling Attack	Type1:var= 1/2	SC01	23.8281
	Type2:var= 1/4	SC02	24.9761
	Type3:var= 1/8	SC03	21.9607
Rightward Translation	Type1:ShiftPixels = 1	TS01	23.8962
	Type2:ShiftPixels = 5	TS02	17.2380
	Type3:ShiftPixels = 10	TS03	15.1121

\*Attack Type refers to the full name of the noise attack type.  
 \*Strength of Attack represents the three different levels of intensity types (Type1, Type2, Type3) for each Attack Type, along with explanations of the attack strength for these modes.  
 \*Abbr. is used to assign a unique number to each noise attack for clarity in the subsequent presentation of experimental results. For the same Attack Type, their initials are identical, and they are numbered sequentially as 01, 02, 03 according to the order of attack modes.  
 \*Avg. PSNR(db) represents the average Peak Signal-to-Noise Ratio (PSNR) values of the images from Fig. 7 after being subjected to the corresponding noise attacks.  
 \*PSNR is a metric used to measure the quality of an image after being subjected to noise attacks. A higher PSNR value indicates a better quality image compared to the original, while a lower value indicates poor quality due to the impact of noise attacks.

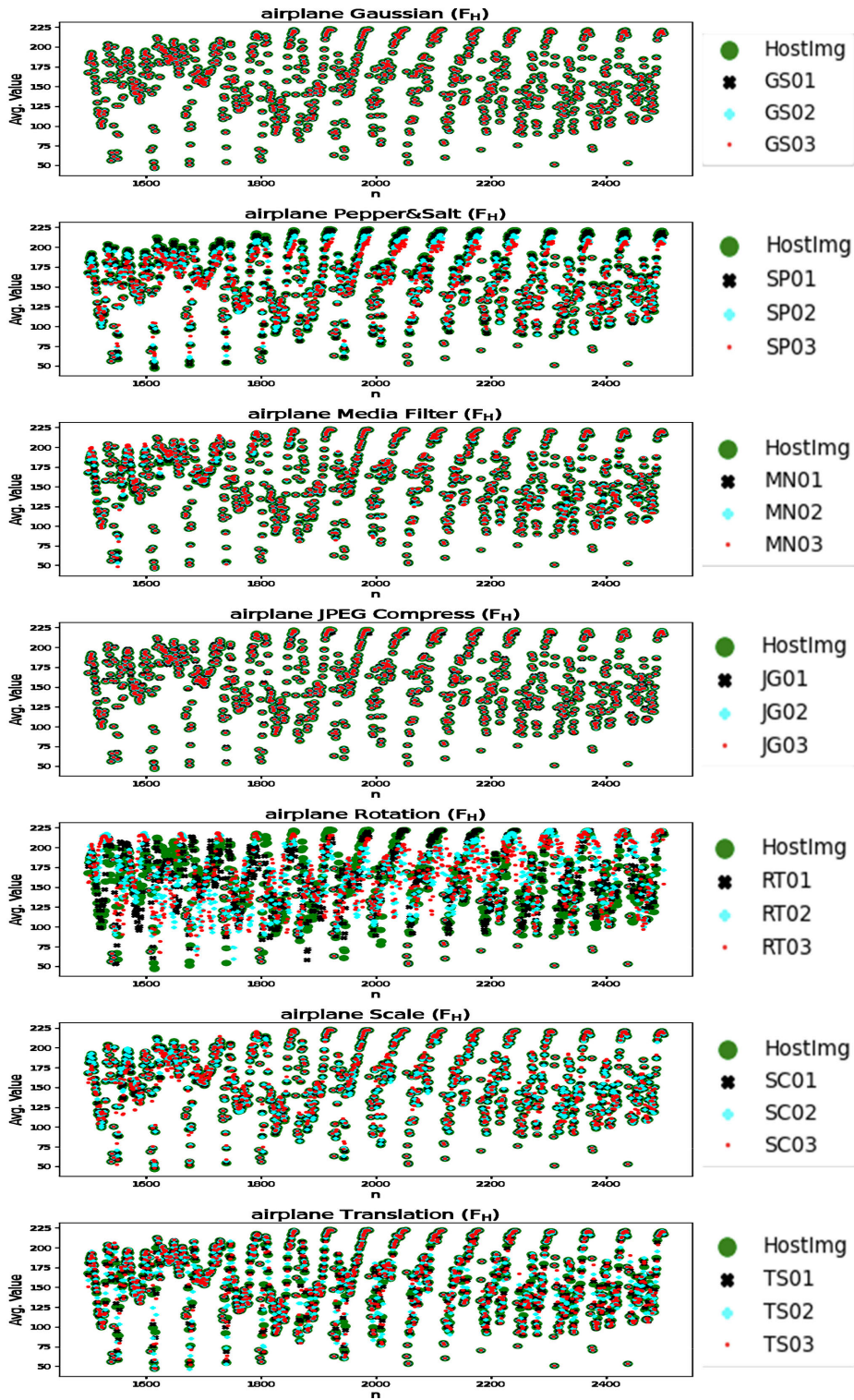
(between 0 and 1), the closer the extracted watermark data is to the original watermark data.

NC is a commonly used metric for evaluating the quality of watermarks, which measures the correlation between the original image and the extracted watermark during the embedding and extraction processes. NC is typically represented by a value ranging from 0 to 1, where a value closer to 1 indicates higher watermark quality, indicating a closer resemblance between the extracted watermark and the original watermark. By calculating the Normalized Correlation, the discriminability and extraction reliability of the watermark can be assessed to determine if the watermark has been affected or damaged.

