



Research article

Two-level QR code scheme based on region matrix image secret sharing algorithm

Li-na Zhang*, Jia-qi Sun, Xiao-yu Zhang, Qing-peng Chen and Jing Zhang

College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710600, China

* **Correspondence:** Email: zhangln@xust.edu.cn; Tel: +8617795724270.

Abstract: Quick response (QR) codes have become increasingly popular as a medium for quickly and easily accessing information through mobile devices. However, the open-source nature of QR code encoding poses a risk of information leakage and potential attacks, especially with the growing use of QR codes in financial services and authentication applications. To mitigate the risk of information leakage due to open-source QR code encoding, this paper proposes a two-level QR code scheme based on a region matrix image secret sharing algorithm. In this scheme, the first-level public information can be directly obtained by scanning with any standard QR code scanner, while the two-level secret information can only be accessed by overlaying the shared images. To enhance the robustness of joint secret information recovery using shared images, this article designs a progressive image secret sharing algorithm based on region matrices. This algorithm meticulously processes high-priority share regions and generates multiple substitute shares. In the event of attacks on key shares, substitute shares can be employed to recover the secret information. For an increased payload capacity of each QR code, an adaptive pixel depth adjustment algorithm is devised. This algorithm ensures that the recovery of two-level secret information maintains high clarity, while not affecting the scanning functionality of each shared QR code. Experimental results validate the feasibility of this scheme, which simplifies the construction matrix, reduces matrix redundancy, and exhibits priority partitioning and higher robustness. Furthermore, QR codes embedding secret shares can safeguard the two-level information, and the recovered images exhibit exceptional clarity.

Keywords: QR code; visual secret sharing; two-level QR code; robustness

1. Introduction

In Chinese national standards, the QR code is known as the Quick Response code. As the mainstream electronic tag, it has been widely utilized in various fields due to its advantages of low cost and convenient information extraction. Due to the open coding and decoding technology of the QR code, the barrier to entry is low, and any user can employ software to generate a QR code, leading to security risks such as personal information leakage and phishing links in the QR code's application. To bolster the authenticity verification functionality of QR codes, numerous scholars are currently delving into information concealment strategies rooted in QR codes. Their primary develops schemes to integrate authentication information into the QR codes.

The information hiding techniques on QR codes mainly include digital steganography [1–4] and digital watermarking [5,6]. Least significant bits (LSB) algorithm is one of the classic digital steganography algorithms, which embeds information by modifying the least significant bit of the carrier image [1,2]. Since only minor modifications are made to the image, the damage to the QR code is relatively small, and has thus been applied within a certain range. However, during transmission, the hidden information may not be extracted after the QR code is compressed. References [3,4] directly embed encrypted authentication information into the QR code by utilizing QR code encoding rules, borrowing the robustness of the QR code itself to protect the information from being destroyed. However, the capacity of the embedded information is limited and the ciphertext is easy to leak. Compared with digital steganography, digital watermarking has certain advantages in robustness and security, and is therefore often used for message verification. Kang et al. [5] proposed a new method of embedding digital watermarks using discrete cosine transform (DCT) to improve the deficiency that QR codes may overflow in the DCT domain. In the preprocessing stage, the QR code image is optimized by blurring and adding noise; however multiple watermark embeddings may cause image distortion on the carrier of the QR code. Reference [6] divides the QR code image into blocks, performs DCT on each block, takes the central block for singular value decomposition (SVD), and finally embeds the encrypted watermark image into the characteristic values obtained by SVD. This method ensures the invisibility of the watermark and effectively resists common attacks, though there is still room for improvement in terms of the payload capacity of the information. Furthermore, in [7], the authors utilize the error correction capabilities of QR codes to embed the QR code of the secret information, which is generated based on the secret payload, into the QR code of the public information. This achieves information hiding on the QR code. However, since the embedding capacity in this scheme relies on the error correction properties of QR codes, the capacity for embedding is limited.

Graded QR codes are a way of hiding information based on QR code encoding technology. It can store longer information content by distributing it into multiple carrier QR codes, thus achieving hierarchical management and transmission of information. Each carrier QR code contains a portion of complete information data, and by obtaining all the QR codes, the complete information content can be restored. Graded QR codes can be flexibly combined according to the requirements to meet the information management and transmission needs in different scenarios.

Therefore, to increase the payload capacity of secret information embedded in QR codes, Lin et al. [8] proposed a threshold secret sharing scheme. This scheme generates n -bit streams using a random number generator and hash algorithm, and then hides them in the carrier image using a wet paper code to generate n two-level QR codes, which can resist noise interference during the printing process. However, the embedding rate of secret information was not adequately considered, and the secret capacity remained low. Tkachenko et al. [9] embedded secret information in QR codes using texture passwords by modifying the black modules of the QR code into different specified textures. This scheme gives QR codes a two-layer information storage function, though the second storage

mechanism has a limited capacity due to the black pixel limitation. The visual secret sharing scheme based on QR code (VSSQR) proposed by Wan et al. [10] recovers hidden information by superimposing a sufficient number of QR codes. The recovered result can be directly recognized by the human eye, but visual distortion may occur when recovering the secret image. The scheme proposed by Cheng et al. [11] obtains secret information by directly performing exclusive OR (XOR) on encrypted QR codes, though the secret image format in this scheme is limited to binary images, and the effective payload capacity is small. Liu et al. [12] designed a secret sharing unit, which consists of 2×2 sub-modules and hides two-level information using the black-and-white sub-module ratio. However, the QR code shares generated by this scheme have a large pixel expansion ratio, making them susceptible to the scanning distance and angle, which increases the difficulty of QR code recognition.

In 2013, Chu et al. [13] extended the single pixel block by utilizing QR code key pixels to replace one QR code module with a 3×3 sub-module. The central module stores unchanged QR code information, and the remaining modules embed a secondary halftone image. This method greatly extends the payload capacity of secret information, and embedded images can be clearly identified without compromising the readability of the QR code. Many researchers combine this method with traditional cryptography and visual cryptography. Yu et al. [14,15] further improved Chu's method by combining visual cryptography to hide and recover secret images using QR codes. Fu et al. [16] designed a three-level QR code based on ideal XOR visual cryptography scheme (XVCS), which enables perfect recovery of the privacy QR code. These approaches expand the storage capacity of QR codes by eight times and effectively increase the payload capacity of secret information. However, these methods cannot be flexibly combined to meet different application scenarios; as the complexity of the usage scenarios increases, there is a greater demand for the clarity of secret image recovery and the payload capacity of secret information.

Most existing schemes in the literature that utilize visual cryptography to achieve a higher payload capacity for secret information in hierarchical QR codes pay little attention to two categories of issues. (1) In practical applications, image secret sharing and recovery often involve complex backgrounds. Single-structured secret sharing schemes lack flexibility in the combination of shared images during secret reconstruction, making them unsuitable for complex application scenarios or with low robustness. (2) Using QR codes as carriers for secret sharing, to ensure the readability of QR codes, the clarity of the recovered secret image during secret reconstruction is often compromised.

Therefore, in order to address the aforementioned issues, this paper proposes a two-level QR code scheme based on a region matrix image secret sharing algorithm, and is summarized as follows implementing different data recovery priorities for each share, embedding the shares into the carrier QR code to construct a two-level secure QR code, and proposing an adaptive pixel depth adjustment algorithm to increase the information payload of the QR code while ensuring high clarity during the recovery of the two-level QR code. The design idea of this scheme is as follows: (1) using the second-order matrix corresponding to the secret image region to implement priority share distribution, and dividing key shares into multiple backup shares at a higher granularity, so that high-priority shares are robust, each share only recovers corresponding encrypted information and does not leak other data, and can resist collusion attacks; and (2) during the QR code embedding process, the pixel unit module is expanded to a 5×5 pixel matrix, and the inner matrix pixel depth of the pattern recognition unit is adjusted by an adaptive pixel depth adjustment algorithm, so that the QR code has high security and robustness without affecting its original functionality, and also increases the clarity of the secret image and the information payload of the QR code.

After this introductory section, the rest of the paper is organized as follows. Section 2 introduces the background knowledge. Section 3 presents the overall scheme design. Section 4 analyzes the security of the proposed scheme. Section 5 provides an analysis and discussion of the experimental results. Finally, the advantages and limitations of this paper are summarized, and future work is discussed.

2. Preliminaries

2.1. Two-level QR code based on module identification unit

Two-level QR code usually refers to a technology that combines a QR code with secret images or information. The application scenarios of the multi-level QR code are relatively wide-ranging. For example, in the fields of electronic ticketing, mobile payment, and logistics distribution, QR codes can be used as either an order number or logistics number, and then embed multiple sub-information inside, each of which corresponds to a specific ticket, product, or goods information. In this way, users can obtain all information with just one scan of the QR code, effectively improving the efficiency and convenience of information transmission.

