

Article

Joint Image Encryption and Screen-Cam Robust Two Watermarking Scheme

Weitong Chen ^{1,2,3}, Na Ren ^{1,2,3,*}, Changqing Zhu ^{1,2,3}, Anja Keskinarkaus ⁴ , Tapio Seppänen ⁴ and Qifei Zhou ^{1,2,3} 

¹ Key Laboratory of Virtual Geographic Environment, Nanjing Normal University, Ministry of Education, Nanjing 210023, China; 171301018@njnu.edu.cn (W.C.); 09322@njnu.edu.cn (C.Z.); 181301014@njnu.edu.cn (Q.Z.)

² State Key Laboratory Cultivation Base of Geographical Environment Evolution, Nanjing 210023, China

³ Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China

⁴ Physiological Signal Analysis Team, Center for Machine Vision and Signal Analysis, University of Oulu, 90014 Oulu, Finland; Anja.Keskinarkaus@oulu.fi (A.K.); Tapio.Seppanen@oulu.fi (T.S.)

* Correspondence: 09359@njnu.edu.cn; Tel.: +86-136-1157-8959

Abstract: This paper proposes a joint encryption and screen-cam robust watermarking scheme. This method combines the advantages of smartphone, encryption and watermarking technologies, thereby achieving watermark extraction with a smartphone, partial decryption and tracking leakage from sneak shots. We design a dual watermarking algorithm to achieve watermark detection from both encrypted and decrypted images. First, a watermark is embedded in the discrete Fourier transform (DFT) domain to enable leakage tracking. Then, a second watermark is generated based on QR (Quick response) code encoding and inverse DFT to achieve high watermark capacity and error correction ability, where the secret key for decryption is included in the watermark message. By hiding this message carrying the watermark for the encrypted image in the changes caused by embedding the first watermark, we can improve imperceptibility and will not affect the effectiveness of the proposed scheme. Finally, to enhance the robustness of watermark after encryption, a chaotic mapping-based segment encryption algorithm is proposed. In the process of watermark detection, to cope with perspective correction, a frame locating based algorithm is employed to achieve watermark synchronization from a recaptured picture of the encrypted image. Considering the severe quality degradation, we use a noise component and local statistic feature-based method to extract the message bits. The experimental results show that the proposed scheme is secure, and highly robust, to screen-cam the process for both before and after decryption. Additionally, after decryption, the proposed scheme also has high robustness against common image processing attacks.



Citation: Chen, W.; Ren, N.; Zhu, C.; Keskinarkaus, A.; Seppänen, T.; Zhou, Q. Joint Image Encryption and Screen-Cam Robust Two Watermarking Scheme. *Sensors* **2021**, *21*, 701. <https://doi.org/10.3390/s21030701>

Received: 24 December 2020

Accepted: 18 January 2021

Published: 20 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: image encryption; chaotic mapping; screen-cam process; robust watermarking; discrete fourier transform; smartphone

1. Introduction

With the continuous improvement of smartphone hardware and mobile applications, the functions of smartphones have become quite powerful. Nowadays, smartphones have become indispensable in our daily life. At the same time, information leakage by taking photos with a smartphone has become more common. To protect image data from being leaked, there are two typical solutions. One solution is to encrypt image data and decrypt it with a secret key when using it [1–4]. The other solution is access control technology [5–8], which prevents unauthorized access to the data. Although these methods can keep the image unreadable or inaccessible, they cannot prevent users from leaking the decrypted image displayed on the screen by taking photos with smartphones.

In order to protect image and tracking leakage, a joint encryption and watermarking scheme is an effective solution. Furthermore, smartphones are a double-edged sword in

data protection. In addition to stealing data through shooting with smartphones, smartphones also have unique advantages in user identity authentication. Therefore, how to combine the advantages of smartphone, encryption and watermarking technologies for identity authentication, key management and leakage tracking is a meaningful issue.

Joint encryption and screen-cam robust watermarking has two typical application scenarios. (1) Smartphone-based message reading and partial decryption. This scenario is like reading a QR code with a mobile phone. As shown in Figure 1, we can read a secret key, access level, recipient ID and other information from the encrypted image through scanning or shooting with a smartphone and perform identity authentication based on the mobile security application. Unauthorized users will not obtain the decryption key, and authorized users will be returned with a decryption sequence that indicates the secret key and the user's access level. Corresponding partial decryption according to the user's access level can be performed after entering the decryption sequence in the PC software. (2) Leakage tracking. In an access control environment, it is difficult for unauthorized users to steal the data. However, authorized users can take a photo of a decrypted image to cause leakage. Once the data has been leaked through the photo, we can extract the watermark information from the photo. After that, we can locate the receiver of this data, so as to achieve accountability.



Figure 1. Application scenario of smartphone-based watermark reading and partial decryption.

The existing researches of joint encryption and watermarking schemes mainly focus on two categories: commutative encryption and watermarking (CEW) [9–15] and reversible data hiding in encrypted images (RDH-EI) [16–28]. CEW achieves mutual independence of encryption and watermarking. RDH-EI aims to achieve lossless recovery of the original image, which is mainly designed for situations in which permanent distortion is strictly forbidden. However, due to the different purposes of the algorithm design, these schemes are designed to be robust to common image processing attacks or fragile watermarking, which means they are not applicable for the screen-cam process.

Screen-cam processing is using a camera device to regenerate the content displayed on the screen into digital signals. Hence, the screen-cam process can be considered as a cross-media signal transmission process containing digital-to-analog and analog-to-digital conversion. Similar cross-media signal transmission includes the print-scan process and print-cam process. Existing research on print-scan or print-cam robust image watermarking can be divided into three categories: watermark pattern-based methods [29–35], Fourier domain-based methods [36–39] and multidomain-based methods [40–42]. Although the ideas of these methods are valuable for studying screen-cam robust watermarking algorithms, these methods are not applicable for the screen-cam process [43]. The screen-cam process has its particularity, which causes various types of distortions [43–45], including linear distortion, gamma tweaking, geometric distortion, moiré noise and low-pass filter attack. To cope with these severe distortions, Fang et al. [43] proposed a feature-based watermarking scheme where the message is embedded in the discrete cosine transform

(DCT) domain of local feature regions. To further improve the robustness, Fang et al. [46] proposed a deep learning-based watermarking scheme. To achieve blind detection under geometric distortion, Chen et al. [44] designed a watermark synchronization method and embedded the message in the discrete Fourier transform (DFT) domain. These methods are effective for screen-cam attack. However, to be able to detect a watermark from encrypted and watermarked images, we need to study new watermarking schemes and investigate matching encryption algorithms.

Chaos-based image encryption algorithms have been extensively researched [47–53] because of the advantages of chaotic system, which include high sensitivity to initial conditions and control parameters and pseudorandom behaviors [3,49]. Typical chaotic map-based encryption has two stages: permutation and diffusion [54]. Permutation operation changes the pixel positions commonly based on a generated chaotic order. Diffusion operation encrypts pixel values based on a generated chaotic sequence or matrix. For example, XOR operation is a widely used diffusion method [55–57]. However, these methods are not robust to cropping attack, which means they cannot achieve partial decryption. Especially for high-resolution satellite images and secret raster maps, when facing users with different access levels, performing corresponding partial decryption is a practical and meaningful function.

As the existing joint encryption and watermarking schemes do not consider the screen-cam process, to solve this issue, a joint encryption and screen-cam robust two watermarking scheme is proposed. Furthermore, a joint encryption and watermarking scheme should not be a simple superposition of two technologies. When utilizing both technologies, they should complement each other. Similarly, how to combine the two watermarking is also important. Therefore, balancing imperceptibility and robustness while employing two watermarking, improving watermark capacity, and achieving mutual cooperation of encryption and watermarking technologies are our research objectives. The main contributions are as follows:

- We propose a dual watermarking algorithm to achieve watermark detection from both encrypted and decrypted images. Additionally, to improve imperceptibility and guarantee effectiveness of the proposed scheme, we hide the watermark for the encrypted image into the changes caused by embedding the watermark for the decrypted image.
- We design a QR (Quick Response) code encoding and inverse discrete Fourier transform (IDFT) based watermark generation method, which can improve watermark capacity and error correction ability.
- We propose a chaotic mapping-based segment encryption algorithm to cooperate with the watermarking algorithm. By applying this, the watermark can be enhanced after encryption, thereby achieving watermark extraction from the encrypted image with smartphones.

In the rest of the paper, the proposed method is introduced in Section 2. Section 3 analyzes the selection of parameters and experiment results. Section 4 gives the discussions, and Section 5 draws the conclusions.

2. Proposed Method

2.1. Embedding and Encryption Scheme

In order to achieve screen-cam robust watermarking both before and after encryption, we propose a dual watermark method and a chaos-based encryption method. With regard to watermarking algorithm, we embed watermark A and watermark B in the host images, where watermark A works in the decrypted image and watermark B works in the encrypted image. In other words, watermark A is designed for leakage tracking that can be detected from a recaptured image of a decrypted host image, and watermark B is designed for real-time information reading from a recaptured image of an encrypted host image. The key to a watermarking algorithm is to design a high capacity and error correction watermark B generation method and ensure imperceptibility by designing embedding

methods. Furthermore, to achieve secure key management, the key for decryption is included in the watermark B message. Therefore, we do not need additional transmission of the secret key as separate data. With respect to the embedding region, considering the advantages of DFT domain in the screen-cam process [44], we employ DFT-based methods to embed both watermark messages in blocks repeatedly. With regard to the encryption algorithm, it is not only to achieve encryption but also to work with the watermark B. The main idea of encryption is using an odd-even segment encryption method to work with the odd-even quantization based watermarking method to achieve the purpose of enhancing the robustness of the watermark after encryption.

Figure 2 illustrates the embedding and encryption process of one block. If it is a multi-band image, we perform embedding and encryption on each band. The process can be divided into three parts. (1) Embed message A in the DFT domain. (2) Generate the watermark matrix of watermark B based on the QR code encoding method and inverse Fourier transform. Then, embed watermark B based on odd-even quantization and the difference caused by embedding message A. (3) Generate chaos mapping sequence for odd and even separately, and then perform encryption. Details are as follows:

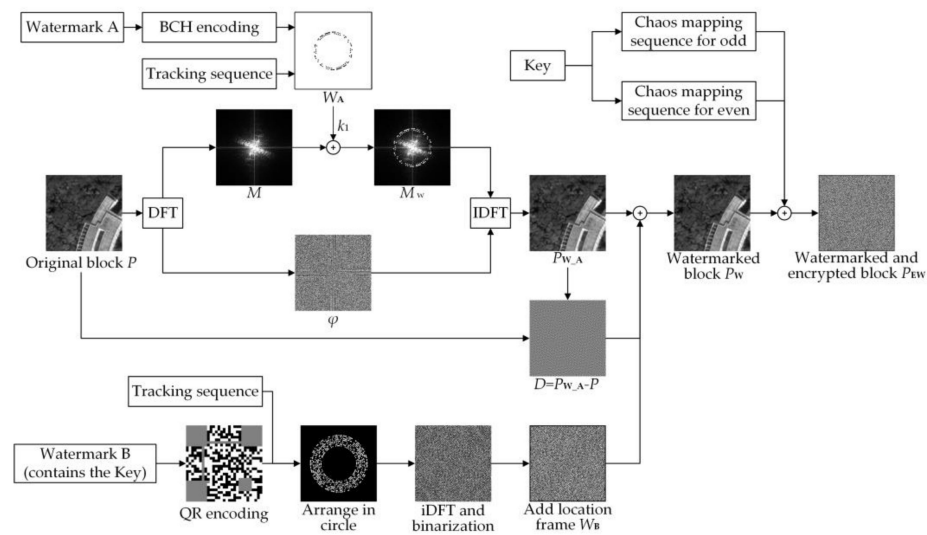


Figure 2. Embedding and encryption process of one block.