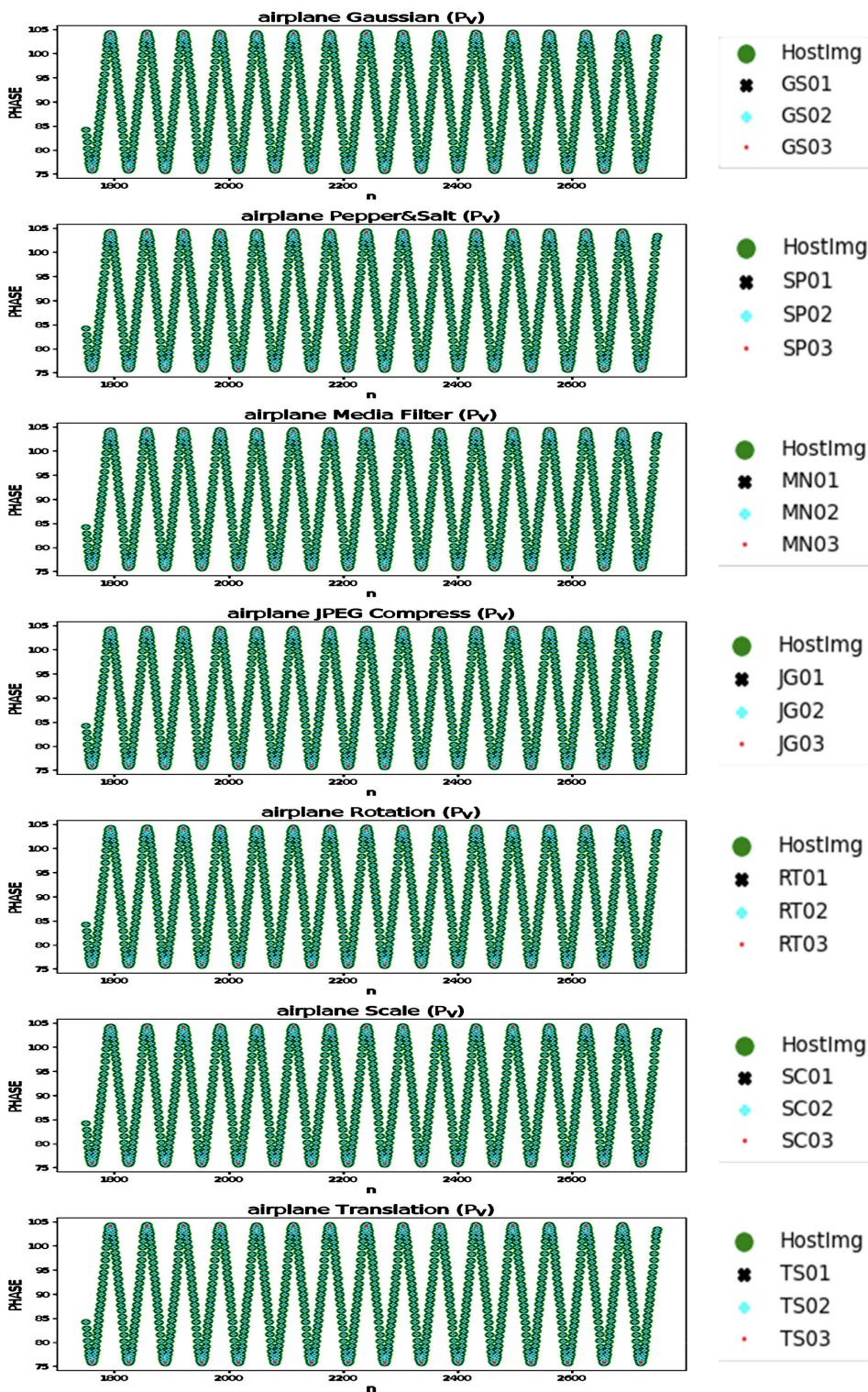
$$NC = \frac{1}{i \times j} \sum_{i=1}^{Hight} \sum_{j=1}^{Width} \overline{(W_{i,j} \oplus W'_{i,j})} \quad (19)$$



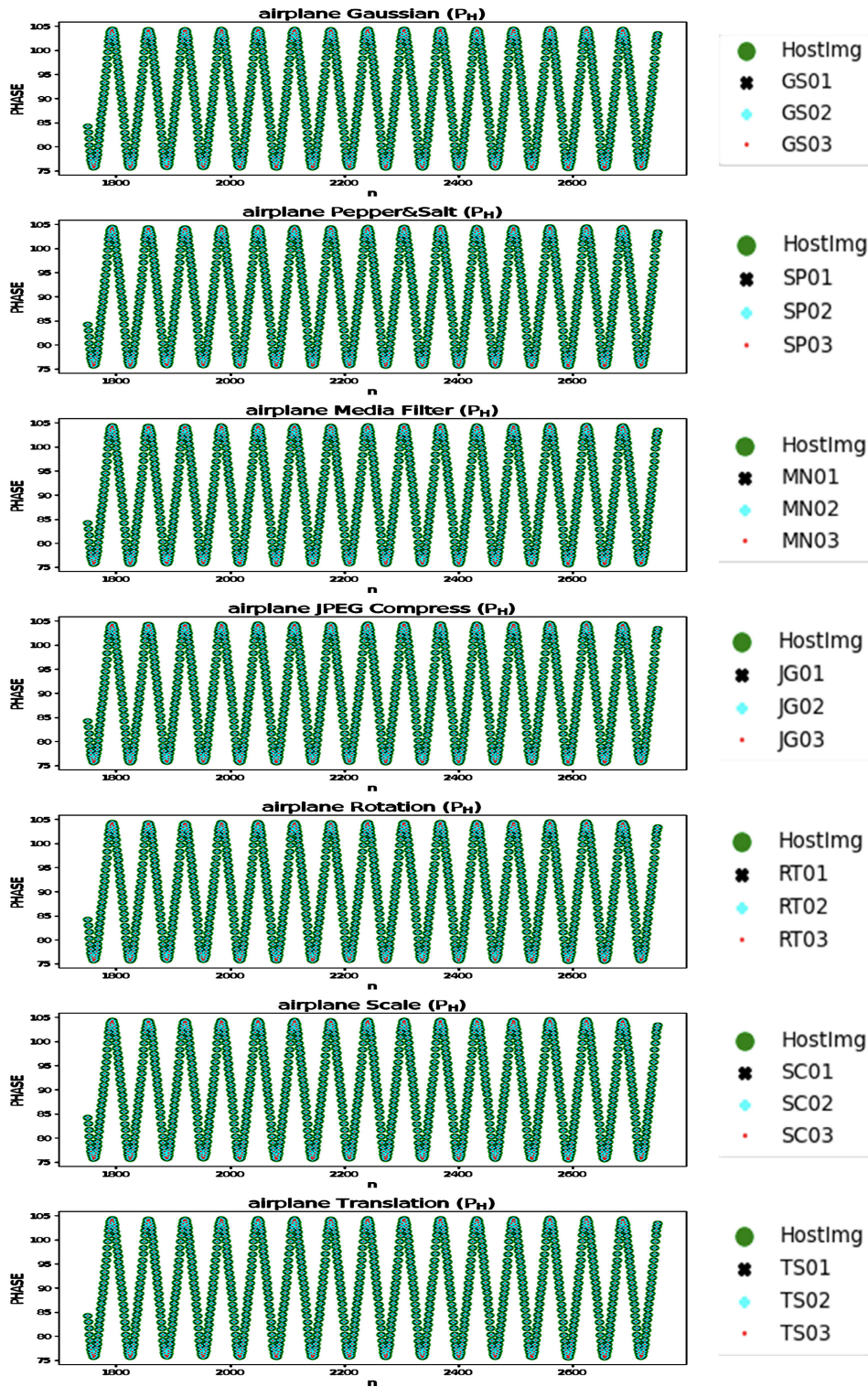
**FIGURE 9.** To analyze the Avg. values (mean) of the vertical ( $F_V$ ) directions of the “Airplane” image under various noise attacks, we will consider a range of values from 1500 to 3000. This analysis involves seven images, each representing seven distinct types of noise interference. In each image, a comparison is made between the original image and the image under various levels of attack intensity (the abbreviations for the attacks correspond to the explanations in Table 1, from top to bottom). The complexity of colors in each image indicates the severity of noise impact, with more intricate colors representing more pronounced effects of noise. For instance, in general attack type, the title of the image is “Airplane Peppers&Salt ( $F_V$ )” in the geometric attack type, and the title of the image is “Airplane Rotation ( $F_V$ )”