Two-level QR codes based on a module identification unit is a high-quality visual QR code, while ensuring that it remains readable. By subdividing one pixel in a QR code into a 3×3 sub-module and binding the pixel value to the fixed central sub-module, the black and white pixel distribution of the information storage structure (i.e., the remaining 8 sub-modules) can be controlled without changing the existing QR code compilation rules. With this flexibility, a set of binary byte streams can be used to fill the outer matrix area of the modules. The resulting image can still be extracted and read by QR code decoding tools. This structure is called the module identification unit, as shown in Figure 1.

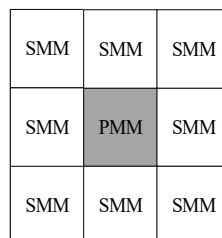


Figure 1. QR code module recognition unit.

The image to be embedded is divided into 3×3 pixel sub-regions, as shown in the above figure, and each sub-region is embedded into the corresponding sub-module of the QR code. When the pixel value of the center area (PMM area) of the module identification unit is consistent with the pixel value of the same position in the carrier QR code, the original carrier information of the QR code can be accurately extracted and interpreted by the machine. Therefore, the original public-level carrier data information of the QR code is stored in the center PMM pixel area of the module identification unit. The remaining eight pixels store secret-level carrier data information in sequence from top to bottom and from left to right, stored in the SMM area of the module identification unit, which is a set of row vectors in the sharing matrix M .

During the image embedding process, the pixel size of the embedded image and the QR code is $m \times n$. Let the secret module $S(i, j)$ complete the secret sharing process using the sharing matrix .

At this time, the module identification unit of the shared QR code $T_i(1 \leq i \leq n)$ and the row vector of the i -th row of matrix M satisfies formula (1):

$$\begin{bmatrix} T_k(i-1, j-1) & T_k(i, j-1) & T_k(i+1, j-1) \\ T_k(i-1, j) & T_k(i, j) & T_k(i+1, j) \\ T_k(i-1, j+1) & T_k(i, j+1) & T_k(i+1, j+1) \end{bmatrix} = \begin{bmatrix} M(i, 1) & M(i, 4) & M(i, 6) \\ M(i, 2) & * & M(i, 7) \\ M(i, 3) & M(i, 5) & M(i, 8) \end{bmatrix} \quad (1)$$

2.2. Block-based region-wise progressive visual cryptography algorithm

The traditional image progressive restoration algorithm achieves enhancement of black and white pixel contrast by constantly increasing it. As the contrast reaches a certain level, even the results restored with a portion of the shares can leak the overall information, making the secret image susceptible to collusion attacks. However, the region progressive recovery algorithm can mitigate this issue. Nonetheless, due to the lack of robustness in the recovery process, the information corresponding to a lost or damaged share cannot be restored, thereby compromising the integrity of the secret image recovery.

The design approach in reference [17] implements a region progressive image recovery algorithm based on the principle of block construction. By calculating the number of authorized subsets, the secret image is divided into several shares, each corresponding to different encryption areas, and the regions are constructed independently. The design utilizes an encryption matrix based on the (n, n) XOR visual secret sharing algorithm to achieve the scheme of encryption area recovery through XORing the shares. The specific steps are as follows:

(1) Secret sharing algorithm

Sequentially traverse each authorized subset $Q = \{Q_1, Q_2, \dots, Q_n\}$ and combine with the set of encryption matrices using the XVCS algorithm. When sharing black/white pixels, extract the random vector matrix of the corresponding matrix $C_0^{(q,q)} / C_1^{(q,q)}$. Participant $i(i \in Q)$ fills in the corresponding black/white pixels at the corresponding position of block T_i^Q in the shared image T_i using the pixel values from the passed matrix. For participants not in the authorized subset Q , random black/white pixels are filled in the corresponding positions of their shares, and each area of the share is constructed in turn. The algorithm process is shown in Algorithm 1.

Algorithm 1. Secret sharing algorithm

Input: Threshold structure (k,n) and secret image S .

Output: Shared images T_i .

Step1: Select an authorized subset $Q_{\min} = \{i_1, i_2, \dots, i_{q_{\min}}\}$ form Γ_Q , the selected authorized subset is a subset with fewer participants, and construct the set of (q_{\min}, q_{\min}) -XVCS encryption matrices $C_0^{(q_{\min}, q_{\min})} / C_1^{(q_{\min}, q_{\min})}$;

Step2: Iterate over each pixel in the area $R_1, R_2, \dots, R_{q_{\min}-k+1}$ in turn. If the pixel is white, jump to step 3; otherwise, execute step 4;

Step3: Select a random vector matrix from the matrix set $C_0^{(q_{\min}, q_{\min})}$;

Step4: Select a random vector matrix from the matrix set $C_1^{(q_{\min}, q_{\min})}$;

Step5: According to the selected vector matrix, fill the corresponding positions in the $T_i^{Q_{\min}}$ part of participant i 's corresponding shared image with black/white pixels. For the remaining

subsets of participants in $\{P-Q_{\min}\}$ fill their corresponding shared images with random black/white pixels;

Step6: Repeat step 2 to step 3 for each pixel in area $R_1, R_2, \dots, R_{q_{\min}-k+1}$ to encrypt it, and complete the assignment of the authorization subset Q_{\min} ;

Step7: Traverse the next authorized subset and perform Steps 1 to 6 sequentially until the remaining authorized subsets are completed for the assignment operation.

Step8: Concatenate the parts $T_i = T_i^{Q_1} + T_i^{Q_2} + \dots + T_i^{Q_d}$, output the shared image T_i , the algorithm ends.

(2) Secret recovery algorithm

For the authorization set $Q = \{i_1, i_2, \dots, i_q\}$, only executing $R(Q) = T_{i_1}^Q \otimes T_{i_2}^Q \otimes \dots \otimes T_{i_q}^Q$ on the shared images corresponding to the participants in the set Q can reveal the corresponding hidden image information in the recovered image $R(Q)$.

This algorithm realizes the process of secret distribution and recovery in the region-incremental visual secret sharing scheme. This paper adopts the region matrix construction idea from this algorithm and simplifies it by removing redundant elements from the original matrix $C_0^{(q,q)} / C_1^{(q,q)}$. Furthermore, it refines the partitioned image regions R_i to enhance the robustness of the region-incremental visual secret sharing algorithm.

3. The proposed methods

This paper presents a two-level QR code scheme for image secret sharing based on the region matrix. Based on the region matrix, a new progressive visual secret sharing algorithm is proposed, based on the region matrix, to simplify the construction process and eliminate the element redundancy of the matrix $C_0^{(q,q)} / C_1^{(q,q)}$. The image is divided into regions using the constructed second-order scheduling matrix for image segmentation, and different weights are allocated to each share, enabling participants to have varying priorities. During secret image recovery, participants with different priorities restore different amounts of secret information, with higher-priority participants recovering more secret information. As high-priority shares contain critical data, to ensure the security of their data, a shared scheduling matrix is used to further divide the critical shares into backup shares. In case high-priority critical shares are either lost or subjected to irrecoverable attacks, the backup shares can substitute and participate in the recovery of the original image, making the regionally incremental visual secret sharing algorithm more robust.

In the process of embedding QR codes into shared images, in order to increase clarity and payload, the pixel unit module is expanded to a 5×5 pixel matrix. An adaptive pixel depth adjustment algorithm is employed to adjust the inner matrix pixel depth of pattern recognition units, while preserving the original functionality of the QR codes. The scheme is illustrated in Figure 2. This approach maintains high security and robustness while ensuring superior image clarity after embedding QR codes. Experimental results demonstrate that the scheme can achieve a two-level QR code scheme through progressive weight allocation for image recovery, preventing information leakage from the carrier QR codes and providing a certain level of robustness.

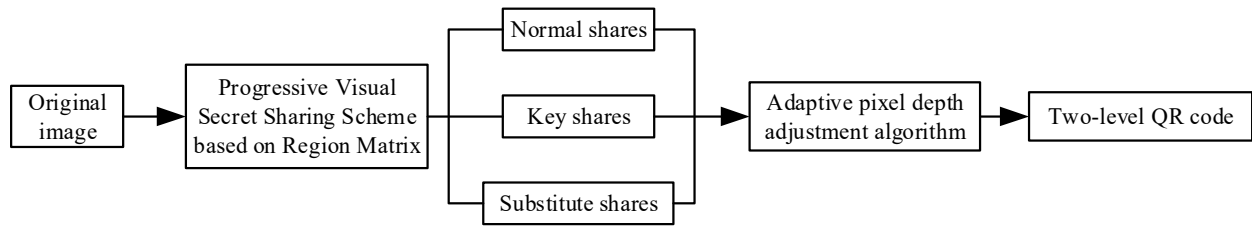


Figure 2. Flowchart of two-level QR code based on region-progressive secret sharing algorithm.