2.1.1. DFT-Based Embedding of Watermark A

Considering the possible rotation, scaling and transform (RST) attacks caused by user operations, we embed the message sequence and tracking sequence in a circle region with different radii, as shown in Figure 1 so that we can locate the tracking message through an exhaustive search after a log-polar transform of DFT coefficients, thereby resynchronizing the watermark message. Details as follows:

- Step 1: We encode watermark message A by a BCH error correction code to achieve the message sequence $M_A = \{m_A(i) | m_A(i) \in \{0, 1\}, i = 0, \dots, l - 1\}$ and generate a 30-bit pseudorandom sequence as the tracking sequence $M_{TA} = \{m_{TA}(i) | m_{TA}(i) \in \{0, 1\}, i = 0, \dots, 29\}$.
- Step 2: Divide host image into square blocks. According to the original size of host image, set the side length of the square block to L_0 . If the edge part is not enough to form blocks, it is supplemented with pixels of 0 value.
- Step 3: M_A and M_{TA} are embedded at R_1 and R_2 separately. The embedding coordinates of M_A is defined as:

$$\begin{aligned} x_i &= \text{floor}\left(\frac{L_0}{2} + 1\right) + \text{floor}\left[R_1 \cdot \cos\left(\frac{i \cdot \pi}{l}\right)\right] \\ y_i &= \text{floor}\left(\frac{L_0}{2} + 1\right) + \text{floor}\left[R_1 \cdot \sin\left(\frac{i \cdot \pi}{l}\right)\right] \end{aligned} \quad (1)$$

where i is the i -th element of M_A . The method of calculating embedding coordinates of M_{TA} is the same. After this, we can achieve the watermark matrix $W_A(x_i, y_i)$.

Step 4: Each time, input one band of one original square block P , and perform DFT transform. The watermark is embedded in the magnitude spectrum. Because the low and medium-frequency magnitude coefficients with high values can be well preserved in screen-cam process and the low values are not [44], the embedding method is defined as:

$$M_W(x, y) = \begin{cases} k_1, & w(i) = 1 \\ \text{no change}, & w(i) = 0 \end{cases} \quad (2)$$

where $M_W(x, y)$ defines the watermarked magnitude spectrum and k_1 defines the embedding strength.

Step 5: Perform inverse DFT to achieve one watermarked band of one block P_{W_A} . Output P_{W_A} each time.

Step 6: Repeat step 4 and step 5 to complete the embedding of all bands and blocks. Then, delete the part for supplement. The result is the watermarked image with watermark A.

2.1.2. Odd-Even Quantization-Based Embedding of Watermark B

The QR code is an error correction code and has high information capacity. Therefore, it is commonly used as watermark generation method [58–60]. Hence, we propose a novel QR code-based watermark generation method. Details are as follows:

First, we encode watermark message sequence B by the QR code encoding method. The structure of the QR code includes the fixed pattern for resynchronization and encoding region for the message, as shown in Figure 3. Because we will rearrange the message bits, we do not need the fixed pattern. We choose the encoding region and record the encoding message line by line as $M_B = \{m_B(i) | m_B(i) \in \{0, 1\}, i = 0, \dots, j - 1\}$. Considering the watermark capacity, the watermark message B is designed to be not more than 42 bytes of 8 bits [61]. The message is encoded by version 3 QR code with M error correction level, and it can be recorded as a sequence of $j = 597$ bits.

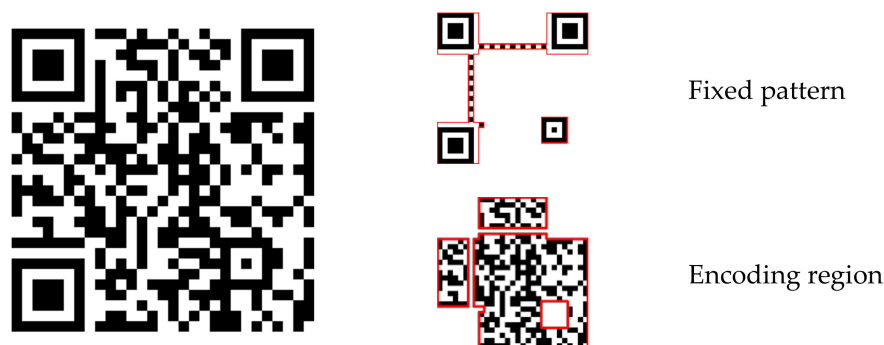


Figure 3. QR code architecture.

Then, as above, a 23-bit pseudorandom sequence $M_{TB} = \{m_{TB}(i) | m_{TB}(i) \in \{0, 1\}, i = 0, \dots, 22\}$ is generated as the tracking sequence. Therefore, the whole watermark message is the combination of M_B and M_{TB} , a total of 620 bits.

Next, we rearrange the whole watermark message sequence in circle regions of a $L_0 \times L_0$ zero matrix, as shown in Figure 4a. The message bits are arranged at radii $R_B(i) \in \{60, 65, 70, 75, 80, 85, 90, 95\}$. At each $R_B(i)$, $R_B(i)$ bits are arranged. For example, at radius 60, 60 bits are arranged. The coordinates are calculated based on Format (2).

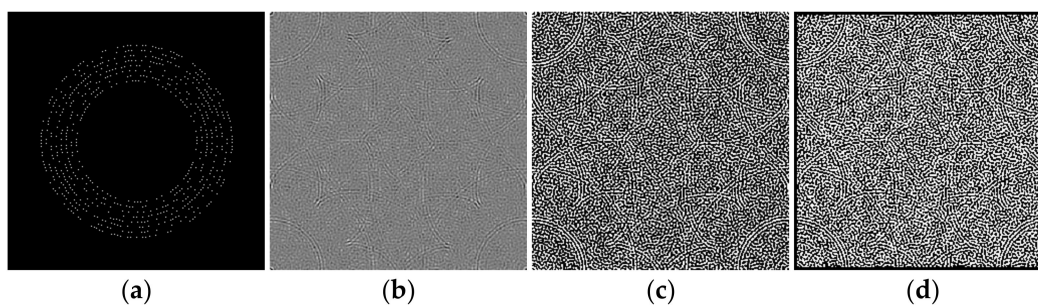


Figure 4. Generation process of W_B . (a) Arrange message sequence in circle regions of a zero matrix; (b) inverse Discrete Fourier transform (IDFT) of matrix (a); (c) binarization of matrix (b); (d) add a frame for matrix (c).

After that, we perform IDFT on the matrix, as shown in Figure 4b, and binarization based on positive and negative values, as shown in Figure 4c.

Finally, we add a frame to the matrix by changing all the values within 3 from the edge to 0, as shown in Figure 4d. The result is the watermark matrix $W_B(x_i, y_i)$.

For ensuring the imperceptibility of the proposed scheme, W_B is hidden in the image changes caused by embedding watermark A. The proposed odd-even quantization-based embedding method only causes around 50% of the pixel values to change by 1. Therefore, it does not affect the use of watermark A.

The W_B is also embedded block by block. The embedding process of one band of one block is as follows: First, we calculate the image changes $D = P_{W_A} - P$ then embed W_B bit by bit. Figure 5 illustrates the embedding procedure of one bit. The main idea is modulating the pixel values in a reverse direction of the changes caused by embedding watermark A. Finally, we achieve the watermarked band of one block P_W with message A and message B.

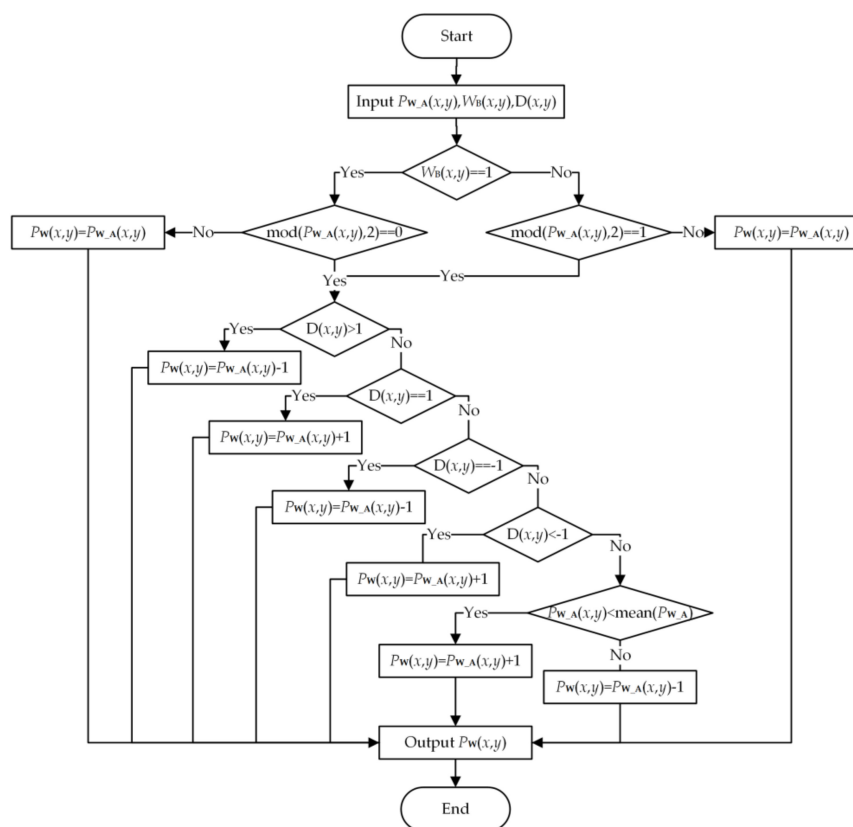


Figure 5. Embedding procedure of one bit of W_B .

2.1.3. Odd-Even Segment Encryption

We encrypt odd numbers and even numbers separately into different numerical ranges based on a logistic map, which is idely used to generate a chaotic mapping sequence [1,62,63]. The logistic map is defined as:

$$X_{n+1} = u \cdot X_n \cdot (1 - X_n) \tag{3}$$

where u is the system parameter. When $X_n \in (0, 1)$, $u \in (3.5699456, 4)$, the logistic map is chaotic.

The encryption process is as follows:

First, given the secret key, which is the combination of two initial values $X_O(0)$, $X_E(0)$ and a parameter u two one-dimensional array $V_1(i)$ and $V_2(i)$ with a length of L_0 are generated by iterating $2 \cdot L_0$ times through Equation (3), respectively. L_0 depends on the data type of the image, where $L_0 = 2^{bit\ depth-1}$.

Then, sort $V_1(i)$ from the smallest to largest to obtain array $V_{s1}(i)$, and record the index that elements of $V_{s1}(i)$ in $V_1(i)$ as $V_O(i)$. For example, suppose element $V_1(9)$ becomes element $V_{s1}(0)$ after sorting, then $V_O(0) = 9$. Perform the same process on array $V_2(i)$ to obtain $V_E(i)$. $V_O(i)$ and $V_E(i)$ with a length of L_0 are the two chaos mapping sequences for odd and even values separately.

Finally, the encryption method is defined as:

$$\begin{aligned} P_{EW}(x, y) &= V_E(P_W(x, y)/2) && \text{if } P_W(x, y) \text{ is even} \\ P_{EW}(x, y) &= V_O((P_W(x, y) - 1)/2) + L_0 && \text{if } P_W(x, y) \text{ is odd} \end{aligned} \tag{4}$$

where P_{EW} defines the encrypted and watermarked image. An example of pixel encryption is shown in Figure 6, where all original even values are encrypted to low values and original odd values are encrypted to high values.

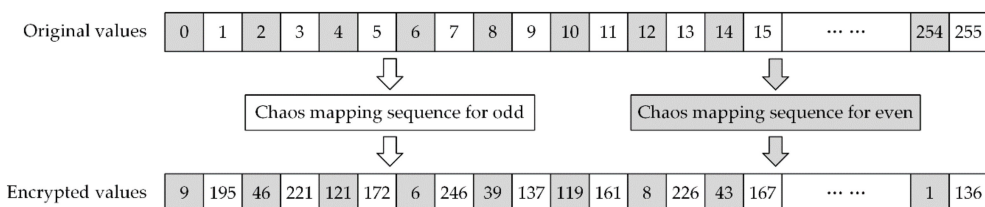
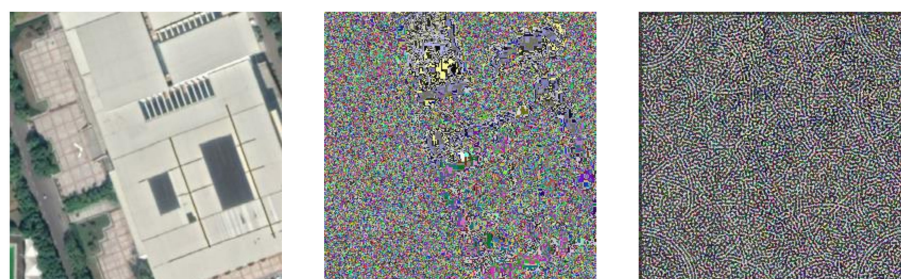


Figure 6. Example of pixel encryption.

The weakness of directly modifying the pixel values based on a mapping sequence for encryption is that the shape of the area with the same pixel values can still be seen after the encryption, as shown in Figure 7a,b. Fortunately, after watermark embedding, most of the same pixel values in one area will become different, which means it will effectively avoid the weakness, as shown in Figure 7c.