**FIGURE 10.** To analyze the Avg. values (mean) of the horizontal ( $F_H$ ) directions of the “Airplane” image under various noise attacks, we will consider a range of values from 1500 to 3000. This analysis involves seven images, each representing seven distinct types of noise interference. In each image, a comparison is made between the original image and the image under various levels of attack intensity (the abbreviations for the attacks correspond to the explanations in Table 1, from top to bottom). The complexity of colors in each image indicates the severity of noise impact, with more intricate colors representing more pronounced effects of noise. For instance, in general attack type, the title of the image is “Airplane Peppers&Salt ( $F_H$ )” in the geometric attack type, and the title of the image is “Airplane Rotation ( $F_H$ )”:



**FIGURE 11.** To analyze the PHASE values of the vertical ( $P_V$ ) directions of the “Airplane” image under various noise attacks, we will consider a range of values from 1500 to 3000. This analysis comprises seven images, each representing seven distinct types of noise interference. Within each image, a comparison is made between the original image and the image under various levels of attack intensity (the abbreviations for the attacks correspond to the explanations in Table 1, from top to bottom). The complexity of colors in each image indicates the severity of noise impact. However, the consistent color representation across these seven images illustrates the stability of the signal performance after the phase transformation.



**FIGURE 12.** To analyze the PHASE values of the horizontal ( $P_H$ ) directions of the “Airplane” image under various noise attacks, we will consider a range of values from 1500 to 3000. This analysis comprises seven images, each representing seven distinct types of noise interference. Within each image, a comparison is made between the original image and the image under various levels of attack intensity (the abbreviations for the attacks correspond to the explanations in Table 1, from top to bottom). The complexity of colors in each image indicates the severity of noise impact. However, the consistent color representation across these seven images illustrates the stability of the signal performance after the phase transformation.

**TABLE 2.** Experimental evaluation of BER/NC under 21 noise attacks using Fig. 6 (a) as binary watermark and Fig. 7 as digital image.

Image Name Abbr. Of Attack	Airplane	Baboon	Cameraman	Fishingboat	Goldhill	Lena	Moon Surface	Peppers
	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC
GS01	0.0002 / 1.	<b>0.</b> / <b>1.</b>	0. / 0.9995	<b>0.</b> / <b>1.</b>	0.0002 / 1.	<b>0.</b> / <b>1.</b>	<b>0.</b> / <b>1.</b>	0.0004 / 0.9995
GS02	0.0004 / 0.9990	0. / 0.9985	0.0002 / 1.	<b>0.</b> / <b>1.</b>	0.0004 / 0.9990	0.0002 / 0.9990	0.0002 / 0.9990	0.0007 / 0.9985
GS03	0.0004 / 0.9995	0.0014 / 0.9980	0.0007 / 1.	0.0014 / 1.	0.0017 / 0.9975	0.0007 / 0.9985	0.0014 / 0.9990	0.0017 / 0.9990
SP01	0.0053 / 0.9960	0.0051 / 0.9990	0.0041 / 0.9946	0.0043 / 0.9990	0.0048 / 0.9965	0.0046 / 0.9985	0.0036 / 0.9970	0.0039 / 0.9990
SP02	0.0065 / 0.9916	0.0068 / 0.9960	0.0056 / 0.9950	0.0063 / 0.9926	0.0058 / 0.9921	0.0061 / 0.9965	0.0073 / 0.9931	0.0046 / 0.9916
SP03	0.0048 / 0.9965	0.0070 / 0.9901	0.0068 / 0.9946	0.0098 / 0.9911	0.0091 / 0.9901	0.0063 / 0.9946	0.0058 / 0.9891	0.0095 / 0.9941
MN01	0.0017 / 0.9980	0.0021 / 0.9975	0.0017 / 0.9995	0.0012 / 0.9995	0.0017 / 0.9985	0.0007 / 1.	0.0004 / 1.	0.0009 / 0.9985
MN02	0.0024 / 0.9985	0.0026 / 0.9980	0.0019 / 0.9990	0.0014 / 0.9995	0.0019 / 0.9975	0.0009 / 0.9995	0.0002 / 0.9990	0.0014 / 0.9980
MN03	0.0026 / 0.9985	0.0041 / 0.9975	0.0019 / 0.9985	0.0014 / 0.9985	0.0021 / 0.9980	0.0007 / 0.9995	0.0004 / 0.9985	0.0017 / 0.9980
JG01	0.0014 / 0.9980	0.0014 / 0.9980	0. / 0.9995	0.0014 / 0.9995	0.0019 / 0.9980	0.0017 / 0.9980	<b>0.</b> / <b>1.</b>	0.0012 / 0.9990
JG02	0.0012 / 0.9980	0.0026 / 0.9980	0.0014 / 0.9995	0.0017 / 1.	0.0021 / 0.9980	0.0017 / 0.9985	0.0007 / 0.9995	0.0012 / 0.9990
JG03	0.0026 / 0.9995	0.0021 / 1.	0.0002 / 0.9995	0.0014 / 1.	0.0009 / 1.	0.0002 / 0.9995	<b>0.</b> / <b>1.</b>	0.0009 / 0.9985
RT01	0.0051 / 0.9901	0.0041 / 0.9960	0.0043 / 0.9941	0.0039 / 0.9960	0.0039 / 0.9965	0.0048 / 0.9975	0.0024 / 0.9970	0.0046 / 0.9970
RT02	0.0085 / 0.9911	0.0056 / 0.9946	0.0048 / 0.9916	0.0080 / 0.9995	0.0036 / 0.9955	0.0029 / 0.9980	0.0039 / 0.9975	0.0063 / 0.9896
RT03	0.0087 / 0.9897	0.0065 / 0.9946	0.0043 / 0.9921	0.0058 / 0.9970	0.0061 / 0.9965	0.0034 / 0.9960	0.0031 / 0.9965	0.0046 / 0.9931
SC01	0.0031 / 0.9990	0.0034 / 0.9980	0.0019 / 0.9990	0.0014 / 0.9995	0.0021 / 0.9980	0.0019 / 0.9995	<b>0.</b> / <b>1.</b>	0.0014 / 0.9990
SC02	0.0024 / 0.9985	0.0056 / 0.9975	0.0019 / 0.9980	0.0012 / 0.9995	0.0021 / 0.9970	0.0021 / 0.9990	0.0004 / 0.9985	0.0012 / 0.9985
SC03	0.0034 / 0.9970	0.0046 / 0.9985	0.0026 / 0.9980	0.0017 / 0.9990	0.0031 / 0.9955	0.0021 / 0.9995	0.0007 / 0.9990	0.0024 / 0.9990
TS01	0.0019 / 0.9955	0.0039 / 0.9975	0.0026 / 0.9985	0.0021 / 0.9990	0.0024 / 0.9965	0.0024 / 0.9970	0.0002 / 0.9985	0.0012 / 0.9970
TS02	0.0051 / 0.9936	0.0061 / 0.9950	0.0034 / 0.9936	0.0051 / 0.9960	0.0041 / 0.9965	0.0029 / 0.9980	0.0019 / 0.9955	0.0048 / 0.9941
TS03	0.0063 / 0.9896	0.0048 / 0.9946	0.0036 / 0.9931	0.0039 / 0.9975	0.0048 / 0.9955	0.0039 / 0.9960	0.0039 / 0.9970	0.0048 / 0.9896