3.1. Progressive image secret sharing algorithm based on region matrix

This section presents a hierarchical progressive image secret sharing algorithm based on region matrices. This algorithm is founded on the secret sharing scheme without pixel expansion. By dividing the original image into non-overlapping regions, the selected vector values are allocated to all participants using the corresponding second-order scheduling matrix during each sharing process. Each participant realizes different priority divisions based on the set of scheduling matrices for the respective regions. The method of selecting random vectors using scheduling matrices maintains the randomness of the generated shares, ensuring that the pixel arrangement of each share is irregular, thereby achieving the goal of preventing individual share leaks of secrets. To recover all secret information, it is necessary to combine the shares provided by each participant.

By specifying the scheduling matrix set for the key area corresponding to the key participants, they can use the key vectors with a high frequency to distribute more key pixels to the key participants. The key participant holds a key share that carries a significantly higher amount of secret data than the normal participants' shares, and is referred to as the normal share. The key share holds more secret data, and if it is subject to high-degree geometric attacks, it may result in either a loss or inability to identify the recovery results. Therefore, by dividing the corresponding key share region into finer granularity, its scheduling matrix portion is shared with the substitute shares. The substitute shares hold the scheduling matrix shared by the key share and the unique scheduling matrix, thereby achieving a fine-grained distribution of the key region.

During the image recovery process, a single substitute share participating with normal shares will not recover any critical information. The critical data will be recovered only when all substitute shares participate together. To ensure the high priority of the key share, the number of shared schedules is limited, which ensures that the clarity of the image recovered by all substitute shares is still lower than that of the key share.

(1) Preliminary work

In the progressive image secret sharing algorithm based on region matrices, an original secret image P is first divided into n arbitrarily sized and non-overlapping image regions P_1, P_2, \dots, P_n , satisfying the definition of formula (2):

$$\begin{cases} P = \cup P_i \\ P_i \cap P_j = \emptyset, 1 \leq i \neq j \leq n \end{cases} \quad (2)$$

The image partition satisfies $KP + NP = P$, and the critical region KP is composed of k substitute regions RP , satisfying $KP = RP_1 + RP_2 + \dots + RP_k$ ($k < n - i - x$). The region relationships are shown in Figure 3, and when partitioning RP , it is necessary to adhere to the specified rules: evenly divide KP

into smaller granular sections, and ensure that the sections of the RP do not overlap with each other.

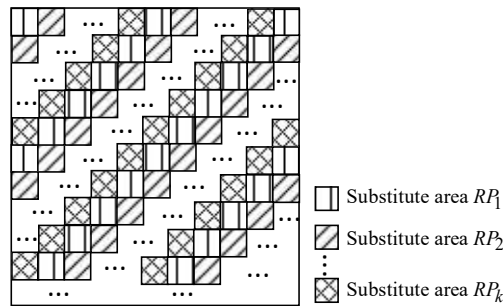


Figure 3. Area granularization.

Construct $n+1$ scheduling matrices of dimension $2 \times n$, as shown in formula (3). The matrix C_0 is responsible for the secret sharing of white pixels in the original image. Its first row consists of all 0s, and the second row consists of all 1s. The matrix C_m is responsible for the secret sharing of black pixels in the m -th region of the original image, where $1 < m < n$. In the first row of C_m , the elements with j coordinates equal to m are set to 1, and in the second row, the elements with j coordinates equal to m are also set to 1, while the rest of the elements are all set to 0. μ_{ij} and ν_{ij} are the corresponding position elements in the scheduling matrix.

$$C_0 = [\mu_{ij}] = \begin{cases} 0, & \text{if } i=1, 1 \leq j \leq n \\ 1, & \text{if } i=2, 1 \leq j \leq n \end{cases}$$

$$C_m = [\nu_{ij}] = \begin{cases} 1, & \text{if } i=1, 1 \leq j = m \leq n \\ 1, & \text{if } i=2, 1 \leq j \neq m \leq n, \text{ where } m \in [1, n] \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

After constructing the scheduling matrix, each region corresponds to its respective scheduling matrix. The partitioning of shared images based on priority is achieved by selectively choosing the row vectors of the scheduling matrix for each region. The key shares and substitute shares share a portion of the scheduling matrix. The matrix sharing rule satisfies Eq (4). Further allocation is performed based on the set of region matrices and the algorithm.

$$NC = \{C_1, C_2, \dots, C_i\}$$

$$KC = \{C_{i+1}, C_{i+2}, \dots, C_j\}, 1 \leq i \leq j < n, 1 < x < j-1 \quad (4)$$

$$RC = \{C_{i+x}, \dots, C_j, \dots, C_n\}$$

The key shares correspond to matrix KC, and the substitute shares correspond to matrix RC, satisfying $KC \cap RC = \{C_{i+x}, C_{i+x+1}, \dots, C_j\}$, indicating that KC and RC share a portion of the scheduling matrix. The key shares hold the majority of the scheduling matrices, while the substitute shares hold a minority of the scheduling matrices. Furthermore, the key shares' regions are subsequently granulated, and the substitute shares are allocated using the corresponding scheduling matrix RC of the RP.

(2) Secret sharing phase

During the secret sharing generation phase, in the process of pixel allocation, a random number k is used to randomly select a row vector from the scheduling matrices C_0 and C_f . Then, each value of the row vector is assigned to the corresponding share (i.e., the i -th vector value is assigned to the i -th share. Therefore, each shared share will have the same size as the secret image. If the pixel of the secret image is white, any row vector from the scheduling matrix C_0 is selected as the distribution content; if the pixel is black, as long as the pixel belongs to region P_f , a row vector is selected from the scheduling matrix C_f . The same scheduling method is used until all secret pixels are processed. The secret sharing algorithm is shown in Algorithm 2.

Algorithm 2. Secret sharing algorithm

Input: Parameter $f \in [1, n]$, original image $P(h \times w)$, subarea of the original image P_f , among them $P = P_1 + P_2 + \dots + P_n$, each area corresponds to a scheduling matrix C_1, C_2, \dots, C_n of size $2 \times n$ for black pixels and a scheduling matrix C_0 of size $2 \times n$ for white pixels; The number of participants is $t (t < n)$, the scheduling matrix parameters are l, q, m , satisfying $t = l + q, n = l + q + m$

Output: Shared images $AC = \{NC, KC, RC\}$, normal shares $NC = \{NC_1, NC_2, \dots, NC_t\}$, key shares $KC = \{KC_1, KC_2, \dots, KC_q\}$, substitute shares $RC = \{RC_1, RC_2, \dots, RC_m\}$;

Step1: Let $i = j = 1$, where i represents the row parameter of the original image, and j represents the column parameter of the original image;

Step2: Let the parameter $k \in [0, 1]$ take random values set the shared share identification parameter $count = 1$;

Step3: Determine the pixel region flag f of pixel point $Point(i, j) \in P_f$, if $C_f \in NS$, jump to step 4; otherwise, jump to step 5;

Step4: If $Pixel(i, j) = 0$, set $AC_{count} = C_0(k, count)$; If $Pixel(i, j) = 1$, set $AC_{count} = C_f(k, count)$, $count = count + 1$ and repeat this step until $count > t$ jump to step 12;

Step5: If $C_f \in KS$ and $C_f \notin RS$, jump to step 6; otherwise, jump to step 7;

Step6: If $Pixel(i, j) = 0$, set $AC_{count} = C_0(k, count)$; If $Pixel(i, j) = 1$, set $KC_{count} = NC_{count} = C_f(k, count)$, $RC_{count} = C_0(k, count)$, and repeat this step $count = count + 1$ until $count > t$ jump to step 12;

Step7: If $C_f \notin KS$ and $C_f \in RS$, jump to step 8; otherwise, jump to step 10;

Step8: If $Pixel(i, j) = 0$, set $AC_{count} = C_0(k, count)$, and repeat this step $count = count + 1$ until $count > t$ return to step 12; if $Pixel(i, j) = 1$, execute step 9;

Step9: If $Point(i, j) \in RP_x$, when $count = x$ set $RC_{count} = NC_{count} = C_f(k, count)$, $KC_{count} = C_0(k, count)$ else $NC_{count} = C_f(k, count)$, $RC_{count} = KC_{count} = C_0(k, count)$, set $count = count + 1$, and repeat this step until $count > n$, jump to step 12;

Step10: If $Pixel(i, j) = 0$, set $AC_{count} = C_0(k, count)$, set $count = count + 1$, and repeat this step until $count > n$, jump to step 12; if $Pixel(i, j) = 1$, execute step 11;

Step11: If $Point(i, j) \in RP_x$, when $count = x$ set $AC_{count} = C_f(k, count)$ else $KC_{count} = NC_{count} = C_f(k, count)$, $RC_{count} = C_0(k, count)$; set $count = count + 1$, and repeat this step until $count > t$ execute step 12;

Step12: Set $j = j + 1$, if $j > h$, execute step 13, otherwise, return to step 2;

Step13: Set $j = j + 1$, if $j > h$, execute step 14, otherwise, return to step 2;

Step14: Embed the set of shared images AC into the hierarchical QR codes, the algorithm ends.