(a) Original image (b) Encrypted image without watermark (c) Encrypted image with watermark A and B

Figure 7. Example of Encryption Result.

2.2. Extraction and Decryption Scheme

Figure 8 shows the extraction and decryption process. Nowadays, authentication and secret key management through smartphones are already mature technologies. When receiving the encrypted and watermarked image, authorized users can use smartphones to detect and extract watermark B by scanning or photographing with a proprietary application. Then, the secret key and other information are obtained by decoding watermark B to decrypt the image.

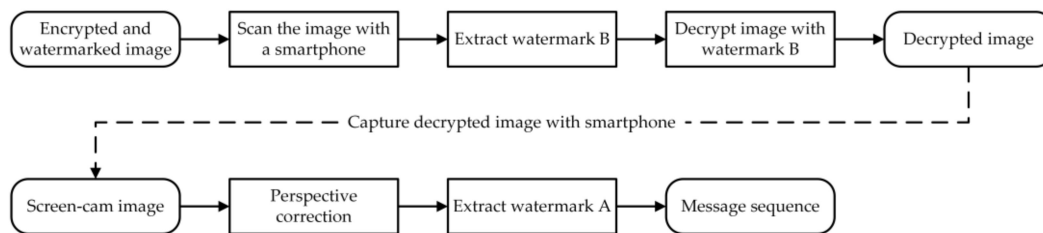


Figure 8. Framework of extraction and decryption process.

If the decrypted image is photographed without authorization, watermark A can be extracted from the screen-cam image to hold data leakage accountability. Because watermark A is designed for leakage tracking, manual operation is acceptable. For screen-cam images, we perform perspective correction of the recaptured image and crop out the needed part for watermark A extraction. This part is divided into blocks, and watermark A is detected block by block. Next, we locate the tracking sequence by calculating the cross-correlation to estimate the positions of the embedded bits. Finally, message A is extracted and decoded.

2.2.1. Watermark B Extraction and Decryption

Using smartphones to extract watermark B from the encrypted image is like using smartphones to read QR code. Because today's smartphones have high-megapixel cameras, the captured image will be highly zoomed-in compared with the original image displayed on the screen when shooting at a close distance. Therefore, for a screen-cam image scanned, as shown in Figure 9a, we zoom and crop out the needed part first. According to the camera resolution of the smartphone, we crop and zoom out the captured image accordingly to obtain image I_b , as shown in Figure 9b. The perspective correction, message extraction and decryption process are as follows:

1. Perspective correction:

- Step 1: Input I_b . Convert I_b to grayscale I_g and calculate $I'_g = 255 - I_g$. Then, perform Gaussian filtering with a two-dimensional Gaussian kernel H_1 , where sigma is set to 1 and window size is set to 6. Hence, the $I_c(x, y) = H_1 * I'_g(x, y)$ is obtained by a convolution process, as shown in Figure 9c.
- Step 2: Binarize I_c based on a threshold T_1 to obtain binary image I_d , as shown in Figure 9d. Then, perform opening operation, which is erosion and dilation in turn, with structuring element se to obtain I_e , as shown in Figure 9e.
- Step 3: Perform Hough transform to search the lines from I_e , and calculate the intersection points of these lines within the image range, as shown in Figure 9f. Record the coordinates of these points as p_i .
- Step 4: Perspective transformation needs four pairs of points [43]. The side length L_0 is known, which means we know transformed coordinates of the four corners of one block. Therefore, we can select four corner points of one block for perspective correction. Select and construct p_i into a point set $S = \{s(i) | s(i) \in \{p_1, p_2, p_3, p_4\}, i = 0, \dots, n\}$, which contains all candidate point sets that can be used for perspective correction, based on the searching method in [44]. The $s(i)$ are sorted according to the sum of the distances

between the points from largest to smallest. An example of the quadrilaterals formed by each $s(i)$ is shown in Figure 9g. The $s(1)$ is selected for message extraction.

Step 5: The perspective correction process is defined as:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H_2 \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \text{ where } H_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (5)$$

where $[x', y', 1]^T$ and $[x, y, 1]^T$ define the homogeneous point coordinates of the corrected image and the captured image, respectively. H_2 is a non-singular 3×3 homogeneous matrix. By using $s(1) \in \{p_1, p_2, p_3, p_4\}$, and setting the four corresponding points representing the four corners of a corrected block as $\{p'_1, p'_2, p'_3, p'_4\}$, the H_2 can be calculated. Then, we perform perspective correction of I_b and crop out the block I_h , as shown in Figure 9h. Output I_h .

2. Message extraction:

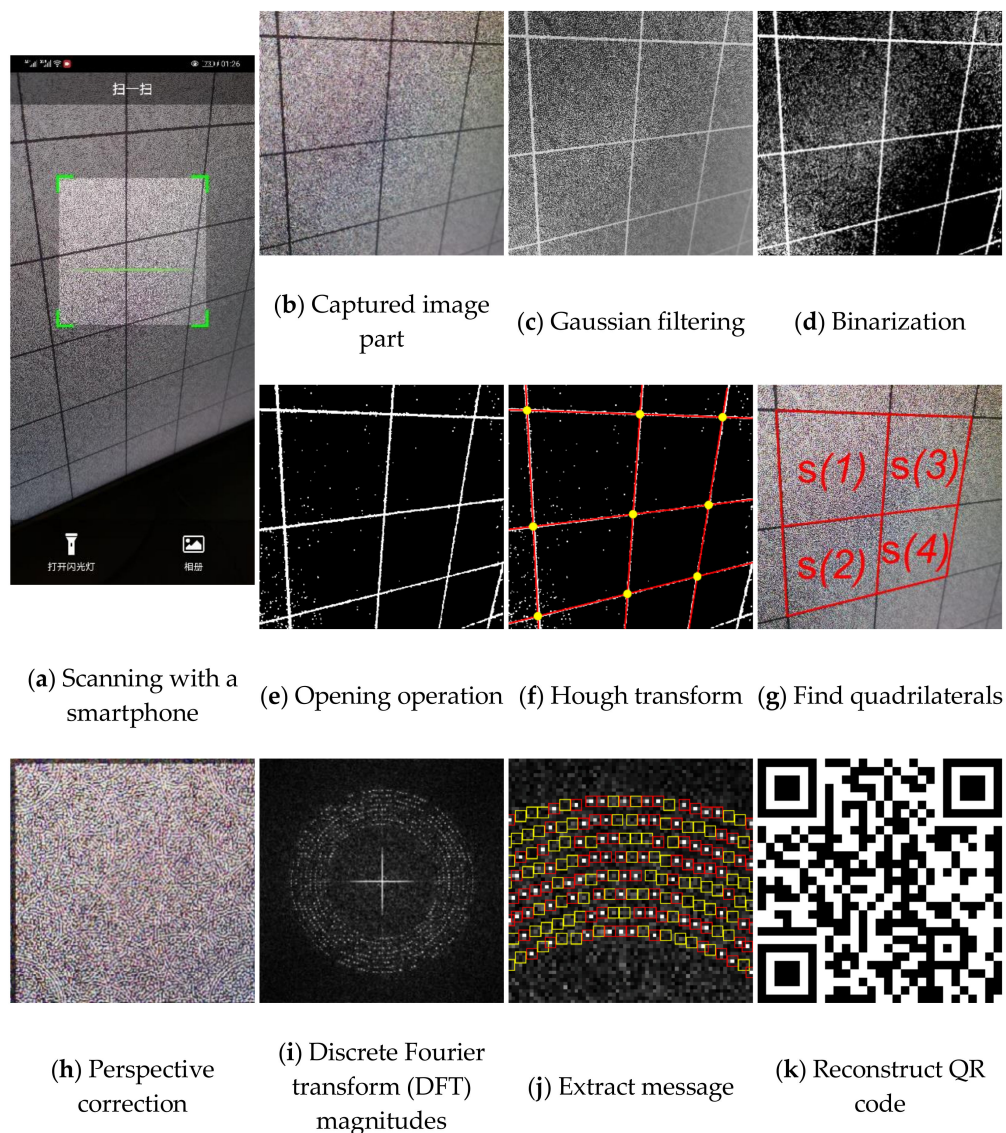


Figure 9. Detection and extraction process of watermark B.

We use the grayscale of I_h and perform DFT to obtain the magnitude spectrum I_i , as shown in Figure 9i. The encrypted image is a noise image, which means the image itself does not have high magnitude values around the embedding region. In other words, the modulated high magnitudes for message embedding are significant.

Furthermore, the manually perspective correction cannot be perfect, which means it will cause the shifting of magnitude coefficients [44]. Therefore, we use the maximum value $v_m(i)$ within a 3×3 region centered at the embedding coordinates to determine the message bit $w'_B(i)$, as shown in Figure 9j, where red boxes and yellow boxes are the 3×3 areas of the positions where the embedded message bit is '1' and '0', respectively. The extraction method of watermark B is defined as:

$$w'_B(i) = \begin{cases} 1, & \text{if } v_m(i) > T_B \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$T_B = E_B + k_2\sigma_B \quad (7)$$

where T_B is the set threshold, E_B and σ_B are the mean value and the standard deviation of all the magnitudes in the range of [60, 95], k_2 is a fixed value.

Although I_h is corrected to the original size, because it is square we still need to consider whether the image is under a rotation by 90° . Based on the nature of Fourier domain, we can easily calculate the coordinates of the embedded tracking sequence M_{TB} in these two cases. Therefore, based on Equations (6) and (7), we extract the messages $M'_{TB}(1)$ and $M'_{TB}(2)$ from the positions of embedded tracking sequence in both cases. If the erroneous bits in either of $M'_{TB}(1)$ and $M'_{TB}(2)$ are less than the given threshold T_2 , we consider the watermark exists.

M'_B is then extracted also based on Equations (6) and (7). Based on the inverse process of the watermark generation method, the QR code is reconstructed with M'_B , as shown in Figure 9k. Finally, by decoding the QR code, the watermark message B containing the decryption key is obtained.

3. Decryption:

Based on the extracted decryption key and the bit depth of the image, the same $V_O(i)$ and $V_E(i)$ can be calculated. The decryption process is defined as:

$$P'_W(x, y) = \begin{cases} f_{index}(V_E, P_{EW}(x, y)) \cdot 2 & , \text{if } P_{EW}(x, y) < 2^{bit\ depth-1} \\ f_{index}(V_O, P_{EW}(x, y)) \cdot 2 + 1 & , \text{else} \end{cases} \quad (8)$$

where P'_W defines the decrypted and watermarked image and function $f_{index}(x, y)$ defines returning the index of element $x(i)$ that equals to y . *bit depth* defines the bit depth refers to the image format.

2.2.2. Watermark A Extraction

For a screen-cam image, we perform perspective correction by manually selecting four points. As watermark A is designed for leakage tracking, manual selection is acceptable. As shown in Figure 10, we can use the four corner points of the host image $\{p_1, p_2, p_3, p_4\}$ or the four corner points $\{p_5, p_6, p_7, p_8\}$ of the screen to correct the captured image to the original size. Then, the portion needed for watermark detection is cropped. If the original size of the image and the screen are unknown, because the watermark A is robust to scaling attack, we can also correct the captured image to an image with the original aspect ratio. As we mentioned in Section 2.2.1, a slight accuracy error in corner point selection and resulting shift of magnitude coefficients is acceptable, because we perform watermark extraction based on the maximum value within 3×3 region of the embedded watermark position.

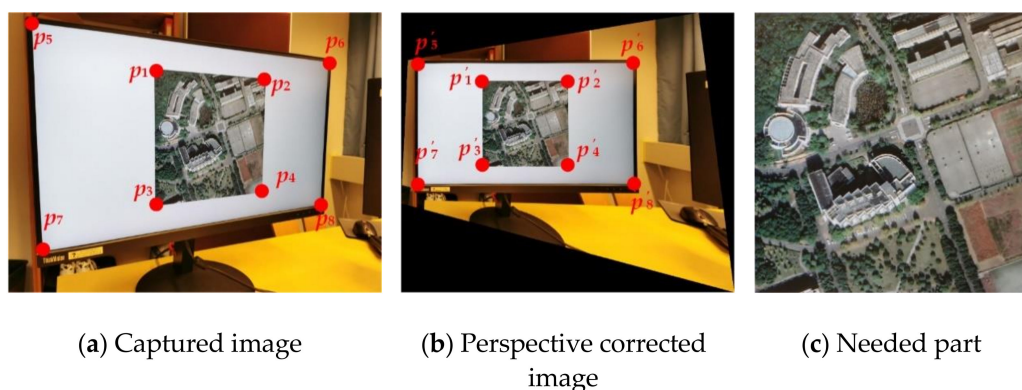


Figure 10. Perspective correction process.