\*Abbr. of Attack means Abbreviations for various noise attacks, please refer to Table 1.

\*Image Name represents the eight images presented in Fig. 7.

\*BER represents the evaluation value of extracting watermarks from various noisy images, and the smaller the value, the better the quality of the watermark. The calculation method for this evaluation approach can be referred to in (18).

\*NC represents the evaluation value for extracting watermarks from various noisy images. The closer the value is to 1, the better the quality of the watermark. Conversely, as the evaluation value approaches 0, it indicates poorer watermark quality. The calculation method for this evaluation approach can be referred to in (19).

\*The highlighted portions in bold font refer to the locations where, under the corresponding noise attacks, the optimal data values are presented for the respective digital images. This includes achieving perfect performance in terms of both BER and NC.

\*The extracted watermarks from Table 2 are sequentially displayed in a symmetrical manner in Fig. 13, providing a mutually corresponding reference and experimental validation.

where the dimensions of the watermarks are typically denoted by their *Height* and *Width* values, which indicate the size or spatial extent of the watermarks within an image. The original watermark is symbolized as  $W$ , representing the embedded

watermark in the host image. To assess the quality and fidelity of the watermarking process, the extracted watermark, used for comparison and evaluation purposes, is represented as  $W'$ . This extracted watermark serves as a reference to measure

Image Name Abbr. Of Attack	Airplane	Baboon	Camerman	Fishingboat	Goldhill	Lena	Moon Surface	Peppers
GS01								
GS02								
GS03								
SP01								
SP02								
SP03								
MN01								
MN02								
MN03								
JG01								
JG02								
JG03								
RT01								
RT02								
RT03								
SC01								
SC02								
SC03								
TS01								
TS02								
TS03								

**FIGURE 13.** The extracted watermarks from Table 2 are symmetrically positioned in relation to the various digital images from Fig. 7 and the different attacks from Table 1. These watermarks are displayed in this image to provide experimental validation and reference, aligning with the corresponding positions in Table 2.

TABLE 3. Experimental evaluation of BER/NC under 21 noise attacks using Fig. 6 (b) as binary watermark.