(3) Secret reconstruction phase

Each participant who holds a certain image region of the original image has its own secret critical

image area. That is, as long as the shared secret of the i -th participant is superimposed, the corresponding region's secret information can be restored, and the content of the corresponding region will be recovered, displaying some secret information. When the key shares participate in the recovery, more data can be recovered. The recovery process of the original image P follows formula (5):

$$P = NC_1 \otimes \dots \otimes NC_1 \otimes KC_1 \otimes \dots \otimes KC_q \quad (5)$$

When any t number of shares are superimposed, the corresponding encrypted area of those t participants will be recovered, while the remaining areas will remain in a noisy state. When the key shares are attacked and cannot be recovered, the critical data can be gradually restored using substitute shares; if only a single or fewer than m substitute shares are involved, the recovery of that specific region cannot be achieved. The original image P is restored through the joint participation of normal, key, and substitute shares according to the given formula (6):

$$P = NC_1 \otimes \dots \otimes NC_1 \otimes KC_1 \otimes \dots \otimes KC_q \otimes RC_1 \otimes \dots \otimes RC_m \quad (6)$$

All substitute shares participating in the recovery have the same clarity as the critical shares. If only a part of the substitute shares participates in the recovery, the original information cannot be restored to prevent information leakage caused by a small number of substitute shares.

3.2. Design of adaptive pixel depth adjustment algorithm

In this scheme, the 3×3 unit pattern substitution algorithm was first extended and improved. By expanding a single pixel of the QR code into a 5×5 pixel block, the information payload can be increased by about three times the original, which supports embedding secret sharing shares of higher pixel images and results in a clearer recovered secret image. However, directly embedding the secret information into the QR code pixels that have been expanded to 5×5 pixel blocks will cause the device to automatically ignore the center code point of the pixel block during reading, resulting in unstable QR code reading results, where the outer pixels may be recognized as the center code point. Although the method of taking the mean value is feasible to a certain extent, when the SMM area of the unit module is all black and the PMM area is all white, or when the SMM area is all white and the PMM area is all black, the inner matrix pixels do not change much, making the QR code still unreadable.

In this paper, the adaptive pixel depth adjustment algorithm is proposed to address the above-mentioned issues. The algorithm adjusts the eight pixels in the inner matrix of the 5×5 unit module to reduce the contrast between the embedded outer pixels and the original central PMM module pixels, without affecting the decoding correctness of the module. Figure 4 shows two extreme examples, where the pixel difference between the central code point and the outer matrix is highest. The result of the adaptive adjustment of a single pixel module is also shown in the figure.

When the central pixel value is opposite to that of the outer pixels, the adjustment algorithm can reduce the contrast between the inner and outer pixels to a certain extent, ensuring that the module is not identified incorrectly. At the same time, since this algorithm has limited impact on the embedded secret information, the recovered secret image from multiple overlapping QR codes is clearer than that from the original 3×3 unit pattern replacement algorithm. The pixel division is shown in Figure 5.

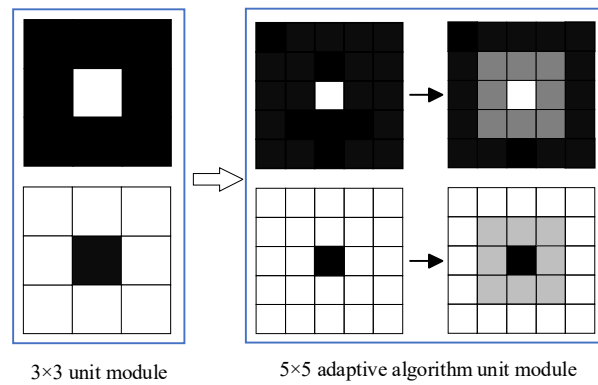


Figure 4. Example of unit module replacement.

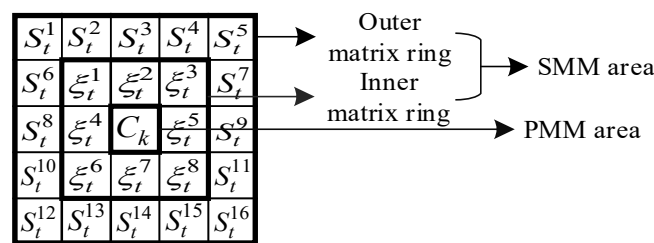


Figure 5. Diagram of a 5×5 unit module matrix.

The 5×5 unit replacement algorithm based on the adaptive pixel depth adjustment algorithm adopts the method of reducing the difference between the key scanning code points of the QR code and the outer matrix pixels, and embeds the shared image into the carrier QR code. The two-level QR code generation process is shown in Figure 6.

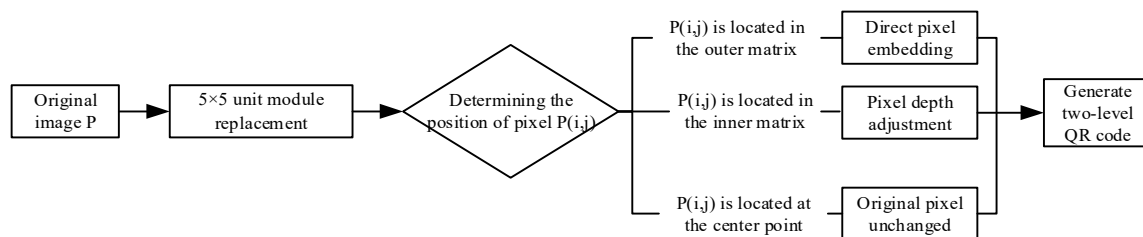


Figure 6. The construction process of a third-level QR code.

In the outer matrix of the 5×5 unit module, directly fill in the shared pixel values $S_t^n(i, j), 1 \leq n \leq 16$ at the corresponding positions of the QR code. In the inner matrix of the unit module, fill in the adjustment variable $\xi_t^n(i, j), 1 \leq n \leq 8$ that is equivalent to the pixel at the corresponding position of the shared portion, as shown in formula (7). The value of the inner matrix is adjusted based on the average of the adjacent three outer matrix values, that is, the variable can automatically adjust the pixel at that position based on the surrounding pixel values, achieving an adaptive adjustment of the inner matrix values and reducing the impact on the original image:

$$\begin{aligned}
\xi_t^1(i, j) &= \frac{1}{2} \left[\frac{S_t^1(i, j) + S_t^2(i, j) + S_t^6(i, j)}{3} - C_k(i, j) \right] \bmod 255 \\
\xi_t^2(i, j) &= \frac{1}{2} \left[\frac{S_t^2(i, j) + S_t^3(i, j) + S_t^4(i, j)}{3} - C_k(i, j) \right] \bmod 255 \\
\xi_t^3(i, j) &= \frac{1}{2} \left[\frac{S_t^4(i, j) + S_t^5(i, j) + S_t^7(i, j)}{3} - C_k(i, j) \right] \bmod 255 \\
\xi_t^4(i, j) &= \frac{1}{2} \left[\frac{S_t^6(i, j) + S_t^8(i, j) + S_t^{10}(i, j)}{3} - C_k(i, j) \right] \bmod 255 \\
\xi_t^5(i, j) &= \frac{1}{2} \left[\frac{S_t^7(i, j) + S_t^9(i, j) + S_t^{11}(i, j)}{3} - C_k(i, j) \right] \bmod 255 \\
\xi_t^6(i, j) &= \frac{1}{2} \left[\frac{S_t^{10}(i, j) + S_t^{12}(i, j) + S_t^{13}(i, j)}{3} - C_k(i, j) \right] \bmod 255 \\
\xi_t^7(i, j) &= \frac{1}{2} \left[\frac{S_t^{13}(i, j) + S_t^{14}(i, j) + S_t^{15}(i, j)}{3} - C_k(i, j) \right] \bmod 255 \\
\xi_t^8(i, j) &= \frac{1}{2} \left[\frac{S_t^{15}(i, j) + S_t^{16}(i, j) + S_t^{11}(i, j)}{3} - C_k(i, j) \right] \bmod 255
\end{aligned} \tag{7}$$

where pixel $C_k(h, w)$ represents the pixel value of the central module in the k-th unit module of the QR code, and pixel $S_t^n(i, j), 1 \leq n \leq 16$ represents the corresponding pixel value in the outer matrix of the t-th shared portion.