According to the nature of the DFT domain, the message embedded in the magnitude spectrum is distributed in the whole image. The message embedded in each block is the same. Therefore, any part of the image can be used for watermark detection and extraction. Considering the severe distortion caused by a screen-cam attack, we can use a square block $B(i)$ with a side length of L_1 , which is larger than L_0 , for detection. Furthermore, if there is no watermark, the DFT magnitude coefficients of the blocks with a small amount of overlap are very different, which will not cause a false alarm. Therefore, the blocks used for detection do not need to be completely nonoverlapping. As shown in Figure 11a, the overlapping block $B(1)$, $B(2)$, and $B(3)$ can all be used for watermark detection at the same time. Therefore, we choose the blocks in turn with a step of $0.7 \cdot L_1$ at both horizontal and vertical directions.

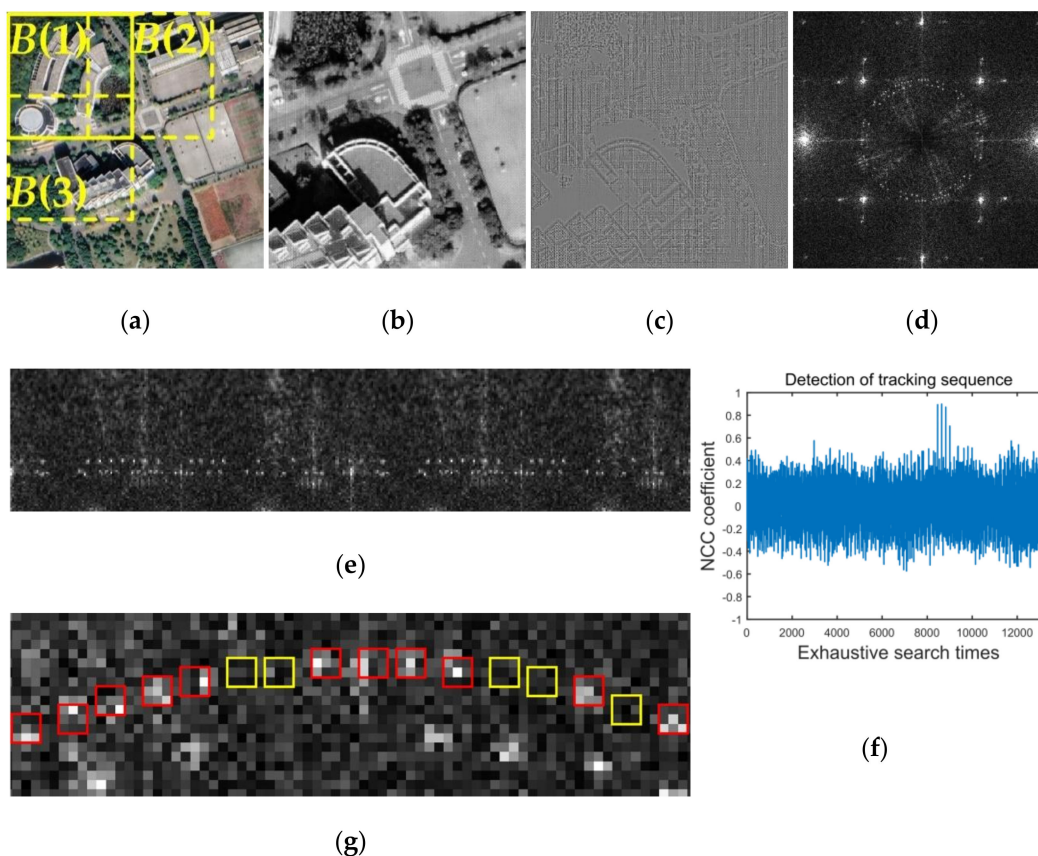


Figure 11. Detection and extraction process of watermark A. (a) Examples of blocks can be used for watermark detection; (b) luminance spectrum of one block; (c) noise component of one block; (d) DFT domain of noise component; (e) map detection range from Cartesian coordinates to polar coordinates; (f) calculate NCC coefficients; (g) 3×3 region centered at embedding coordinates.

Because the size of a selected block $B(i)$ is larger than an embedded block, the positions of embedded messages are changed accordingly. Besides, watermarks can be considered as a form of noise [64]. Detecting the watermark from the noise component can reduce the negative impact of the image itself [44]. Therefore, to resynchronize the watermark, we locate the embedded tracking sequence from the noise component using the normalized cross-correlation (NCC) function. The noise component B_n is defined as:

$$B_n = B_l - H_3 * B_l \quad (9)$$

where B_l defines the luminance spectrum of selected block B and H_3 defines a 3×3 spatial domain Wiener filter. Figure 11b,c show examples of a $B_l(i)$ and $B_n(i)$.

We transform B_n to the DFT domain B_f , as shown in Figure 11d. Considering the size and scaling difference between $B(i)$ and the original block, the detection range is set from radius 50 to radius 150. Mapping the detection range from Cartesian coordinates to polar coordinates is done, as shown in Figure 11e. Then, we perform an exhaustive search by calculating the NCC coefficients between the extracted coefficients and the tracking sequence M_{TA} , which is defined as:

$$C(j) = \frac{\sum_{i=0}^{29} (V'_{TA,j}(i) - \overline{V'_{TA,j}}) (M_{TA}(i) - \overline{M_{TA}})}{\sqrt{\sum_{i=0}^{29} (V'_{TA,j}(i) - \overline{V'_{TA,j}})^2 \sum_{i=0}^{29} (M_{TA}(i) - \overline{M_{TA}})^2}} \quad (10)$$

where $C(j)$ defines the NCC coefficient of j -th search and $V'_{TA,j}$ defines the extracted message sequence of the j -th search. $\overline{V'_{TA,j}}$ and $\overline{M_{TA}}$ defines the mean of extracted message sequence and the original tracking sequence. $V'_{TA,j}(i)$ is the maximum coefficient value within the 3×3 region centered at the detection position. Because of this, if the watermark exists, more than one high NCC coefficient may be calculated. An example of calculation resulting from Figure 11e is shown in Figure 11f. If the maximum value of $C(j)$ is greater than 0.65, which is an experimental threshold, we consider the positions of corresponding $V'_{TA,j}$ is the positions of embedded tracking sequence.

Based on the detected tracking sequence, we can estimate the positions and the radius R'_1 of embedded M_A in B_f . Because the polar mapping process interpolates the data, which causes a slight change, we extract the watermark message from B_f directly. The extraction method of watermark A is the same as watermark B, the maximum value $v_m(i)$ within the 3×3 region centered at the embedding coordinates is used to determine the message bit $w'_A(i)$, as shown in Figure 11g, but with different parameters. The extraction method of watermark A is defined as:

$$w'_A(i) = \begin{cases} 1, & \text{if } v_m(i) > T_A \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

$$T_A = \overline{E}_A + k_3 \sigma_A \quad (12)$$

where T_A is the set threshold, \overline{E}_A and σ_A are the mean value and the standard deviation of all the magnitudes in the range of $[R'_1 - 2, R'_1 + 2]$ and k_3 is a fixed value. Finally, an extracted watermark message $W'_A(i)$ is obtained by BCH decoding. To avoid a false positive, watermark detection is successful only when two of $W'_A(i)$ are the same. The same decoded message will be used as extraction result.

3. Experimental Results

In our experiment, we set message A to 24 bits, which means it can support 16,777,216 IDs. The ID sequence was encoded by BCH (63.24) to generate M_A with 63 bits, which can correct 7 error bits. Watermark B was set as {key = 8190/1713/398232; level = NNU; ID = 15821018}, including the decryption key, user ID, and other information.

To ensure the size of a block for resynchronizing from an encrypted image is applicable for practical application, L_0 was set to 256. The middle frequency coefficients at $R_1 = 60$ and $R_2 = 55$ were selected to embed M_A and M_{TA} , respectively. The threshold T_2 was set to 4. Because M_{TB} is 23 bits, the false positive rate for judging whether watermark B exists can be calculated as $\sum_{23-T_2+1}^{23} (0.5)^{23} \cdot \left(\frac{23!}{3!20!}\right) = 2.44\text{E-}04$ [65]. This false positive rate is not very low. Fortunately, if the reconstructed QR code based on extracted watermark B is wrong, it cannot be decoded. This can also be regarded as double insurance to prevent false positives.

The monitor we used was a 27-inch ‘ThinkVision P27q’ monitor with 2560×1440 pixels. The photography equipment we used was a P30PRO smartphone with a 40 MP pixel camera. The application for extracting the watermark from encrypted images was developed by Java running on the platform of P30PRO. The rest of the experiments were performed by Matlab 2019b on a Windows 10 operation system with an Intel i7-9700 CPU. The host data was five images from database [66] and five images restitched by tile images obtained from Google Earth.

In Section 3.1, the selection of parameters through statistical experiments is presented. In Section 3.2, the security of encryption scheme is discussed. In Section 3.3, the robustness of the watermark B against screen-cam attack is analyzed. In Sections 3.4 and 3.5, we verify the robustness of watermark A against common image processing attacks and screen-cam attacks, respectively. As our method can achieve partial decryption, in Section 3.6, we verify the robustness of the partially decrypted image against screen-cam attack.

3.1. Parameter Settings

3.1.1. Selection of Embedding Strength k_1

Embedding strength balances the robustness and imperceptibility of the proposed scheme. One thousand tile images obtained from Google Earth were utilized for statistical experiments to select the appropriate embedding strength k_1 in Equation (1). Image quality degradation was evaluated by the widely used peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [67]. The average PSNR and SSIM values of the embedded images with different k_1 are shown in Figure 12a,b. In order to ensure the PSNR values of most images after embedding was greater than 40, we set k_1 to 85. The average PSNR was 40.9446 dB and the average SSIM was 0.9868. With the selected k_1 , the PSNR and SSIM values of all the test images after embedding are shown in Figure 12c,d. The examples of original the host image, encrypted and watermarked host images and decrypted and watermarked host images are shown in Figure 13.

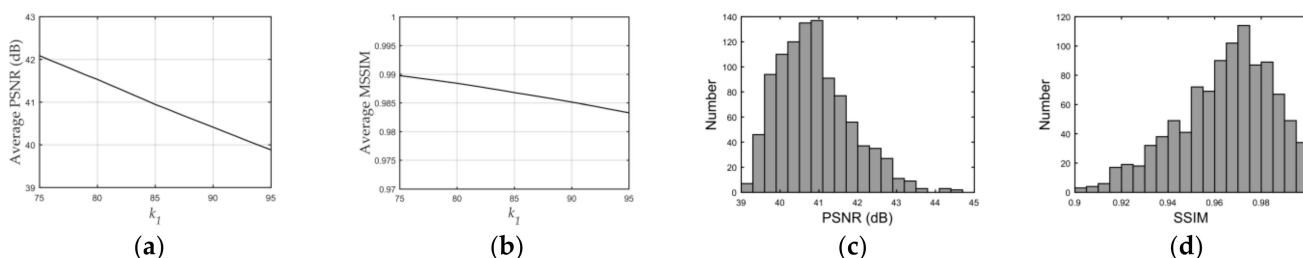


Figure 12. PSNR and SSIM with different and selected k_1 . (a) and (b) are average PSNR and SSIM values with different k_1 ; (c) and (d) are PSNR and SSIM values with selected k_1 .