Image Name Abbr. Of Attack	Airplane	Baboon	Cameraman	Fishingboat	Goldhill	Lena	Moon Surface	Peppers
	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC	BER / NC
GS01	<b>0.</b> / <b>1.</b>	<b>0.</b> / <b>1.</b>	0.0002 / 0.9997	<b>0.</b> / <b>1.</b>	<b>0.</b> / <b>1.</b>	<b>0.</b> / <b>1.</b>	0.0002 / 1.	0.0002 / 1.
GS02	0.0004 / 0.9997	0.0002 / 0.9980	<b>0.</b> / <b>1.</b>	<b>0.</b> / <b>1.</b>	0.0007 / 0.9992	0.0004 / 0.9994	<b>0.</b> / <b>1.</b>	0.0007 / 0.9992
GS03	0.0007 / 0.9994	0.0002 / 0.9983	0.0002 / 0.9997	<b>0.</b> / <b>1.</b>	0.0017 / 0.9983	0.0007 / 0.9992	0. / 0.9986	0.0012 / 0.9989
SP01	0.0029 / 0.9969	0.0014 / 0.9983	0.0056 / 0.9941	0.0012 / 0.9986	0.0034 / 0.9963	0.0034 / 0.9963	0.0039 / 0.9955	0.0019 / 0.9980
SP02	0.0097 / 0.9899	0.0031 / 0.9961	0.0053 / 0.9947	0.0065 / 0.9927	0.0070 / 0.9930	0.0043 / 0.9958	0.0070 / 0.9924	0.0053 / 0.9944
SP03	0.0043 / 0.9958	0.0043 / 0.9932	0.0063 / 0.9935	0.0097 / 0.9896	0.0102 / 0.9896	0.0078 / 0.9927	0.0091 / 0.9880	0.0061 / 0.9952
MN01	0.0012 / 0.9989	0.0029 / 0.9966	0.0009 / 0.9992	0.0002 / 0.9997	0.0009 / 0.9989	0.0004 / 0.9997	<b>0.</b> / <b>1.</b>	0.0007 / 1.
MN02	0.0012 / 0.9989	0.0026 / 0.9977	0.0007 / 0.9992	0.0007 / 0.9992	0.0021 / 0.9975	0.0009 / 0.9992	0.0007 / 0.9992	0.0009 / 0.9997
MN03	0.0012 / 0.9989	0.0026 / 0.9969	0.0009 / 0.9989	0.0019 / 0.9980	0.0017 / 0.9980	0.0009 / 0.9992	0.0009 / 0.9989	0.0009 / 0.9994
JG01	0.0014 / 0.9989	0.0024 / 0.9966	0.0009 / 0.9992	0.0002 / 0.9997	0.0017 / 0.9983	0.0017 / 0.9983	0. / 0.9994	0.0012 / 0.9992
JG02	0.0009 / 0.9989	0.0009 / 0.9972	0.0007 / 0.9994	0.0002 / 0.9997	0.0012 / 0.9986	0.0014 / 0.9989	0.0004 / 0.9994	0.0004 / 1.
JG03	0.0004 / 0.9997	0.0002 / 0.9994	0.0002 / 0.9997	<b>0.</b> / <b>1.</b>	<b>0.</b> / <b>1.</b>	0.0004 / 0.9994	0.0004 / 1.	0.0009 / 0.9992
RT01	0.0073 / 0.9932	0.0046 / 0.9952	0.0061 / 0.9941	0.0053 / 0.9944	0.0029 / 0.9969	0.0034 / 0.9961	0.0034 / 0.9961	0.0021 / 0.9977
RT02	0.0068 / 0.9927	0.0041 / 0.9935	0.0070 / 0.9932	0.0026 / 0.9972	0.0041 / 0.9961	0.0021 / 0.9975	0.0026 / 0.9969	0.0092 / 0.9901
RT03	0.0092 / 0.9907	0.0061 / 0.9938	0.0065 / 0.9932	0.0043 / 0.9952	0.0041 / 0.9963	0.0046 / 0.9949	0.0036 / 0.9958	0.0058 / 0.9947
SC01	0.0019 / 0.9994	0.0026 / 0.9975	0.0012 / 0.9992	0.0004 / 0.9997	0.0019 / 0.9986	0.0017 / 0.9992	0.0002 / 0.9997	0.0007 / 0.9994
SC02	0.0009 / 0.9992	0.0024 / 0.9972	0.0012 / 0.9986	0.0009 / 0.9992	0.0024 / 0.9972	0.0017 / 0.9983	0.0014 / 0.9983	0.0012 / 0.9994
SC03	0.0021 / 0.9977	0.0024 / 0.9972	0.0024 / 0.9977	0.0014 / 0.9986	0.0031 / 0.9966	0.0017 / 0.9986	0.0007 / 0.9992	0.0009 / 0.9994
TS01	0.0029 / 0.9966	0.0034 / 0.9969	0.0014 / 0.9983	0.0014 / 0.9983	0.0024 / 0.9983	0.0026 / 0.9969	0.0007 / 0.9992	0.0024 / 0.9983
TS02	0.0058 / 0.9938	0.0056 / 0.9947	0.0061 / 0.9938	0.0034 / 0.9963	0.0026 / 0.9963	0.0041 / 0.9961	0.0026 / 0.9969	0.0056 / 0.9947
TS03	0.0087 / 0.9913	0.0070 / 0.9941	0.0061 / 0.9941	0.0024 / 0.9975	0.0041 / 0.9975	0.0048 / 0.9944	0.0031 / 0.9969	0.0087 / 0.9913

\*Abbr. of Attack means Abbreviations for various noise attacks, please refer to Table 1.

\*Image Name represents the eight images presented in Fig. 7.

\*BER represents the evaluation value of extracting watermarks from various noisy images, and the smaller the value, the better the quality of the watermark. The calculation method for this evaluation approach can be referred to in 18.

\*NC represents the evaluation value for extracting watermarks from various noisy images. The closer the value is to 1, the better the quality of the watermark. Conversely, as the evaluation value approaches 0, it indicates poorer watermark quality. The calculation method for this evaluation approach can be referred to in 19.

\*The highlighted portions in bold font refer to the locations where, under the corresponding noise attacks, the optimal data values are presented for the respective digital images. This includes achieving perfect performance in terms of both BER and NC.

\* The extracted watermarks from Table 3 are sequentially displayed in a symmetrical manner in Fig. 14, providing a mutually corresponding reference and experimental validation..

the effectiveness of the watermark extraction algorithm and to determine the level of distortion or alteration that may have occurred during the embedding and extraction processes.

In order to comprehensively and meticulously present our research findings, we have organized the experimental results

into two distinct tables, namely Table 2 and Table 3, based on Fig. 13 and Fig. 14 correspondingly. These tables are utilized to showcase the experimental data, and the watermark samples extracted from Table 2 and Table 3 are symmetrically displayed in Fig. 13 and Fig. 14, respectively.



Image Name Abbr. Of Attack	Airplane	Baboon	Cameraman	Fishingboat	Goldhill	Lena	Moon Surface	Peppers
GS01	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
GS02	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
GS03	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
SP01	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
SP02	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
SP03	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
MN01	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
MN02	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
MN03	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
JG01	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
JG02	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
JG03	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
RT01	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
RT02	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
RT03	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
SC01	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
SC02	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
SC03	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
TS01	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
TS02	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>
TS03	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>	<b>A</b>

**FIGURE 14.** The extracted watermarks from Table 3 are symmetrically positioned in relation to the various digital images from Fig. 7 and the different attacks from Table 1. These watermarks are displayed in this image to provide experimental validation and reference, aligning with the corresponding positions in Table 3.

Table 2 represents the experimental data results obtained using complex watermarks under various attack scenarios. The corresponding extracted watermarks are symmetrically displayed in Fig 13. Conversely, Table 3 presents the experimental data results using simple watermarks, with the corresponding extracted watermarks symmetrically showcased in Fig 14. These tables offer a comprehensive and detailed performance analysis of the proposed method under distinct attack scenarios.

Based on the results in Table 2, the BER values for 21 types of attacks across the 7 categories range from 0 to 0.0098, while the NC values range from 0.9896 to 1. Similarly, Table 3 demonstrates that within the same 7 categories and 21 types of attacks, the BER values fall within the range of 0 to 0.0102, and the NC values range from 0.9880 to 1. These data indicate that regardless of whether a complex-type or a simple-type binary watermark is embedded in both experimental scenarios, the quality of the extracted watermark remains robust.

Furthermore, in this study, the majority of evaluation values exhibit BER values approaching 0 and NC values approaching 1. This provides evidence of the resilience of our proposed method in extracting high-quality watermarks while effectively resisting noise interference.