During the image embedding process, the module to be embedded $S(i, j)$ uses the scheduling matrix $T_k(i, j)$ to complete the secret sharing process. At this time, the module identification unit of the carrier QR code $T_k(1 \leq k \leq (m \times n) / 25)$ and the unit matrix $S_t(i, j)$ at the corresponding position of the matrix to be embedded satisfy formula (8):

$$\begin{aligned}
& \begin{bmatrix} T_k(i-2, j-2) & T_k(i-1, j-1) & T_k(i, j-2) & T_k(i+1, j-2) & T_k(i+2, j-2) \\ T_k(i-2, j-1) & T_k(i-1, j-1) & T_k(i, j-1) & T_k(i+1, j-1) & T_k(i+2, j-1) \\ T_k(i-2, j) & T_k(i-1, j) & T_k(i, j) & T_k(i+1, j) & T_k(i+2, j) \\ T_k(i-2, j+1) & T_k(i-1, j+1) & T_k(i, j+1) & T_k(i+1, j+1) & T_k(i+2, j+1) \\ T_k(i-2, j+2) & T_k(i-1, j+2) & T_k(i, j+2) & T_k(i+1, j+2) & T_k(i+2, j+2) \end{bmatrix} \\
& = \begin{bmatrix} S_t^1(i, j) & S_t^2(i, j) & S_t^3(i, j) & S_t^4(i, j) & S_t^5(i, j) \\ S_t^6(i, j) & \xi_t^1(i, j) & \xi_t^2(i, j) & \xi_t^3(i, j) & S_t^7(i, j) \\ S_t^8(i, j) & \xi_t^4(i, j) & C_k(i, j) & \xi_t^5(i, j) & S_t^9(i, j) \\ S_t^{10}(i, j) & \xi_t^6(i, j) & \xi_t^7(i, j) & \xi_t^8(i, j) & S_t^{11}(i, j) \\ S_t^{12}(i, j) & S_t^{13}(i, j) & S_t^{14}(i, j) & S_t^{15}(i, j) & S_t^{16}(i, j) \end{bmatrix}
\end{aligned} \tag{8}$$

Among them $S_t^n(i, j), 1 \leq n \leq 16$ refers to the corresponding position elements of the t-th shared block, $\xi_t^n(i, j), 1 \leq n \leq 8$ refers to the adjustment variable for the pixels at the corresponding position as the shared block, and $C_k(i, j)$ refers to the center code element in the k-th unit module of the carrier QR code.

When embedding the shared set AC into the carrier QR code, the matrices in the set are embedded

into the same position of the corresponding carrier QR code to ensure correctness during recovery, while avoiding the central key code point area. In the process of secret image recovery, the pixel positions in the third-level QR code correspond to the pixel values at the same position in the original image. When participating in the superposition of shares, the pixel values at the same position are calculated according to the matrix allocation vector value. The QR code embedding process based on the 5×5 adaptive pixel depth adjustment algorithm is shown in Algorithm 3.

Algorithm 3. Adaptive pixel depth adjustment algorithm based on 5×5 unit pattern

Input: Shared images AC, n carrier QR codes $C(h \times w)$

Output: The n two-level QR codes T for embedding the secret image.

Step1: Divide the shared image S into sub-images of size 5×5 S' ;

Step2: Let $i = j = 3$, where i represents the row of the carrier QR code, and j represents the column of the carrier QR code.

Step3: If the current module $k(1 \leq k \leq (m \times n) / 25)$ is a functional area, execute formula (9):

$$\begin{aligned}
 & \begin{bmatrix} T_k(i-2, j-2) & T_k(i-1, j-1) & T_k(i, j-2) & T_k(i+1, j-2) & T_k(i+2, j-2) \\ T_k(i-2, j-1) & T_k(i-1, j-1) & T_k(i, j-1) & T_k(i+1, j-1) & T_k(i+2, j-1) \\ T_k(i-2, j) & T_k(i-1, j) & T_k(i, j) & T_k(i+1, j) & T_k(i+2, j) \\ T_k(i-2, j+1) & T_k(i-1, j+1) & T_k(i, j+1) & T_k(i+1, j+1) & T_k(i+2, j+1) \\ T_k(i-2, j+2) & T_k(i-1, j+2) & T_k(i, j+2) & T_k(i+1, j+2) & T_k(i+2, j+2) \end{bmatrix} \\
 & = \begin{bmatrix} C_k(i, j) & \dots & C_k(i, j) \\ \vdots & \ddots & \vdots \\ C_k(i, j) & \dots & C_k(i, j) \end{bmatrix} \tag{9}
 \end{aligned}$$

Otherwise, proceed to the next step.

Step4: For any arbitrary $k(1 \leq k \leq 25)$, select $S_k^{n'}$, $1 \leq n \leq 16$ form S' , Extract S'_i form $S_k(i, j)$ excluding the central element, which consists of the remaining 24 pixel values. Adjust the pixel values of the inner matrix to convert them to $\xi(i, j)$ and fill them according to formula (10):

$$\begin{aligned}
 & \begin{bmatrix} T_k(5i-4, 5j-4) & T_k(5i-3, 5j-4) & T_k(5i-2, 5j-4) & T_k(5i-1, 5j-4) & T_k(5i, 5j-4) \\ T_k(5i-4, 5j-3) & T_k(5i-3, 5j-3) & T_k(5i-2, 5j-3) & T_k(5i-1, 5j-3) & T_k(5i, 5j-3) \\ T_k(5i-4, 5j-2) & T_k(5i-3, 5j-2) & T_k(5i-2, 5j-2) & T_k(5i-1, 5j-2) & T_k(5i, 5j-2) \\ T_k(5i-4, 5j-1) & T_k(5i-3, 5j-1) & T_k(5i-2, 5j-1) & T_k(5i-1, 5j-1) & T_k(5i, 5j-1) \\ T_k(5i-4, 5j) & T_k(5i-3, 5j) & T_k(5i-2, 5j) & T_k(5i-1, 5j) & T_k(5i, 5j) \end{bmatrix} \\
 & = \begin{bmatrix} S_i^{1'}(i, j) & S_i^{2'}(i, j) & S_i^{3'}(i, j) & S_i^{4'}(i, j) & S_i^{5'}(i, j) \\ S_i^{6'}(i, j) & \xi_i^1(i, j) & \xi_i^2(i, j) & \xi_i^3(i, j) & S_i^{7'}(i, j) \\ S_i^{8'}(i, j) & \xi_i^4(i, j) & C_k(i, j) & \xi_i^5(i, j) & S_i^{9'}(i, j) \\ S_i^{10'}(i, j) & \xi_i^6(i, j) & \xi_i^7(i, j) & \xi_i^8(i, j) & S_i^{11'}(i, j) \\ S_i^{12'}(i, j) & S_i^{13'}(i, j) & S_i^{14'}(i, j) & S_i^{15'}(i, j) & S_i^{16'}(i, j) \end{bmatrix} \tag{10}
 \end{aligned}$$

Step5: Let $i = j + 5$, if $j > w$, jump to step 6; otherwise, execute step 4;

Step6: Let $j = 3$, $i = i + 5$, if $i > h$, jump to step 7; otherwise, execute step 4;

Step7: The shares are successfully embedded into the carrier QR codes, the algorithm concludes.

By employing the above algorithm to embed the shared set AC into the corresponding carrier QR code as the two-level information of the QR code, the QR code is graded and designed to carry confidential information that other users cannot decipher. This approach overcomes the threat of information leakage resulting from the public nature of the QR code.

4. Algorithm analysis and proof

4.1. Analysis of shared transparency rate

Using each pixel of the distributed shares as a basic unit, each basic unit assigns black/white pixel values according to the rules of the scheduling matrix and a randomly selected unit vector. Additionally, the value of the pixel is random due to the random value of k . Table 1 provides a transmittance example of a single pixel for a (n,n) progressive image secret sharing algorithm based on matrix regions.

Table 1. An example of the transmittance of a single pixel in the (n,n) progressive image secret sharing algorithm based on matrix regions.