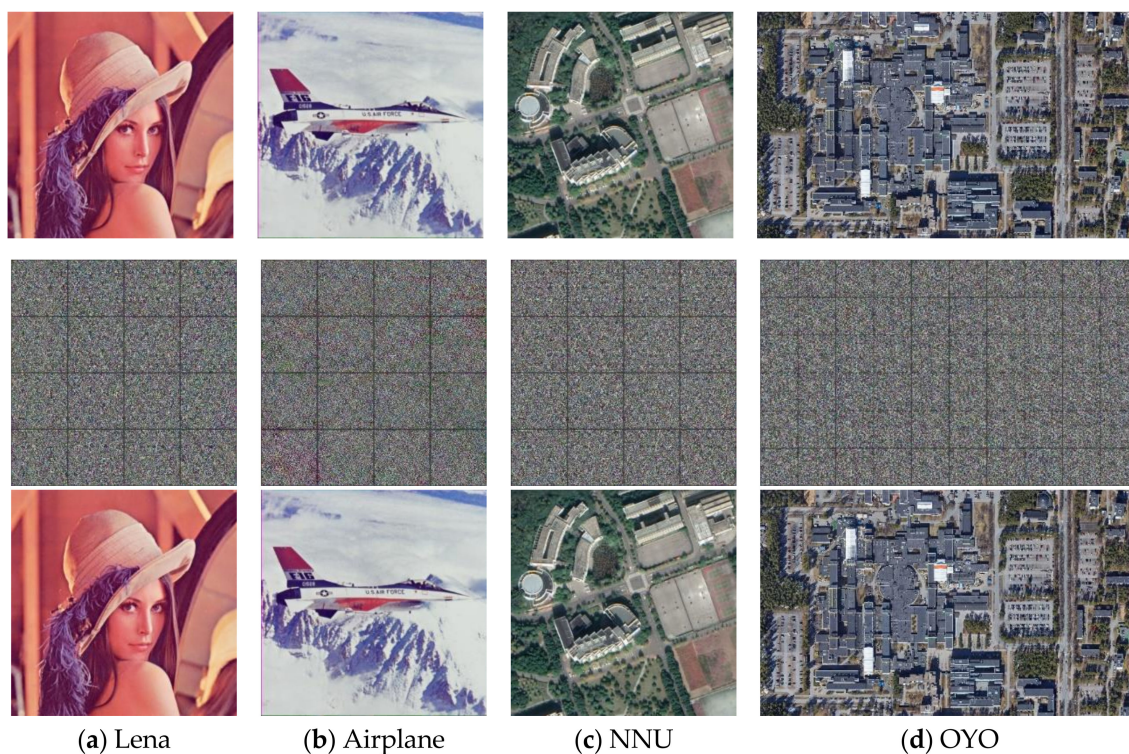


Figure 13. Examples of host images (**first row**), encrypted and watermarked images (**second row**), and decrypted and watermarked images (**third row**).

3.1.2. Selection of Threshold T_1 and Structuring Element se for Synchronization

To ensure the success of the automatic perspective correction of recaptured encrypted images, we need to select the most suitable threshold T_1 for binarizing and structuring element se . According to the shooting distance, the camera resolution and the screen resolution, the scaling ratio of the captured images is quite different. To process the recaptured images of different scaling levels, the required parameters vary greatly. Therefore, we tested the performance of different parameters with different shooting distances.

When we use a smartphone to scan the code on the screen, the distance between the smartphone and the screen is commonly within 40 cm. Therefore, we counted the results of automatic perspective correction with different T_1 and se at the shooting distance of 10 cm, 20 cm, 30 cm and 40 cm, and the shooting angle of 0 degree perpendicular to the screen. Because our photography equipment had a high resolution that causes the captured image to be zoomed, we set the captured image to zoom 60% before the processing. The structuring element se we used here consisted of only '1'.

The results are shown in Tables 1–3, where '✓' defines perspective correction succeeded and '×' defines it failed. As shown in Tables 1–3, there are three groups of T_1 and se that can satisfy all the scenarios in our experiment. Therefore, we chose one of the three groups. In our experiment, we set $T_1 = 0.65$ and se with 5×5 size.

Table 1. Resynchronizing Result of se with 5×5 Size.

T_1	Shooting Distance (cm)			
	10	20	30	40
0.55	×	✓	✓	✓
0.6	×	✓	✓	✓
0.65	✓	✓	✓	✓
0.7	✓	✓	✓	✓
0.75	✓	✓	✓	×
0.8	✓	×	×	×

Table 2. Resynchronizing result of se with 7×7 Size.

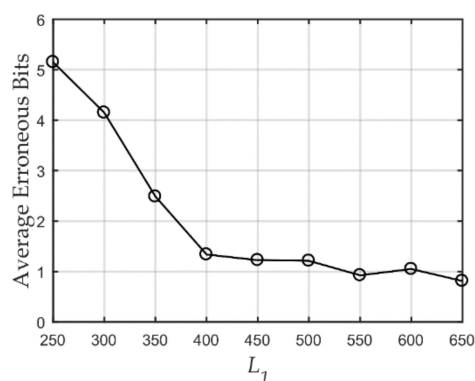
T_1	Shooting Distance (cm)			
	10	20	30	40
0.55	×	✓	✓	✓
0.6	✓	✓	✓	✓
0.65	✓	✓	✓	×
0.7	✓	✓	×	×
0.75	✓	×	×	×
0.8	✓	×	×	×

Table 3. Resynchronizing result of se with 9×9 Size.

T_1	Shooting Distance (cm)			
	10	20	30	40
0.55	✓	✓	✓	×
0.6	✓	✓	✓	×
0.65	✓	✓	×	×
0.7	✓	×	×	×
0.75	✓	×	×	×
0.8	×	×	×	×

3.1.3. Selection of Side Length L_1 of Detection Block for Watermark A Extraction

In theory, the larger the size of selected block $B(i)$ in watermark detection, the clearer the watermark information should be. However, considering the size of the original image is restricted, and the image needs to be divided into multiple blocks for watermark detection, the size of $B(i)$ should be as small as possible. To balance this contradiction, we analyzed the number of erroneous bits when using $B(i)$ with different side length L_1 in the watermark extraction. In this experiment, we set the shooting distance from 30 cm to 100 cm at an interval of 10 cm and the shooting angle to 0 degrees. Hence, 80 captured images of the 10 host images were utilized. The average erroneous bits with different L_1 are shown in Figure 14. When the L_1 was greater than 400, although the number of erroneous bits was lower, the variation tended to be stable. Therefore, to ensure low erroneous bits and also low side length, we set $L_1 = 400$.

**Figure 14.** Average erroneous bits with different L_1 .

3.1.4. Selection of the Fixed Value k_2 and k_3 for Message Extraction Threshold

According to Equations (6), (7), (11), and (12), the fixed value k_2 and k_3 are used to calculate the detection threshold for w'_B and w'_A , respectively, which can determine the validity of the message extraction result. Based on the 80 captured images mentioned in Section 3.1.3, we analyzed the number of erroneous bits with different thresholds. We analyzed the extracted result of synchronizing failed and unwatermarked. The number of

average erroneous bits when synchronizing failed or unwatermarked was independent of threshold, is shown in Figure 15. When $k_2 = 1.5$, as shown in Figure 15a, and $k_3 = 1.5$, as shown in Figure 15b, we achieved the minimum average erroneous bits in extracting w'_B and w'_A , respectively. Therefore, k_2 and k_3 were both set to 1.5.

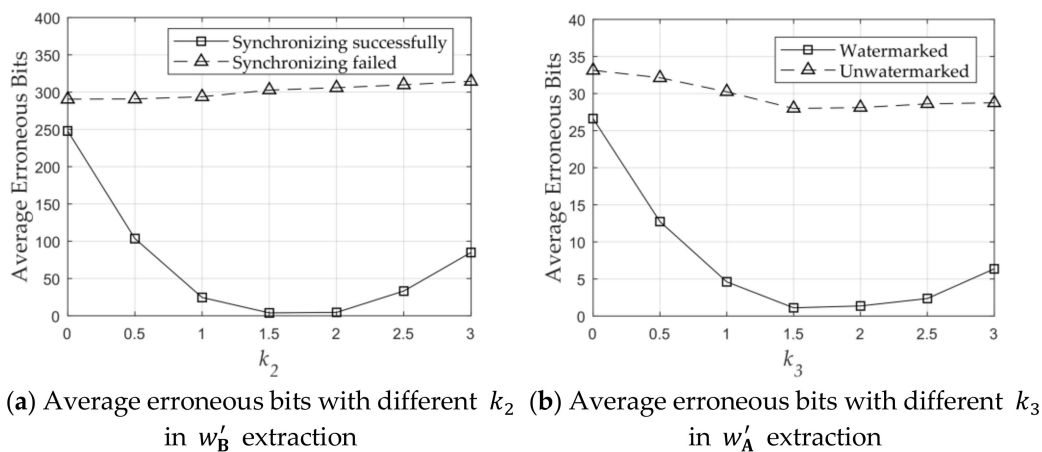


Figure 15. Erroneous bits with different message extraction thresholds.

3.2. Security of Encryption

We used three commonly used statistical analysis metrics [68,69] to measure encryption security. The experiment data was the 1000 images we obtained from Google Earth. First, we performed a correlation analysis. Because two adjacent pixels in a plain image are strongly correlated vertically and horizontally [3], a good encryption method needs to reduce this correlation, which means the correlation coefficient should be near to 0. The correlation coefficient between the encrypted image and decrypted image of the watermarked image is shown in Figure 16a, where the a-axis means the serial number of host images. The average correlation coefficient of the test images was 0.0091.

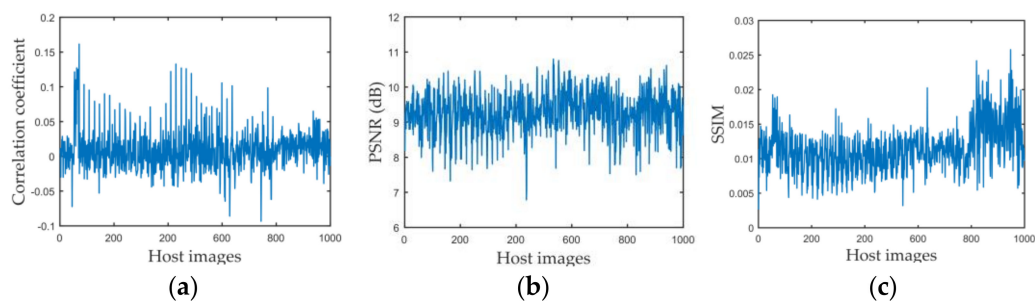


Figure 16. Evaluation of proposed encryption method. (a–c) are correlation coefficients, PSNR and SSIM values, respectively.

Then, the PSNR and SSIM were used to analyze image degeneration and similarity between the encrypted and decrypted images. The results are shown in Figure 16b,c. The average PSNR value was 9.2738 dB, and the average SSIM was 0.0113.

An ideal encryption scheme should be sensitive to the secret key, which means if a single bit in the original key is modified, the image remains unrecoverable. As our secret key was the combination of $X_O(0)$, $X_E(0)$, and u , we used different parameters to decrypt the image. An example is shown in Figure 17, where the first row is the decryption results, and the second row is the corresponding secret key used. The first image in Figure 17 is decrypted with the right key. Decryption with the wrong key cannot be recovered, even when the difference to original secret key is minimal.

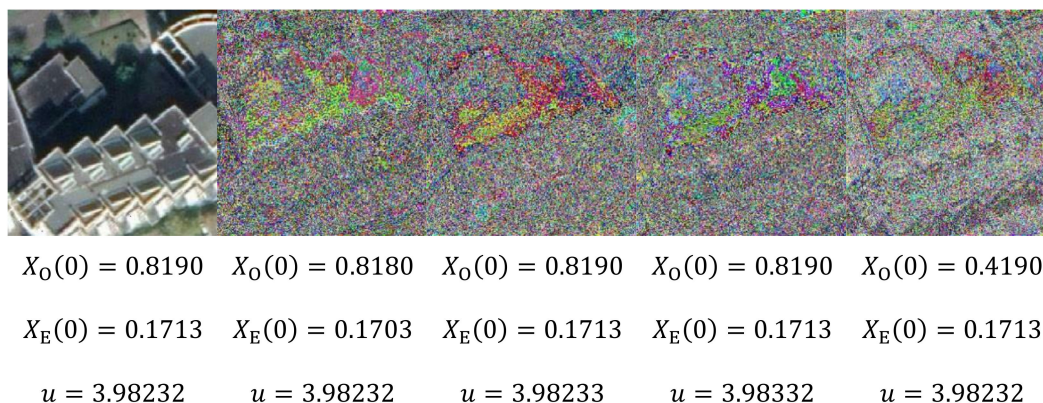


Figure 17. Decryption results with different key.

3.3. Robustness of Watermark in Encrypted Image against Screen-Cam Attack

This section verifies the robustness of watermark B in encrypted images against screen-cam attack with different shooting conditions. Considering the real use requirements, using a smartphone to read the watermark from an encrypted image in real time is similar to using a smartphone to scan a QR code, where the phone is usually close to the screen. Therefore, in our experiment, we set the shooting distance at {10 cm, 20 cm, 30 cm, 40 cm} and the shooting angle at {0°, 15°, 30°, 45°} of horizontal left.

We employed the commonly used metrics Bit Error Rate (BER) to measure robustness. BER is defined as the ratio of the number of erroneous bits to the length of the message sequence.

Table 4 lists the average BER in extracting the watermark from encrypted images with different shooting conditions. Table 5 shows a set of examples when the shooting angle was 45°. The encrypted images are not related to the original images, and all encrypted images are similar to the noise images. Therefore, the image itself does not have high magnitude coefficients at the embedding region of the DFT domain, which makes the coefficients of embedded watermark bit '1' significantly different from other coefficients. Hence, the BER can be maintained very low. The proposed method has high robustness to this situation.