## 2) EXPERIMENTAL RESULTS COMPARISON WITH RELATED WORKS

In this section, we compare our proposed method with the approaches presented by Wang et al. [19], Chen et al. [20], Wang et al. et al. [21], and Huang et al. [22], all of which utilize zero-watermarking frameworks. The comparison is based on the evaluation of watermark quality using the NC metric [19]. The NC evaluation results of the method proposed by [19] are presented in Table 4. The NC evaluation results of the method proposed by [20] are presented in Table 5. The NC evaluation results of the method proposed by [21] are presented in Table 6 and the NC evaluation results of the method proposed by [22] are presented in Table 7. For all the comparisons of NC data mentioned above, we utilized the average NC values of all images in the experiment under the same type of attack as our basis for comparison.

According to the NC evaluation data in Table 4, our proposed method demonstrates a significant disparity in the comparative analysis. It is worth noting that our method achieves a minimum gap of 0.0004 and even reaches a maximum gap of 0.1426. Among various attack scenarios, the lowest gap is observed in JPEG compression attacks, while the highest disparity is encountered in noise attacks under the context of media filtering. In comparison to the [19] method, our proposed approach exhibits more stability and reliability in terms of median filtering.

Based on the NC evaluation data presented in Table 5, our proposed method achieved a numerical value of 0.0054 for the lowest gap and even reached 0.1152 for the highest gap. This implies that our proposed method exhibits greater

TABLE 4. Comparing [19] on the NC metric.

Attacks	Method in [19]	Proposed
Gaussian noise (var=0.15)	0.9493	0.9991
Gaussian noise (var=0.20)	0.9415	0.9987
Gaussian noise (var=0.25)	0.9365	0.9976
Median filtering (7 × 7)	0.9874	0.9978
Median filtering (9 × 9)	0.8558	0.9978
Median filtering (11 × 11)	<b>0.8549</b>	<b>0.9975</b>
Salt & peppers noise(var=0.15)	0.9667	0.9904
Salt & peppers noise(var=0.20)	0.9606	0.9906
Salt & peppers noise(var=0.25)	0.9527	0.9899
JPEG compression (Q=1%)	0.9500	0.9968
JPEG compression (Q=5%)	0.9707	0.9977
JPEG compression (Q=20%)	0.9941	0.9981
Rotation attack (angle = 5° )	0.9672	0.9960
Rotation attack (angle = 10° )	0.9532	0.9954
Rotation attack (angle = 30° )	0.9204	0.9944
Scaling attack (1/32)	0.8831	0.9961
Scaling attack (1/16)	0.9765	0.9971
Scaling attack (1/8)	0.9927	0.9982
Translation attack (5 pixels)	0.9205	0.9955
Translation attack (6 pixels)	0.9058	0.9962
Translation attack (7 pixels)	0.8914	0.9960

\*Gaussian noise: is adjusted through variable manipulation to control the intensity of noise attacks, introducing Gaussian-distributed noise.

\*Median filtering: for each pixel, the pixel values within a specified neighborhood are sorted, and the value at the middle position after sorting is chosen as the new pixel value. The specified neighborhood refers to the strength of the attack, such as a 3x3 region.

\* Salt & Peppers noise:we degraded the quality by changing the noise densities. Through variable to change the intensity of noise attacks.

\* JPEG compression: this attack with different quality factors of 1%, 5%, 10%, 20% were applied to the host images.

\*Rotation attack: rotate around the center point in a clockwise direction, where the angle corresponds to the intensity of the attack.

\*Scaling attack: scale the image according to the intensity of the attack, and then resize it back to its original dimensions.

\*Translation attack: translate the host images from left to right by n pixels. n is the intensity of the attack.

\*The bold highlighted portion represents a group with larger spacing in the data comparison within the table.

stability in addressing diverse scaling attack scenarios compared to [20].

According to the NC evaluation data presented in Table 6, our proposed method demonstrates a range of gap values from 0.0006 to a remarkable 0.1821 when compared to the method [21]. However, it's worth noting that in the case of the Scaling attack (0.75), our method exhibits a slightly lower performance by 0.0006 compared to method [21]. Nevertheless, considering the overall performance and stability, our method still maintains a significant advantage over most of the compared data, as indicated by the results.

In Table 7, our proposed method is compared with method [22], with a minimum gap value of 0.0009 and a maximum gap value of up to 0.0704. Despite this, we show a slight disadvantage of 0.0014 compared to method [22] in the JPEG compression (Q=30%) attack. However, when considering the overall comparison of attacks, our data still demonstrates excellent performance across various aspects.

Upon reviewing the comparisons in Table 4 to Table 7 against other zero-watermark methods, it is evident that our

TABLE 5. Comparing [20] on the NC metric.

Attacks	Method in [20]	Proposed
Gaussian noise (var=0.001)	0.9837	1.
Gaussian noise (var=0.02)	0.9236	0.9998
Gaussian noise (var=0.05)	0.8964	0.9998
Median filtering (3 × 3)	0.9752	0.9984
Median filtering (4 × 4)	0.9482	0.9982
Salt & peppers noise(var=0.001)	0.9871	0.9962
Salt & peppers noise(var=0.02)	0.9568	0.9947
Salt & peppers noise(var=0.1)	0.9018	0.9926
JPEG compression (Q=20%)	0.9500	0.9981
JPEG compression (Q=50%)	0.9707	0.9980
JPEG compression (Q=90%)	0.9941	0.9995
Rotation attack (angle = 0.5° )	0.9672	0.9969
Scaling attack (0.25)	<b>0.8831</b>	<b>0.9983</b>
Scaling attack (0.9)	0.9765	0.9987
Scaling attack (1.1)	0.9927	0.9989
Scaling attack (2)	0.9927	0.9989

\*For an explanation of the types of attacks, please refer to Table 4 comment.

\*The bold highlighted portion represents a group with larger spacing in the data comparison within the table.

TABLE 6. Comparing [21] on the NC metric.

Attacks	Method in [21]	Proposed
Gaussian noise (var=0.01)	0.9298	0.9999
Gaussian noise (var=0.03)	0.8583	0.9998
Gaussian noise (var=0.04)	<b>0.8175</b>	<b>0.9996</b>
Median filtering (2 × 2)	0.9730	1.
Median filtering (4 × 4)	0.9645	0.9982
Median filtering (16 × 16)	0.9131	0.9981
Salt & peppers noise(var=0.02)	0.9807	0.9950
JPEG compression (Q=10%)	0.9713	0.9986
JPEG compression (Q=20%)	0.973	0.9981
JPEG compression (Q=30%)	0.9872	0.9984
Rotation attack (angle =15° )	0.9783	0.9944
Rotation attack (angle =45° )	0.9663	0.9948
Rotation attack (angle =75° )	0.9777	0.9955
Scaling attack (0.25)	0.9933	0.9983
Scaling attack (0.75)	1.	0.9994
Scaling attack (1.25)	0.9975	0.9995
Translation attack (10 pixels)	0.9642	0.9941
Translation attack (20 pixels)	0.9702	0.9937
Translation attack (30 pixels)	0.9621	0.9930

\*For an explanation of the types of attacks, please refer to Table 4 comment.