Original pixel	Share 1	Share 2	Share n	Probability	Superimposed result
White pixel				1/2	
				1/2	
Black pixel				1/2n	
				1/2n	
					
					1/2	

Each white pixel in a single share is encrypted by a unit vector randomly selected from the scheduling matrix C_0 , resulting in a 50% probability of being encrypted as a black or white pixel with a 50% transparency. Each black pixel is encrypted by a unit vector randomly selected from the scheduling matrix C_f , resulting in a probability of $P_1 = (2n^2 + 1) / 4n^2$ of being encrypted as a black pixel and a probability of $P_0 = 1 - P_1$ of being encrypted as a white pixel. As the number of n shares n increases, the transparency of pixel x is given by Eq (11).

$$pr[AC(i, j)] = [(2n^2 + 1) / 4n^2 + 1 / 2] / 2 = (4n^2 + 1) / 8n^2 \quad (11)$$

For each column of each scheduling matrix, the number of black and white pixels is equal. The content being shared is randomly selected from the sharing matrix, so each corresponding shared pixel has a 50% chance of being black regardless of whether the secret pixel is white or black. Therefore, the black pixels are evenly dispersed across the shares, resulting in an average pixel transparency of $pr = 50\%$ for a single share AC_k . Each pixel in a single share has an equal probability of being encrypted as white or black, and the visual identification result of the entire share is a noise image with randomly arranged black and white pixels.

However, when two shared images are overlapped, according to the scheme shown in Table 1, the

black and white pixels at the same position in the white area remain unchanged after overlapping, which means that the average transparency of the white area remains 50%. For the black area, the encrypted area of a share will reduce the transparency to 0 in the corresponding area when the pixel value is black. When other shares are overlapped, the transparency of the corresponding encrypted area remains 50%. The hidden information can be restored by changing the transparency of the encrypted area without revealing the data of other shares.

4.2. Proof of leakage prevention in non-decryptable areas

When multiple shared images AC are overlapped, the resulting overlap is denoted as R, and its pixel values are given by Eq (12):

$$Pixel[R(i, j)] = \sum_{k=1}^n Pixel[AC_k(i, j)] \quad (12)$$

Since the white area pixels are randomly selected row vectors from the scheduling matrix C_0 and each value of the row vector is assigned to the corresponding share, the white area overlap result remains unchanged, and the average transparency remains 50%. According to the distribution of pixels in the black area, pixel values are selected from randomly selected row vectors in scheduling matrix C_f , and the transparency of the encrypted area corresponding to share AC_f will be reduced to 0 after overlapping due to the participation of black pixels in the specified position. The transparency of the encrypted area not involved in that share remains 50%. The overlapping result of the encrypted area will be affected by the critical pixels in C_f . Therefore, the transparency of the black and white areas in the decrypted area in the overlapped image is shown by Eq (13), and the pixel contrast of the decrypted area is shown by Eq (14).

$$L[R_{AC_1, AC_2, \dots, AC_n}(P_0)] = \frac{1}{2}, L[R_{AC_1, AC_2, \dots, AC_n}(P_1)] = 0 \quad (13)$$

$$\xi = \frac{L[R(AC_{pixel_0})] - L[R(AC_{pixel_1})]}{1 + L[R(AC_{pixel_1})]} = \frac{1}{2} \quad (14)$$

For the non-encrypted area, as the overlapping process is not affected by the corresponding key pixels in C_f , the transmittance is shown in Eq (15). Therefore, the contrast of the non-decryption area in the overlapped image is the result of Eq (16).

$$L[R(AC_{pixel_0})] = L[R(AC_{pixel_1})] = \frac{1}{2} \quad (15)$$

$$\xi = \frac{L[R(AC_{pixel_0})] - L[R(AC_{pixel_1})]}{1 + L[R(AC_{pixel_1})]} = \frac{1/2 - 1/2}{1 + 1/2} = 0 \quad (16)$$

The above formulas show that the contrast of the black and white regions in the decrypted area significantly changes after superimposition, while the contrast of the black and white regions in the non-decrypted area remains unchanged. When the shared parts participate in decryption, only the contrast in the decrypted area changes, while the non-decrypted area remains unchanged. Therefore, without sufficient participants, it is impossible to recover the corresponding area, effectively resisting

collusion attacks by participants.

4.3. Proof of leakage prevention in share

Due to the fact that white pixels on the original image are displayed as random black/white pixels at the same positions in the shares, and the pixels at the corresponding positions in the shares with the same index are identical, when the i -th and j -th shares are superimposed, the colors of the regions corresponding to the white pixels in the image blocks P_i and P_j with indices i and j will remain unchanged. In contrast, complementary shares distribute black pixels to the specified share at the positions corresponding to the original image's black pixels. At the same positions, except for the specified share, the pixels are still randomly distributed as black or white, resulting in the encrypted area of this share in the superimposed image appearing completely black. This allows for the recovery of the secret information in image blocks P_i and P_j using contrast, and can be visually recognized by the human visual system.

(1) Proof of leakage prevention for partial shares

Each individual shared portion has an encrypted area size of $(h \times w) / n$. The probability of the generated shared portion decrypting to the original content is $P_1 = C_{h \times w}^{\frac{h \times w}{n}} / 2^{\frac{h \times w}{n}}$, while the probability of a single shared portion recovering all the secret information is $P = \prod_{1 \leq i \leq n} C_{\frac{h \times w \times (n-i+1)}{n}}^{\frac{h \times w}{n}} / 2^{\frac{h \times w}{n}}$. By choosing

an appropriate value for n , the probability of an individual shared portion recovering the secret information can be made to approach 0. For instance, in the proposed scheme with an image size of 200×200 divided into four encrypted regions, the recovery probability is close to 0.

Additionally, regardless of the pixel content in the original image, when superimposing images with non-corresponding encryption regions, they will be sent to each share following the same rule, resulting in an equal probability of black and white pixels. As a result, when superimposing these two shares, the number of black pixels in other non-corresponding encryption regions will not increase, and there will be no contrast difference, ensuring that the secret content in these regions remains unrevealed.

When any t ($2 < t \leq n$) shares are overlaid, since each share decrypts different image blocks, there are C_n^t possible overlay results, as defined by the above formula. As the number of overlaid shares increases, the decrypted area of the overlaid result will expand, gradually revealing the image. When the combination of participants is different, the recovered results will differ; however, the decrypted content only includes the specified decryption area of the share, and the secrets of other blocks will not be revealed.

(2) Proof of leakage prevention in important regions

In the scheme, when important shares are obtained, the crucial share region contains more secret information compared to other regions. However, since it lacks the shared information from other regions, the secrets in the other blocks will not be compromised, and therefore, the entire information cannot be obtained.

When the attacker manages to identify important shares regions, they will not be able to access the secrets in other blocks due to the lack of shared information from those regions. As a result, they will be unable to obtain the complete secret information. Additionally, if attackers target the important area for recovery, the number of participants involved in the attack must reach the threshold value to successfully recover the secret information within that region.

4.4. Correctness analysis

Since the contents of each row vector in the scheduling matrix C_0 are identical, when the secret pixel is white, the same color will be assigned to the corresponding position in each share. The probability of the color being black or white in that position is 50%. Therefore, no matter how many shares are stacked, the corresponding white areas on the stacked image will retain the same color, and the probability of retaining black or white pixels is 50%.

The t value of the row vector in the scheduling matrix C_t is different from other values, which means that if a black (white) pixel appears at a certain position in the t -th share, random white (black) pixels will appear at the same position in all other shares. This randomness ensures that the area will not leak secret information, and the t -th share will be in a complementary state with other shares. After stacking this share, the black pixel area of the corresponding area P_i in the scheduling matrix C_t will be completely restored with a transparency of zero. Once the t -th share is stacked onto other shares, the black part corresponding to the t -th image block in the original image will be completely black (100% black), and the contrast of the white part will be 50%. Therefore, the secret content of m image blocks can be intuitively restored. According to formula (17), the accuracy of the restored area is $\alpha = 75\%$, and the difference in the accuracy between black and white pixels can be recognized by the human eye as the contour of the hidden information. The scheduling matrix C_t plays a major role in controlling participant t to recover the corresponding image block t .

$$\alpha = \frac{\sum_i^{h_{P_i}} \sum_j^{w_{P_i}} \begin{cases} 1, B_1[i, j] = A_1[i, j] \\ 0, B_1[i, j] \neq A_1[i, j] \end{cases}}{h_{P_i} \times w_{P_i}} = \frac{Pixel_1 + \frac{1}{2} Pixel_0[P_i]}{Pixel_0[P_i]} \quad (17)$$

$$= \frac{\alpha_{Pixel_1[P_i]} + \alpha_{Pixel_0[P_i]}}{2} = \frac{3}{4}$$

If the t -th share is not available during the decryption process, the black pixels in image block m cannot be increased by other shares, which means that the corresponding part of the secret image block remains unchanged and still has a contrast of 50% black and 50% white. Additionally, this also demonstrates that the transparency of black and white pixels is the same, resulting in random black and white pixels that cannot be recognized by the human eye, and the secret information in image block m cannot be distinguished. In addition, based on the results of similarity experiments, the mean hash algorithm has a similarity of 0.69, the interpolated hash algorithm has a similarity of 0.59, and the perceptual hash algorithm has a similarity of 0.81. After removing the influence of the QR code, the results are consistent with the results of the formula proof. Therefore, this algorithm has a high recovery accuracy.