Table 4. Average BER under different shooting conditions in watermark B extraction.

Shooting Horizontal Angle (Left)	Shooting Distance			
	10 cm	20 cm	30 cm	40 cm
0°	1.8/597	2.3/597	3/597	6.4/597
15°	2.8/597	3.2/597	4.3/597	6.7/597
30°	3.3/597	3.4/597	5/597	8.1/597
45°	3.6/597	6.8/597	8.8/597	11.5/597

When shooting at a long distance, the captured image may contain more interference factors, which will affect the automatic perspective correction. Table 6 lists some examples. These captured images cannot be automatically corrected with the proposed automatic perspective correction method. However, after simple cropping and scaling, the watermark can be effectively detected and extracted, as shown with the experiments. In practice, in real applications, we can design a zoom and partial cropping function for the watermark reading application to achieve watermark extraction at a long shooting distance.

Table 5. Examples of automatically extraction result of watermark B.



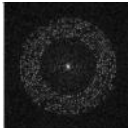


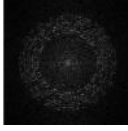


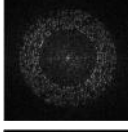










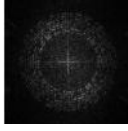




Shooting Condition	Captured Photo	Extracted Block	Magnitude Spectrum	BER
Shooting distance: 10 cm. Shooting angle: 45°				2/597
Shooting distance: 20 cm. Shooting angle: 45°				5/597
Shooting distance: 30 cm. Shooting angle: 45°				8/597
Shooting distance: 40 cm. Shooting angle: 45°				10/597

Table 6. Examples of extraction result of watermark B with manual operation.

Captured Image	Manual Cropping and Scaling	Extracted Block	Magnitude Spectrum	BER
				6/597
				7/597
				15/597

3.4. Robustness of Watermark in Decrypted Image Against Common Attacks

The proposed scheme is aimed at screen-cam attacks but, at the same time, it can resist common image processing attacks. In this section, we verify the robustness of watermark A in a decrypted image to common attacks and compare the proposed scheme with three existing schemes, which are all mainly designed for print-cam or screen-cam attacks. For fair comparison, we adjusted the parameters and the size of embedding blocks of the three algorithms accordingly. The block size was set to 256×256 in [32], and embedded message was 64 bits. The embedding unit of one bit was changed from 8×8 to 16×16 in [43], and the embedded message was 63 bits. The method of [44] embeds 93 bits. In comparison, we set the watermarking imperceptibility of these methods at the same level by adjusting the embedding strength to keep the PSNR values similar. An example is shown in Table 7.

Table 7. Watermarked images generated by different methods.





Methods	Pramila et al. [32]	Fang et al. [43]	Chen et al. [44]	Proposed
Image				
PSNR (dB)	40.3206	40.2210	40.7022	40.8513
SSIM	0.9589	0.9539	0.9661	0.9672

Table 8 lists the average BER of host images under different common image processing attacks, where—defines not robust to this attack. As shown in Table 8, the proposed scheme had better performance against most common image processing attacks.

Table 8. Average bit error rate (BER) under different common attacks.

Attacks	BER			
	Primila et al. [32]	Fang et al. [43]	Chen et al. [44]	Proposed
JPEG 40	43.91%	0.16%	0.11%	0.16%
JPEG 30	49.69%	4.76%	0.86%	1.11%
JPEG 20	50.31%	23.02%	5.59%	6.98%
Scaling 200%	49.84%	49.84%	0%	0%
Scaling 50%	47.66%	50.16%	—	0%
Scaling 40%	50.31%	51.43%	—	—
Rotation 10° + cropping	52.81%	49.68%	0.00%	0.00%
Rotation 15° + cropping	48.75%	48.25%	0.00%	0.00%
Rotation 30° + cropping	49.53%	52.06%	—	0.00%
Median filter 3 × 3	4.69%	0.00%	1.18%	0.00%
Median filter 4 × 4	9.69%	0.79%	10.32%	6.51%
Gaussian Noise (0.005)	19.06%	5.08%	7.10%	4.92%
Gaussian Noise (0.01)	28.59%	11.27%	20.54%	6.83%
Salt & Pepper (0.05)	35.00%	14.76%	2.58%	0.16%
Poisson	31.72%	3.33%	1.94%	0.63%
Speckle	39.22%	15.08%	1.94%	0.79%
Sharpening	0.63%	0.48%	1.08%	0.95%
Linear adjustment	0.63%	0.00%	0.00%	0.00%
Histogram equalization	0.47%	0.00%	0.00%	0.00%

The proposed method had high robustness to JPEG compression, where the message can still be recovered correctly under JPEG compression with QF = 20. With regard to scaling attack, we extracted the watermark message without correcting the image to its original scale. Method [32] was not robust to large scaling distortion. Method [43] needed to correct the image to original size, which was also not robust. The proposed method had better robustness than method [44] to scaling distortion. When scaling to 50%, only the proposed method could extract the watermark message completely. Rotation and cropping attack in Table 8 means the rotated image was cropped to the original size. Method [32,43] could not detect this kind of desynchronization. The synchronization method of [44] had limitations on the angle of rotation. The proposed method could resist any angle of rotation attack. With regard to median filter attack, although method [43] had the best performance, our method also performed well in comparison. Furthermore, the proposed scheme had good robustness to different types of noise attack and image enhancement process, and lower BER than the other three methods.

3.5. Robustness of Watermark in Decrypted Image against Screen-Cam Attack

In this section, the robustness of watermark A in the decrypted image against screen-cam attack is tested. First, we performed a comparison with the three methods mentioned above with different shooting distances and shooting angles. Because method [44] was designed for automatic perspective correction, for fair comparison, we manually corrected the captured image if the automatic correction algorithm did not work. When shooting direction was perpendicular to the screen, the average BER of all methods with different shooting distances is shown in Figure 18a. When shooting at a distance of 60 cm, the average BER of all methods with shooting angle from perpendicular to 60° of horizontal left is shown in Figure 18b. The proposed method and method [43,44] had similar robustness against screen-cam attack.

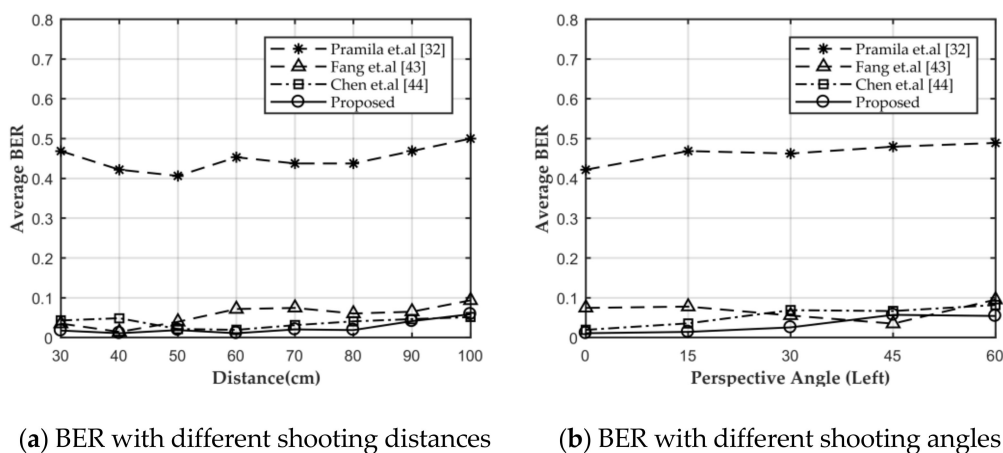


Figure 18. Comparison of different methods with different shooting conditions.

We verified the robustness of the proposed scheme with more shooting conditions. Because, in theory, the distortions caused by shooting at the same angle of horizontal perspective or vertical perspective are similar, only the distorted part in the host image is different. Therefore, in this experiment, we set the shooting from being perpendicular to the screen up to 60° of horizontal left at intervals of 15°. Besides, the shooting distance was set from 30 cm to 100 cm at intervals of 10 cm. When shooting with an angle to capture the whole image, the closest shooting distance was adjusted to 40 cm. Experimental results are shown in Table 9, where the average BER did not include the case where the tracking sequence was not detected, and ‘/’ defines the tracking sequence is not detected in all captured images. Figure 19 shows the watermark detection result of different host images. Table 10 lists the recovered image NNU from captured images with different shooting conditions and the corresponding BER.

Table 9. Average EBR with different shooting conditions.

Horizontal Angle (Left)	Shooting Distance							
	30 cm	40 cm	50 cm	60 cm	70 cm	80 cm	90 cm	100 cm
0°	1.3/63	0.8/63	1.2/63	0.7/63	1.3/63	1.2/63	2.6/63	3.7/63
15°		1.3/63	1.4/63	0.9/63	1.5/63	1.7/63	2.3/63	8.8/63
30°		3.3/63	2.3/63	1.6/63	2.2/63	2.6/63	5.4/63	11.5/63
45°		2.4/63	3.3/63	3.6/63	3.9/63	5.8/63	/	/
60°		3.0/63	3.6/63	4.9/63	11.7/63	/	/	/

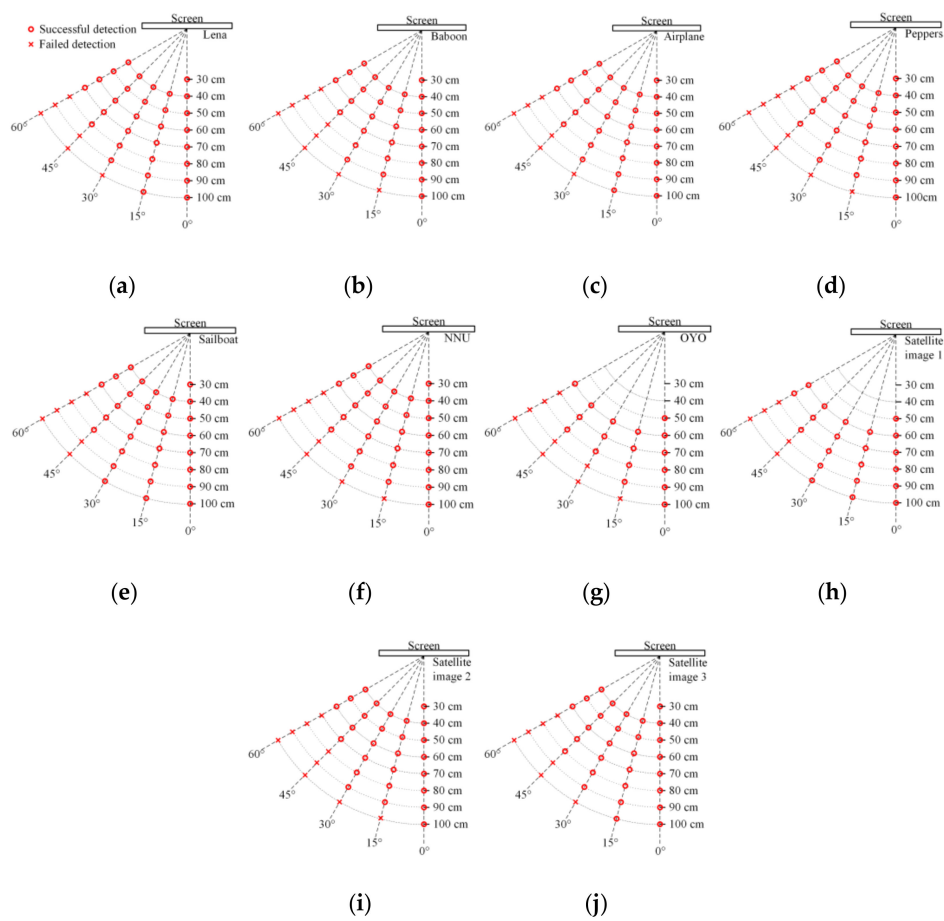


Figure 19. Watermark detection results against screen-cam attack. (a–j) are the detection result of Lena, Baboon, Airplane, Peppers, Sailboat, NNU, OYO, Satellite image 1, Satellite image 2, and Satellite image 3, respectively.




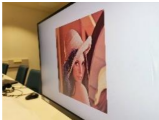




Table 10. Examples of NNU recovered from different captured images.