\*The bold highlighted portion represents a group with larger spacing in the data comparison within the table.

proposed approach consistently maintains a stable quality with an average NC value of at least 0.9902 across various noise attack scenarios. In essence, the experimental results affirm the robustness of our proposed method, as it consistently upholds a stable level of quality even amidst diverse noise disturbances.

In the experiment, we selected different types of images to conduct tests, including complex, smooth, landscape, and portrait types, among others. We also applied strong noise

TABLE 7. Comparing [22] on the NC metric.

Attacks	Method in [22]	Proposed
Gaussian noise (var=0.05)	0.9828	0.9993
Gaussian noise (var=0.35)	0.9672	0.9980
Gaussian noise (var=0.5)	0.9688	0.9969
Median filtering (3 × 3)	0.9936	0.9984
Median filtering (5 × 5)	0.9799	0.9987
Median filtering (7 × 7)	0.9673	0.9978
JPEG compression (Q=2%)	0.9812	0.9982
JPEG compression (Q=15%)	0.9983	0.9992
JPEG compression (Q=30%)	0.9998	0.9984
Rotation attack (angle =5%)	0.9890	0.9955
Rotation attack (angle =20%)	0.9703	0.9944
Rotation attack (angle =35%)	0.9734	0.9946
Scaling attack (0.125)	0.9672	0.9982
Scaling attack (0.25)	0.9781	0.9983
Scaling attack (0.5)	0.9921	0.9990
Translation attack (25 pixels)	0.9844	0.9938
Translation attack (80 pixels)	0.9610	0.9912
Translation attack (256 pixels)	<b>0.9198</b>	<b>0.9902</b>

\*For an explanation of the types of attacks, please refer to Table 4 comment. But the rotation attack is an exception.

\*Rotation attack: in this table are clockwise and the attack intensity is expressed as a percentage, representing the rotation angle.

\*The bold highlighted portion represents a group with larger spacing in the data comparison within the table.

attacks to these eight types of images and evaluated the watermark quality using the BER. The results showed excellent performance of our method in all cases.

Furthermore, we compared our method with other relevant watermarking frameworks. Our method outperformed the comparison methods in both general and geometric attack scenarios. Summarizing the above experimental results, our proposed method demonstrated outstanding quality and stability in handling diverse attack scenarios.

## V. CONCLUSION

The proposed framework in this article possesses several advantages: (a) It utilizes a zero-watermarking architecture, ensuring the integrity of the original image without any destructive embedding. (b) By leveraging the characteristics of Arnold’s transform and the diverse selection of scrambling coefficients, it enhances the invisibility and security of the watermark. (c) The stability of embedding/extracting the watermark is further enhanced by transforming the feature values through phase conversion. (d) The Golden Key is encrypted/decrypted using the RSA asymmetric encryption algorithm, thereby enhancing the system’s resistance to tampering and the attribution of copyright authentication. Our proposed approach demonstrates excellent performance in various types of general noise/geometric noise, under different attack intensities, as evidenced by experimental results and comparisons with other methods.

In conclusion, the novel, robust, and secure watermarking method presented in this study exhibits robust resilience and security, making it suitable for verifying digital images against unauthorized use and tampering. In the future, we will

further optimize and expand this method to cope with evolving security threats and attack techniques. We will also face the challenge of integrating this technology into applications such as multi-media or document.