5. Experimental comparison and result analysis

This chapter first conducts experimental analysis on two aspects of the region matrix-based progressive secret sharing algorithm and the adaptive pixel contrast algorithm to demonstrate their good performance. Then, the feasibility and clarity of the overall scheme are experimentally validated and compared with results to show that the scheme has certain advantages and practicality.

5.1. Experimental results of the secret sharing algorithm

(1) Experimental results

The experimental results of the progressive image secret sharing algorithm based on the region matrix in Section 3.1 are shown in Figure 7. Taking the (2,3,3) structure as an example, the image is divided into three shares, including two normal shares and one key share. Two shares are required to recover partial information, and there are three substitute shares available for repairing the key share.

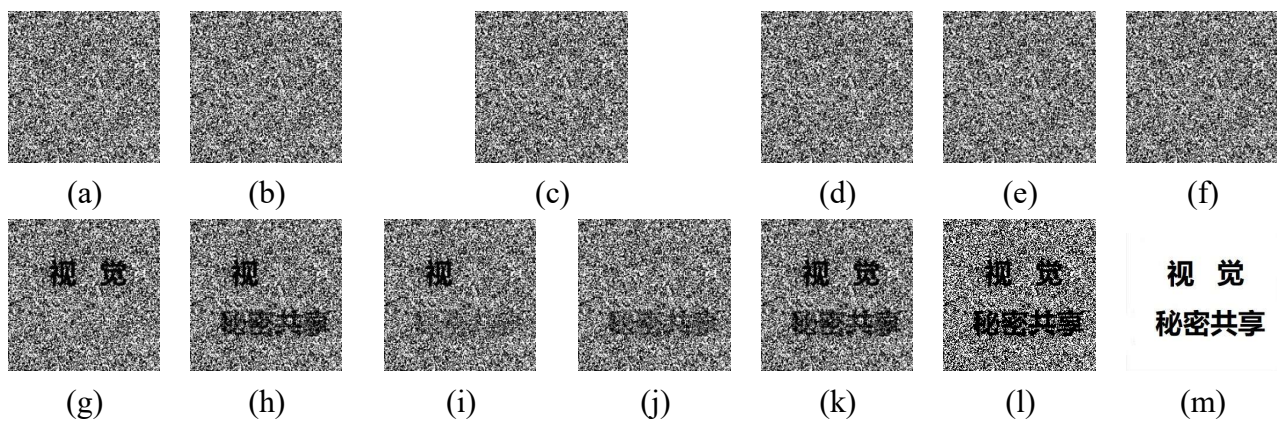


Figure 7. Shares and recovery results in (2,3,3) structure.

In Figure 7, images (a) and (b) are normal shares, image (c) is a key share, and images (d–f) are substitute shares. Image (m) is the original secret image. Image (g) is the result of overlaying normal shares (a) and (b), while image (h) is the result of overlaying normal share (a) and key share (c). It can be seen that overlaying normal shares only recovers some information in the corresponding area, while key shares carry more encrypted information, so the content recovered after overlaying them is significantly different from that of normal shares. Image (i) is the result of overlaying normal share (a) with a single substitute share (d), showing that a single substitute share is not sufficient to correctly recover the secret image. Image (j) is the result of overlaying multiple substitute shares (d–e) below the threshold, while image (k) is the result of overlaying all substitute shares and normal shares. Image (l) is the result of overlaying all shares (a–f).

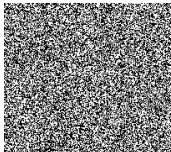


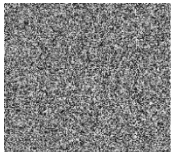


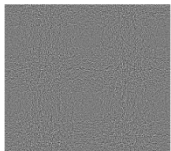


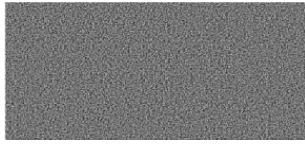


From the experimental results, it can be seen that when a single substitute share is involved in the recovery of the secret image, it cannot restore the information of the corresponding area due to the lower priority of the substitute share, so the encrypted information corresponding to the key share will not be leaked. As the number of substitute shares involved in the recovery increases, the encrypted information in the key area is gradually restored, but the clarity of the recovery results is still lower than when the key shares are involved in the recovery. Only when all substitute shares are involved together, can all the data of the secret image be restored clearly.

(2) Clarity comparison experiment

The comparison of recovery clarity between this scheme and other literature schemes is shown in Table 2. Reference [18] is a progressive image secret sharing scheme based on random grid. Reference [19] proposes a pixel not-expanding visual secret sharing scheme based on two-level threshold. Reference [20] designs a visual cryptography scheme using circular sharing. All these schemes improve the clarity of the recovered secret image. However, by comparing the recovery processes, it can be seen that as the number of shares involved increases, the contrast of black and

white pixels begins to decrease. When the secret image is completely restored, the result becomes darker, which reduces the contour recognition of the recovery result.

Table 2. Comparison of our scheme with visual sharing schemes.





References	Shared image	Partial shared images involved in recovery	Complete restoration
This scheme			
[18]			
[19]			
[20]			

The contrast of black and white pixels in the recovery result of the proposed scheme is independent of the number of shares involved in the recovery. Even if all shares participate in the recovery, the contrast of black and white pixels in the final recovery result still remains at a certain level, and the light transmittance of white pixels unrelated to the secret always remains at 50%. Therefore, the recovery result of this scheme has higher distinguishability.

5.2. QR code unit module replacement result

Based on the module recognition characteristics of QR codes, this section uses a 5×5 matrix as the module recognition unit. Compared to the original algorithm, the secondary information payload has been increased by three times, allowing QR codes to carry more data and the restored image contours to be clearer. This approach uses QR code version 5 with an error correction level of H as an example, but it can be applied to QR codes of different versions as well for image information embedding. The comparison results with other schemes are presented in Table 3.

Table 3. Experimental results comparison.

Image embedding scheme	Experimental results	Hash similarity	Readability of QR code
3×3 module substitution		63.3%	readable
Halftone Adaptive Processing as described in [21]		67.5%	readable
Direct embedding using 5×5 module substitution		79%	unreadable
Adaptive adjustment algorithm for 5×5 unit module		75%	readable

From Table 3, it can be seen that the similarity of the original algorithm is low, and directly using the 5×5 embedding algorithm can slightly increase the similarity but may cause the QR code to be unscannable. After adopting adaptive pixel depth adjustment, the clarity of the image in this solution will not be greatly affected, and it has a high Hash similarity, which is higher than the result of the image preprocessing in 21, and can maintain the original function of the QR code. One of the purposes of using the 5×5 module replacement algorithm is to increase the image clarity by enhancing the secret payload capacity. Since the QR code's module identification is based on the recognition of key pixels, it extends the single pixel block to a 5×5 size during QR code reading. As a result, the capacity for embedding secret information is independent of the error correction level but solely dependent on the size of the QR code version.

The comparison results of the payload capacity between this approach and other literature are shown in Figure 8.

Reference 810 embeds secret information by directly modifying some pixels of the QR code using the fault tolerance of QR codes, but this is limited by the upper limit of QR code fault tolerance. Reference 9 uses texture patterns to replace black modules in QR codes to embed secret information. Compared with the secret information embedding algorithms in 810, the 3×3 module replacement algorithm replaces all modules with sub-modules, so it has a higher information embedding capacity. However, this solution still does not meet the current algorithm requirements. By using an extended 5×5 adaptive contrast adjustment algorithm for secret information embedding, the secret information payload can be effectively increased. As shown in the line chart, compared with the 3×3 module replacement solution, this paper increases the payload capacity by three times for a single QR code. This solution has more significant advantages in terms of payload capacity compared to other solutions.

5.3. Analysis of two-level QR code experiment

Using the (2,3,3) structure as an example, the experimental results of applying the progressive recovery algorithm and the adaptive pixel depth adjustment algorithm to the two-level QR code, are shown in Figure 9. Two out of the three shares can be superimposed to reveal partial secret information in this example, and the original image is divided into three sub-shares, including two normal shares and one key share. In addition, three substitute shares are involved in repairing the key share. The carrier QR code corresponds to the same sequence number as the generated secret share according to the algorithm, and the corresponding two-level QR code is generated after embedding.