Horizontal Angle (Left)	Shooting Distance							
	30 cm	40 cm	50 cm	60 cm	70 cm	80 cm	90 cm	100 cm
0°								
BER	0/63	1/63	2/63	0/63	0/63	0/63	0/63	3/63
15°								
BER		1/63	0/63	0/63	1/63	1/63	5/63	11/63
30°								
BER		2/63	1/63	2/63	1/63	1/63	4/63	8/63
45°								
BER		1/63	1/63	3/63	1/63	5/63	/	/
60°								
BER		3/63	5/63	6/63	9/63	/	/	/

Because OYO and satellite image 1 are big size images, to capture the whole host image, the closest shooting distance was adjusted as shown in Figure 19g,h. When shooting perpendicular to the screen, the watermark could be extracted at all shooting distances with low BERs. When the shooting angles were 15° and 30°, the watermark could be extracted basically at a shooting distance below 90 cm, also with low BERs. When the shooting angle was 45°, the watermark could be extracted from most captured images taken within 80cm. When shooting at a large angle of 60°, the watermark could still survive at a close shooting distance.

The captured images in the experiment above were obtained with the help of a tripod. In a real scene, we captured the images by holding a smartphone, which causes camera shake and leads to more blurring. Therefore, we also test the performance with handheld shooting. The results of some cases are shown in Table 11, showing good performance.

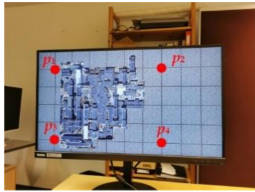

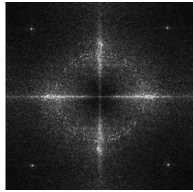

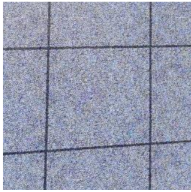
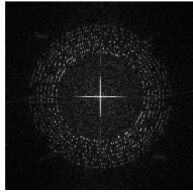
Table 11. Examples of handheld shooting.

Handhold Scenarios	Example 1	Example 2	Example 3	Example 4
Captured image				
Recovered image				
BER	3/63	5/63	1/63	2/63

3.6. Robustness of Watermark in Partial Decrypted Image Against Screen-Cam Attack

For a screen-cam partially decrypted image, we can extract the watermark information from both the encrypted part or the decrypted part. Therefore, in essence, the verification of robustness of watermark in a partial decrypted image is the same as Sections 3.3 and 3.5. Two examples are shown in Table 12. The partial decrypted image has an advantage. Because the size of the encrypted block is known, the corner points of the encrypted blocks can be used as reference points for perspective correction. As shown in the first example, the image used for detection is perspective-corrected by the four points marked in the captured image. An example of magnitude spectrum of selected detection block is shown following. If we use the encrypted part for watermark extraction, we can crop out the needed part directly. As shown in the second example, we cut out the part marked by the red box in the captured image for detection. Both methods can achieve good performance.

Table 12. Watermark extractions from partial decrypted image.

Captured Image	Used for Detection	Magnitude Spectrum	BER
			4/63
			3/597

4. Discussion

4.1. Characteristic of Screen-Cam Robust Watermarking

The screen-cam process causes severe image quality degradation [45]. In other words, we need to improve the robustness of the watermarking algorithm to deal with a screen-cam attack. A robust watermarking algorithm has three mutually restrictive characteristics [70]: robustness, imperceptibility, and watermark capacity. Under these circumstances, commonly, we need to sacrifice some watermark capacity or imperceptibility to meet the screen-cam robust requirements. For example, the length of message sequences in [43,44,65] were only 63, 60, and 94 bits, which are less than normal. Besides, these methods all embedded the message repeatedly to deal with the loss of detailed information during screen-cam process.

In the proposed method, we also employed the above ideas to achieve screen-cam robust of watermark A in a decrypted image. Furthermore, we designed a DFT-based global watermarking algorithm to deal with the loss of detailed information during screen-cam process. As we mentioned in Section 2.2.2, by employing this method, we could select a block larger than one watermark embedded block to contain more detailed information for watermark extraction.

The characteristic of watermark B against screen-cam attack in the encrypted image is special, that is because the encrypted image is a noise-like image. If the encrypted image can be modulated into a noise image similar to the meaningless watermark pattern, this is equivalent to enhancing the perception of the watermark and the robustness is significantly improved. Therefore, it provides the possibility to increase the watermark capacity. Based on this, we can design a QR code-based watermark generation method that contains a message sequence of 620 bits.

4.2. Analysis of Joint Encryption and Watermarking Mechanism

How to combine encryption and watermarking technology is a scientific issue. In previous research, the encryption and watermarking worked independently to a certain extent or watermarking was limited by the method itself. The previous joint encryption and watermarking methods were mainly divided into two categories: CEW and RDH-EI.

CEW methods can be further divided into three types [71]. The first one is based on different data fields, which means two independent parts are used for encryption and watermarking respectively [72,73]. Therefore, to some degree, encryption and watermarking work independently. The second type is invariant-based, where the watermark is embedded in a subset that is invariant before and after encryption [11,74]. However, the robustness is also limited by the used invariants. For example, because global histogram statistics are invariable after encryption by scrambling pixel positions, [11] employed a histogram-based watermarking method to achieve CEW. However, the histogram-based method is susceptible to cropping attacks and certainly not applicable for screen-cam attack. The third type of CEW is based on homomorphic encryption, where algebraic operations on the original data can be realized by performing (possibly different) algebraic operations on the encrypted data [75]. Similarly, homomorphic-based CEW is limited by the method itself. Because the algebraic operations that can achieve homomorphism are limited, the corresponding watermarking algorithms that can be designed are also limited.

RDH-EI methods distinguish between content owner and data hider [17], where data hider can only read the reversible watermark but cannot access the encrypted data. Most RDH-EI methods can be divided into two frameworks: vacating room after encryption and reserving room before encryption. Therefore, to some degree, the encryption and watermarking of RDH-EI methods also work independently.

The joint encryption and watermarking mechanism of the proposed scheme is different from CEW or RDH-EI. We embedded the watermark through odd-even quantization and encrypted odd and even to different numerical ranges. In this way, the encryption method could enhance the perceptibility of watermark B in the encrypted image, thereby achieving screen-cam robust. In addition, as we mentioned in Section 2.1.2, the watermarked and en-

encrypted image could effectively avoid the weakness of the proposed encryption algorithm compared to the only-encrypted image. Therefore, this proposed design achieved the mutual cooperation of encryption and watermarking technologies. However, there is no doubt that the design of encryption and watermarking methods are still mutually restricted.

In practical applications, the joint mechanisms of encryption and watermarking are neither superior nor inferior to each other. The joint mechanism needs to be decided according to the requirements of algorithm design. In order to meet more application scenarios and requirements, the joint encryption and watermarking mechanism is worthy of further exploration.

5. Conclusions

This paper proposes a joint encryption and screen-cam robust watermarking scheme, which can achieve watermark extraction from both encrypted and decrypted images taken by a smartphone. In watermark embedding and image encryption, first we embed a watermark A with a DFT-based algorithm, then the watermark B was generated based on QR encoding and IDFT to achieve high watermark capacity and error correction ability. After that, watermark B was hidden in the changes caused by embedding watermark A, which can improve imperceptibility and does not affect the effectiveness of watermark A. Finally, a chaotic mapping-based segment encryption algorithm was proposed, which can match with watermark B and enhance its robustness after encryption. With respect to watermark detection from an encrypted image, a frame detection method was utilized to achieve watermark synchronization. With respect to watermark detection from the decrypted image, we used a large size of block and searched the tracking sequence based on NCC coefficients to locate the watermark message. The watermark messages were all extracted from the noise component with a local statistic feature. The proposed scheme is proved to have a high robustness to the screen-cam process before and after decryption, and also has a remarkable performance against common image processing attacks after decryption.

Author Contributions: Conceptualization, W.C. and N.R.; methodology, W.C., N.R. and C.Z.; software, W.C. and Q.Z.; data curation, W.C., A.K. and T.S.; writing—original draft preparation, W.C. and N.R.; writing—review and editing, A.K. and T.S.; funding acquisition, C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 42071362 and 41971338, the Natural Science Foundation of Jiangsu Province, grant number BK20191373.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, H.; Zhao, B.; Huang, L. A remote-sensing image encryption scheme using DNA bases probability and two-dimensional logistic map. *IEEE Access* **2019**, *7*, 65450–65459. [[CrossRef](#)]
2. Zhang, X.; Wang, X. Remote-sensing image encryption algorithm using the advanced encryption standard. *Appl. Sci.* **2018**, *8*, 1540. [[CrossRef](#)]
3. Ghadirli, H.M.; Nodehi, A.; Enayatifar, R. An overview of encryption algorithms in color images. *Signal Process.* **2019**, *164*, 163–185. [[CrossRef](#)]
4. Kumari, M.; Gupta, S.; Sardana, P. A Survey of Image Encryption Algorithms. *3D Res.* **2017**, *8*, 37. [[CrossRef](#)]
5. Xu, G.; Li, H.; Dai, Y.; Yang, K.; Lin, X. Enabling efficient and geometric range query with access control over encrypted spatial data. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 870–885. [[CrossRef](#)]
6. Jha, S.; Sural, S.; Atluri, V.; Vaidya, J. Specification and verification of separation of duty constraints in attribute-based access control. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 897–911. [[CrossRef](#)]
7. Xue, K.; Chen, W.; Li, W.; Hong, J.; Hong, P. Combining data owner-side and cloud-side access control for encrypted cloud storage. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2062–2074. [[CrossRef](#)]

8. Castiglione, A.; De Santis, A.; Masucci, B.; Palmieri, F.; Castiglione, A.; Huang, X. Cryptographic hierarchical access control for dynamic structures. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2349–2364. [[CrossRef](#)]
9. Liu, H.; Xiao, D.; Zhang, R.; Zhang, Y.; Bai, S. Robust and hierarchical watermarking of encrypted images based on compressive sensing. *Signal Process. Image Commun.* **2016**, *45*, 41–51. [[CrossRef](#)]
10. Jiang, L.; Xu, Z.; Xu, Y. Commutative encryption and watermarking based on orthogonal decomposition. *Multimed. Tools Appl.* **2014**, *70*, 1617–1635. [[CrossRef](#)]
11. Schmitz, R.; Li, S.; Grecos, C.; Zhang, X. Towards robust invariant commutative watermarking-encryption based on image histograms. *Int. J. Multimed. Data Eng. Manag. (IJMDEM)* **2014**, *5*, 36–52. [[CrossRef](#)]
12. Jiang, L.; Xu, Z. Commutative encryption and watermarking for remote sensing image. *Int. J. Digit. Content Technol. Appl.* **2012**, *6*, 197–205.
13. Cancellaro, M.; Battisti, F.; Carli, M.; Boato, G.; De Natale, F.G.; Neri, A. A commutative digital image watermarking and encryption method in the tree structured Haar transform domain. *Signal Process. Image Commun.* **2011**, *26*, 1–12. [[CrossRef](#)]
14. Li, M.; Xiao, D.; Zhu, Y.; Zhang, Y.; Sun, L. Commutative fragile zero-watermarking and encryption for image integrity protection. *Multimed. Tools Appl.* **2019**, *78*, 22727–22742. [[CrossRef](#)]
15. Jiang, L. The identical operands commutative encryption and watermarking based on homomorphism. *Multimed. Tools Appl.* **2018**, *77*, 30575–30594. [[CrossRef](#)]
16. Liu, J.; Zhao, K.; Zhang, R. A fully reversible data hiding scheme in encrypted images based on homomorphic encryption and pixel prediction. *Circuits Syst. Signal Process.* **2020**, *39*, 3532–3552. [[CrossRef](#)]
17. Huang, D.; Wang, J. High-capacity reversible data hiding in encrypted image based on specific encryption process. *Signal Process. Image Commun.* **2020**, *80*, 115632. [[CrossRef](#)]
18. Qiu, Y.; Qian, Z.; Zeng, H.; Lin, X.; Zhang, X. Reversible data hiding in encrypted images using adaptive reversible integer transformation. *Signal Process.* **2020**, *167*, 107288. [[CrossRef](#)]
19. Senthilnathan, T.; Prabu, P.; Sivakumar, R.; Sakthivel, S. An enhancing reversible data hiding for secured data using shuffle block key encryption and histogram bit shifting in cloud environment. *Clust. Comput.* **2019**, *22*, 12839–12847. [[CrossRef](#)]
20. Ge, H.; Chen, Y.; Qian, Z.; Wang, J. A high capacity multi-level approach for reversible data hiding in encrypted images. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 2285–2295. [[CrossRef](#)]
21. Zhang, R.; Lu, C.; Liu, J. A high capacity reversible data hiding scheme for encrypted covers based on histogram shifting. *J. Inf. Secur. Appl.* **2019**, *47*, 199–207. [[CrossRef](#)]
22. Wu, H.-T.; Cheung, Y.-M.; Yang, Z.; Tang, S. A high-capacity reversible data hiding method for homomorphic encrypted images. *J. Vis. Commun. Image Represent.* **2019**, *62*, 87–96. [[CrossRef](#)]
23. Chen, Y.-C.; Hung, T.-H.; Hsieh, S.-H.; Shiu, C.-W. A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 3332–3343. [[CrossRef](#)]
24. Shiu, P.-F.; Tai, W.-L.; Jan, J.-K.; Chang, C.-C.; Lin, C.-C. An interpolative AMBTC-based high-payload RDH scheme for encrypted images. *Signal Process. Image Commun.* **2019**, *74*, 64–77. [[CrossRef](#)]
25. Liu, L.; Zhang, Z.; Chen, R. Cryptanalysis and improvement in a plaintext-related image encryption scheme based on hyper chaos. *IEEE Access* **2019**, *7*, 126450–126463. [[CrossRef](#)]
26. Puteaux, P.; Puech, W. An efficient MSB prediction-based method for high-capacity reversible data hiding in encrypted images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1670–1681. [[CrossRef](#)]
27. Menendez-Ortiz, A.; Feregrino-Uribe, C.; Hasimoto-Beltran, R.; Garcia-Hernandez, J.J. A survey on reversible watermarking for multimedia content: A robustness overview. *IEEE Access* **2019**, *7*, 132662–132681. [[CrossRef](#)]
28. Shi, Y.-Q.; Li, X.; Zhang, X.; Wu, H.-T.; Ma, B. Reversible data hiding: Advances in the past two decades. *IEEE Access* **2016**, *4*, 3210–3237. [[CrossRef](#)]
29. Pramila, A.; Keskinarkaus, A.; Seppänen, T. Increasing the capturing angle in print-cam robust watermarking. *J. Syst. Softw.* **2018**, *135*, 205–215. [[CrossRef](#)]
30. Pramila, A.; Keskinarkaus, A.; Takala, V.; Seppänen, T. Extracting watermarks from printouts captured with wide angles using computational photography. *Multimed. Tools Appl.* **2016**, *76*, 16063–16084. [[CrossRef](#)]
31. Keskinarkaus, A.; Pramila, A.; Seppänen, T. Image watermarking with feature point based synchronization robust to print-scan attack. *J. Vis. Commun. Image Represent.* **2012**, *23*, 507–515. [[CrossRef](#)]
32. Pramila, A.; Keskinarkaus, A.; Seppänen, T. Toward an interactive poster using digital watermarking and a mobile phone camera. *Signal Image Video Process.* **2011**, *6*, 211–222. [[CrossRef](#)]
33. Keskinarkaus, A.; Pramila, A.; Seppänen, T. Image watermarking with a directed periodic pattern to embed multibit messages resilient to print-scan and compound attacks. *J. Syst. Softw.* **2010**, *83*, 1715–1725. [[CrossRef](#)]
34. Nakamura, T.; Katayama, A.; Yamamuro, M.; Sonehara, N. Fast watermark detection scheme for camera-equipped cellular phone. In Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, College Park, ML, USA, 27–29 October 2004; Association for Computing Machinery: New York, NY, USA, 2004; pp. 101–108.
35. Katayama, A.; Nakamura, T.; Yamamuro, M.; Sonehara, N. New high-speed frame detection method: Side Trace Algorithm (STA) for i-appli on cellular phones to detect watermarks. In Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia, College Park, ML, USA, 27–29 October 2004; Association for Computing Machinery: New York, NY, USA, 2004; pp. 109–116.