## REFERENCES

- [1] F. Boland, J. J. O'Ruanaidh, and C. Dautzenberg, "Watermarking digital images for copyright protection," in *Proc. Int. Conf. Image Process.*, Jul. 1995, pp. 326–330.
- [2] I. J. Cox and M. L. Miller, "The first 50 years of electronic watermarking," *EURASIP J. Adv. Signal Process.*, vol. 2002, no. 2, Feb. 2002, Art. no. 820936.
- [3] S.-J. Lee and S.-H. Jung, "A survey of watermarking techniques applied to multimedia," in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, Jun. 2001, pp. 272–277.
- [4] X. Liu, L. Jieting, Y. Wang, J. Du, B. Zou, and Y. Chen, "Discriminative and robust zero-watermarking scheme based on completed local binary pattern for authentication and copyright identification of medical images," *Proc. SPIE*, vol. 10579, Mar. 2018, Art. no. 105791I.
- [5] A. A. Elrowayati, M. A. Alrshah, M. F. L. Abdullah, and R. Latip, "HEVC watermarking techniques for authentication and copyright applications: Challenges and opportunities," *IEEE Access*, vol. 8, pp. 114172–114189, 2020.
- [6] V. S. Verma and R. K. Jha, "An overview of robust digital image watermarking," *IETE Tech. Rev.*, vol. 32, no. 6, pp. 479–496, Jun. 2015.
- [7] M. Moosazadeh and A. Andalib, "A new robust color digital image watermarking algorithm in DCT domain using genetic algorithm and coefficients exchange approach," in *Proc. 2nd Int. Conf. Web Res. (ICWR)*, Apr. 2016, pp. 19–24.
- [8] L. An, X. Gao, X. Li, D. Tao, C. Deng, and J. Li, "Robust reversible watermarking via clustering and enhanced pixel-wise masking," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3598–3611, Aug. 2012.
- [9] H. Lu, R. Shen, and F.-L. Chung, "Fragile watermarking scheme for image authentication," *Electron. Lett.*, vol. 39, no. 12, p. 898, 2003.
- [10] M. Botta, D. Cavagnino, and V. Pomponiu, "Image fragile watermarking through quaternion linear transform in secret space," *J. Imag.*, vol. 3, no. 3, p. 34, Aug. 2017.
- [11] H. Zhang, C. Wang, and X. Zhou, "Fragile watermarking for image authentication using the characteristic of SVD," *Algorithms*, vol. 10, no. 1, p. 27, Feb. 2017.
- [12] J. Yang, K. Hu, X. Wang, H. Wang, Q. Liu, and Y. Mao, "An efficient and robust zero watermarking algorithm," *Multimedia Tools Appl.*, vol. 81, no. 14, pp. 20127–20145, Mar. 2022.
- [13] C. Wang, X. Wang, Z. Xia, and C. Zhang, "Ternary radial harmonic Fourier moments based robust stereo image zero-watermarking algorithm," *Inf. Sci.*, vol. 470, pp. 109–120, Jan. 2019.
- [14] K. Chaitanya, E. S. Reddy, and K. G. Rao, "Digital color image watermarking in RGB planes using DWT-DCT-SVD coefficients," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 2413–2417, 2014.
- [15] Z. Xia, X. Wang, C. Wang, C. Wang, B. Ma, Q. Li, M. Wang, and T. Zhao, "A robust zero-watermarking algorithm for lossless copyright protection of medical images," *Appl. Intell.*, vol. 52, no. 1, pp. 607–621, May 2021.
- [16] A. Anand and A. K. Singh, "Watermarking techniques for medical data authentication: A survey," *Multimedia Tools Appl.*, vol. 80, no. 20, pp. 30165–30197, Apr. 2020.
- [17] X. Wang, D. Huang, and Z. Zhang, "A robust zero-watermarking algorithm for vector digital maps based on statistical characteristics," *J. Softw.*, vol. 7, no. 10, p. 2349, Oct. 2012.
- [18] Q. Wen, T. Sun, and S. Wang, "Concept and application of zero-watermark," *Acta Electron. Sinica*, vol. 31, pp. 214–216, Jan. 2003.
- [19] R. Wang, H. Shaocheng, P. Zhang, M. Yue, Z. Cheng, and Y. Zhang, "A novel zero-watermarking scheme based on variable parameter chaotic mapping in NSPD-DCT domain," *IEEE Access*, vol. 8, pp. 182391–182411, 2020.
- [20] T. Chen, Z. Qiu, G. Xie, L. Yuan, S. Duan, H. Guo, D. Fu, and H. Huang, "A image copyright protection method using zero-watermark by blockchain and IPFS," *J. Inf. Hiding Privacy Protection*, vol. 3, no. 3, pp. 131–142, 2021.
- [21] B. Wang, S. Jiawei, W. Wang, and P. Zhao, "Image copyright protection based on blockchain and zero-watermark," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2188–2199, Jul. 2022.
- [22] T. Huang, J. Xu, S. Tu, and B. Han, "Robust zero-watermarking scheme based on a depthwise overparameterized VGG network in healthcare information security," *Biomed. Signal Process. Control*, vol. 81, Mar. 2023, Art. no. 104478.
- [23] D. R. I. M. Setiadi, S. Rustad, P. N. Andono, and G. F. Shidik, "Digital image steganography survey and investigation (goal, assessment, method, development, and dataset)," *Signal Process.*, vol. 206, May 2023, Art. no. 108908.
- [24] L. Agilandeswari, M. Prabukumar, and F. A. Alenizi, "A robust semi-fragile watermarking system using pseudo-Zernike moments and dual tree complex wavelet transform for social media content authentication," *Multimedia Tools Appl.*, Apr. 2023.
- [25] B. Khesin, S. Tabachnikov, A. Givental, Y. Sinai, S. Smale, M. Sevryuk, A. Khovanskii, A. Varchenko, and M. Berry, "Tribute to Vladimir Arnold," *Notices Amer. Math. Soc.*, vol. 59, no. 3, p. 378, Mar. 2012.
- [26] M. Mishra, A. R. Routray, and S. Kumar, "High security image steganography with modified Arnold's cat map," *Int. J. Comput. Appl.*, vol. 37, no. 9, pp. 16–20, Jan. 2012.
- [27] C.-T. Ku, K.-C. Hung, T.-C. Wu, and H.-S. Wang, "Wavelet-based ECG data compression system with linear quality control scheme," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 6, pp. 1399–1409, Jun. 2010.
- [28] J.-H. Hsieh, R.-C. Lee, K.-C. Hung, and M.-J. Shih, "Rapid and coding-efficient SPIHT algorithm for wavelet-based ECG data compression," *Integration*, vol. 60, pp. 248–256, Jan. 2018.
- [29] Wikipedia Contributors. (Jan. 31, 2022). *Fisher-Yates Shuffle*. [Online]. Available: [https://en.wikipedia.org/wiki/Fisher-Yates\\_shuffle](https://en.wikipedia.org/wiki/Fisher-Yates_shuffle)
- [30] S. Sattolo, "An algorithm to generate a random cyclic permutation," *Inf. Process. Lett.*, vol. 22, no. 6, pp. 315–317, May 1986.
- [31] A. OluAde-Ibijola, "A simulated enhancement of Fisher-Yates algorithm for shuffling in virtual card games using domain-specific data structures," *Int. J. Comput. Appl.*, vol. 54, no. 11, pp. 24–28, Sep. 2012.
- [32] S. Saeed, M. S. Umar, M. A. Ali, and M. Ahmad, "Fisher-Yates chaotic shuffling based image encryption," Oct. 2014, *arXiv:1410.7540*.
- [33] N. Miyaho, Y. Ueno, S. Suzuki, K. Mori, and K. Ichihara, "Study of a secure backup network mechanism for disaster recovery and practical network applications," *Int. J. Adv. Netw. Services*, vol. 3, no. 1, pp. 273–285, 2010.
- [34] M. Cobb. (Nov. 2021). What is the RSA algorithm? Definition from SearchSecurity. SearchSecurity. [Online]. Available: <https://www.techtarget.com/searchsecurity/definition/RSA>
- [35] B. Kaliski, "Public-key cryptography standards (PKCS)# 8: Private-key information syntax specification version 1.2," Internet Eng. Task Force (IETF), 2008.



**HSIU-CHI TSENG** received the degree in information management from the National Kaohsiung University of Science and Technology, in 2000, where she is currently pursuing the Ph.D. degree with the Miniature System Integration Laboratory. She is also a senior engineer responsible for information system planning with telecommunications company. Her research interests include image processing, information hiding, virtual networks, and cryptographic algorithms.



**KING-CHU HUNG** received the Ph.D. degree in electrical engineering from the National Cheng-Kung University, in 1987. He was an Associate Professor with the Department of Electrical Engineering, National Central University, and the Department of Electrical Engineering, I-Shou University. Currently, he is a Professor with the Department of Computer and Communication Engineering, National Kaohsiung University of Science and Technology. He has published over

30 journals and conference papers. His research interests include signal processing, very large-scale integration circuits, image processing, and coffee roasting technology.

...