36. Gourrame, K.; Douzi, H.; Harba, R.; Riad, R.; Ros, F.; Amar, M.; Elhajji, M. A zero-bit Fourier image watermarking for print-cam process. *Multimed. Tools Appl.* **2019**, *78*, 2621–2638. [[CrossRef](#)]
37. Gourrame, K.; Douzi, H.; Harba, R.; Ros, F.; El Hajji, M.; Riad, R.; Amar, M. Robust print-cam image watermarking in fourier domain. In Proceedings of the International Conference on Image and Signal Processing, Trois-Rivières, QC, Canada, 30 May–1 June 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 356–365.
38. Riad, R.; Harba, R.; Douzi, H.; Ros, F.; Elhajji, M. Robust fourier watermarking for id images on smart card plastic supports. *Adv. Electr. Comput. Eng.* **2016**, *16*, 23–30. [[CrossRef](#)]
39. Mirza, M.T.; Ahmed, Q.; Munib, S.; Khan, A.; Khalil, R.K. A new hybrid domain based print-scan resilient image watermarking technique. In Proceedings of the 12th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 17–19 December 2014; pp. 170–175.
40. Jassim, T.; Abd-Alhameed, R.; Al-Ahmad, H. A new robust and fragile watermarking scheme for images captured by mobile phone cameras. In Proceedings of the 1st International Conference on Communications, Signal Processing, and Their Applications ICCSPA 2013, Sharjah, UAE, 12–14 February 2013; pp. 1–5.
41. Pramila, A.; Keskinarkaus, A.; Seppänen, T. Multiple domain watermarking for print-scan and JPEG resilient data hiding. In Proceedings of the International Workshop on Digital Watermarking, Busan, Korea, 10–12 November 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 279–293.
42. Keskinarkaus, A.; Pramila, A.; Seppänen, T.; Sauvola, J. Wavelet domain print-scan and JPEG resilient data hiding method. In Proceedings of the International Workshop on Digital Watermarking, Busan, Korea, 10–12 November 2008; Springer: Berlin/Heidelberg, Germany, 2006; pp. 82–95.
43. Fang, H.; Zhang, W.; Zhou, H.; Cui, H.; Yu, N. Screen-shooting resilient watermarking. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 1403–1418. [[CrossRef](#)]
44. Chen, W.; Zhu, C.; Ren, N.; Seppänen, T.; Keskinarkaus, A. Screen-Cam Robust and Blind Watermarking for Tile Satellite Images. *IEEE Access* **2020**, *8*, 125274–125294. [[CrossRef](#)]
45. Schaber, P.; Kopf, S.; Wetzel, S.; Ballast, T.; Wesch, C.; Effelsberg, W. CamMark: Analyzing, modeling, and simulating artifacts in camcorder copies. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2015**, *11*, 1–23. [[CrossRef](#)]
46. Fang, H.; Chen, D.; Huang, Q.; Zhang, J.; Ma, Z.; Zhang, W.; Yu, N. Deep Template-based Watermarking. *IEEE Trans. Circuits Syst. Video Technol.* **2020**. [[CrossRef](#)]
47. Wang, X.; Zhao, H.; Wang, M. A new image encryption algorithm with nonlinear-diffusion based on multiple coupled map lattices. *Opt. Laser Technol.* **2019**, *115*, 42–57. [[CrossRef](#)]
48. Alawida, M.; Samsudin, A.; Teh, J.S.; Alkhaldeh, R.S. A new hybrid digital chaotic system with applications in image encryption. *Signal Process.* **2019**, *160*, 45–58. [[CrossRef](#)]
49. Li, Y.; Wang, C.; Chen, H. A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation. *Opt. Lasers Eng.* **2017**, *90*, 238–246. [[CrossRef](#)]
50. Mishra, D.C.; Sharma, R.; Suman, S.; Prasad, A. Multi-layer security of color image based on chaotic system combined with RP2DFRFT and Arnold Transform. *J. Inf. Secur. Appl.* **2017**, *37*, 65–90. [[CrossRef](#)]
51. Pak, C.; Huang, L. A new color image encryption using combination of the 1D chaotic map. *Signal Process.* **2017**, *138*, 129–137. [[CrossRef](#)]
52. Hamza, R.; Titouna, F. A novel sensitive image encryption algorithm based on the Zaslavsky chaotic map. *Inf. Secur. J. Glob. Perspect.* **2016**, *25*, 162–179. [[CrossRef](#)]
53. Wang, X.; Zhang, H.-L. A color image encryption with heterogeneous bit-permutation and correlated chaos. *Opt. Commun.* **2015**, *342*, 51–60. [[CrossRef](#)]
54. Wang, B.; Xie, Y.; Zhou, C.; Zhou, S.; Zheng, X. Evaluating the permutation and diffusion operations used in image encryption based on chaotic maps. *Optik* **2016**, *127*, 3541–3545. [[CrossRef](#)]
55. Wang, L.; Song, H.; Liu, P. A novel hybrid color image encryption algorithm using two complex chaotic systems. *Opt. Lasers Eng.* **2016**, *77*, 118–125. [[CrossRef](#)]
56. Wang, X.; Zhao, Y.; Zhang, H.; Guo, K. A novel color image encryption scheme using alternate chaotic mapping structure. *Opt. Lasers Eng.* **2016**, *82*, 79–86. [[CrossRef](#)]
57. Liu, H.; Kadir, A. Asymmetric color image encryption scheme using 2D discrete-time map. *Signal Process.* **2015**, *113*, 104–112. [[CrossRef](#)]
58. Chen, J.-H.; Chen, W.-Y.; Chen, C.-H. Identification recovery scheme using quick response (QR) code and watermarking technique. *Appl. Math. Inf. Sci.* **2014**, *8*, 585. [[CrossRef](#)]
59. Su, Y.; Han, P.; Jia, Z.J.; Liang, S.H. Digital watermarking based on two-way arnold transform and QR code. *Adv. Mater. Res.* **2012**, *591–593*, 2564–2567. [[CrossRef](#)]
60. Kang, Q.; Li, K.; Yang, J. A digital watermarking approach based on DCT domain combining QR code and chaotic theory. In Proceedings of the 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 4–6 September 2014; pp. 331–337.
61. Information Capacity and Versions of the QR Code. Available online: <https://www.qrcode.com/en/about/version.html> (accessed on 7 October 2020).
62. Hua, Z.; Zhou, Y. Image encryption using 2D Logistic-adjusted-Sine map. *Inf. Sci.* **2016**, *339*, 237–253. [[CrossRef](#)]

63. Liu, Y.; Tang, S.; Liu, R.; Zhang, L.; Ma, Z. Secure and robust digital image watermarking scheme using logistic and RSA encryption. *Expert Syst. Appl.* **2018**, *97*, 95–105. [[CrossRef](#)]
64. Voloshynovskiy, S.V.; Pereira, S.; Herrigel, A.; Baumgartner, N.; Pun, T. Generalized watermarking attack based on watermark estimation and perceptual remodulation. *Proc. SPIE Secur. Watermarking Multimed. Contents II* **2000**, *3971*, 358–370.
65. Chen, W.; Ren, N.; Zhu, C.; Zhou, Q.; Seppänen, T.; Keskinarkaus, A. Screen-Cam robust image watermarking with feature-based synchronization. *Appl. Sci.* **2020**, *10*, 7494. [[CrossRef](#)]
66. Related Images of the Experiments. Available online: <http://decsai.ugr.es/cvg/dbimágenes/c512.php> (accessed on 6 October 2020).
67. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)]
68. Kaur, M.; Kumar, V. A comprehensive review on image encryption techniques. *Arch. Comput. Method Eng.* **2020**, *27*, 15–43. [[CrossRef](#)]
69. Geetha, S.; Punithavathi, P.; Infanteena, A.M.; Sindhu, S.S.S. A Literature Review on Image Encryption Techniques. *Int. J. Inf. Secur. Privacy (IJISP)* **2018**, *12*, 42–83. [[CrossRef](#)]
70. Begum, M.; Uddin, M.S. Digital Image Watermarking Techniques: A Review. *Information* **2020**, *11*, 110. [[CrossRef](#)]
71. Ren, N.; Zhu, C.; Tong, D.; Chen, W.; Zhou, Q. Commutative encryption and watermarking algorithm based on feature invariants for secure vector map. *IEEE Access* **2020**, *8*, 221481–221493. [[CrossRef](#)]
72. Guan, B.; Xu, D.; Li, Q. An Efficient commutative encryption and data hiding scheme for HEVC video. *IEEE Access* **2020**, *8*, 60232–60245. [[CrossRef](#)]
73. Zhang, X. Commutative reversible data hiding and encryption. *Secur. Commun. Netw.* **2013**, *6*, 1396–1403. [[CrossRef](#)]
74. Boato, G.; Conci, N.; Conotter, V.; Natale, F.D.; Fontanari, C. Multimedia asymmetric watermarking and encryption. *Electron. Lett.* **2008**, *44*, 601–602. [[CrossRef](#)]
75. Boho, A.; Van Wallendael, G.; Dooms, A.; De Cock, J.; Braeckman, G.; Schelkens, P.; Preneel, B.; Van de Walle, R. End-To-End security for video distribution: The combination of encryption, watermarking, and video adaptation. *IEEE Signal Proc. Mag.* **2013**, *30*, 97–107. [[CrossRef](#)]