As shown in the recovery results in the figure, the image clarity is not affected after embedding the secret share into the QR code, and the primary information of the two-level QR code itself is not destroyed and can still be read. The superimposed result a is the result of ordinary share superimposition (i.e., the superimposition of two ordinary two-level QR codes and one key two-level QR code). In this case, the key two-level QR code c has not been attacked, so there is no need for the involvement of substitute two-level QR codes (d,e,f).

Upon cropping attack on the encrypted area of the key two-level QR code c, the corresponding two-level QR code g is obtained. In this case, the overlapping of the shares cannot restore the corresponding encrypted area, as shown in overlapping result b, where it can be seen that the critical data cannot be decrypted and identified. Overlapping result c is the result of recovery with the participation of substitute two-level QR codes (d,e,f), where it can be seen that the critical area is restored, but the clarity of identification is slightly lower than that of the key share.

The overlaid result d shows the recovery result of all shared parts in normal conditions. It can be seen that the result of using substitute parts to participate in recovery instead of critical parts has the same clarity as the result in normal conditions, thereby achieving the function of substitute parts to ensure the integrity of critical information.

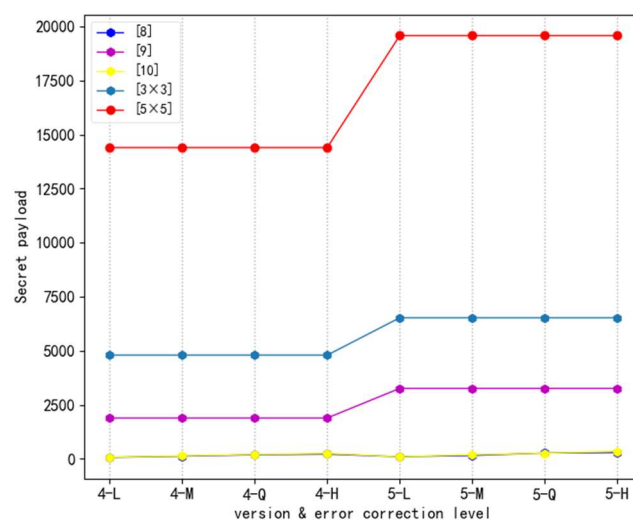


Figure 8. Comparison chart of secret payload capacity.

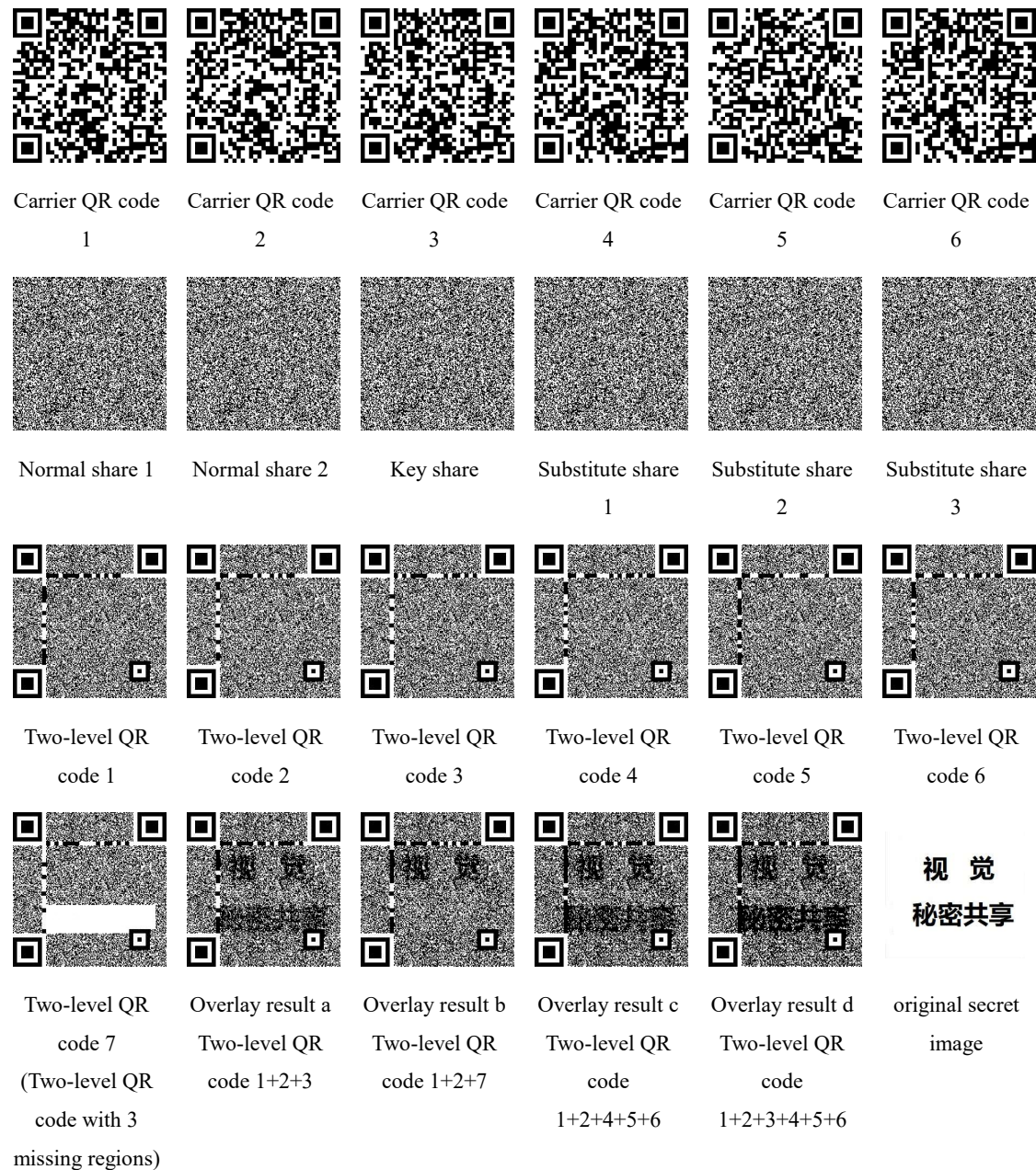


Figure 9. Embedding and recovery results of a two-level QR code under the (2,3,3) structure.

5.4. Robustness verification

In the process of transmission and storage of QR codes, they are inevitably subject to various attacks. This section tests the image robustness under three common attacks. First, it verifies whether the shared parts of the generated two-level QR code can be read. Second, it verifies whether the recovered secret image can be visually distinguished by the human visual system.

(1) Compression attack

In the process of image storage and transmission, various communication tools often compress the QR code sharing image to reduce storage space. To verify the effectiveness of this scheme after image compression attacks, tests were conducted by compressing the image with 5%, 10% and 15% lossy compression. The results of the compressed sharing QR code, decoded information, and secret

recovery are shown in Table 4.

Table 4. Result of compression attack test.

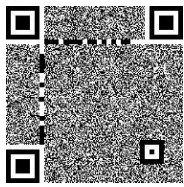
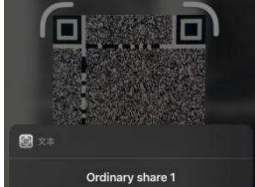

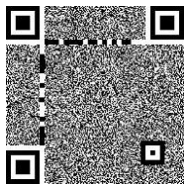
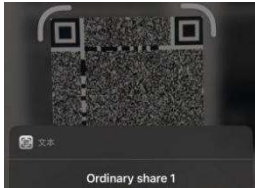

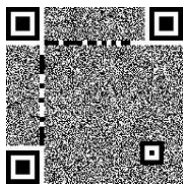
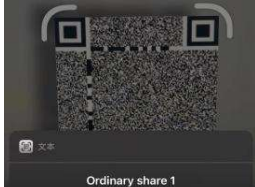

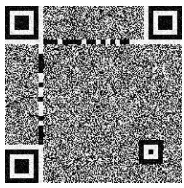

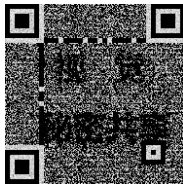
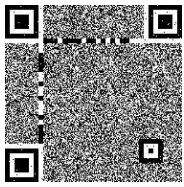
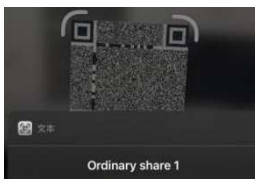
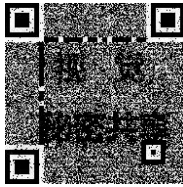
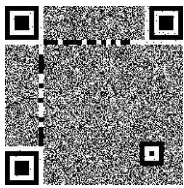
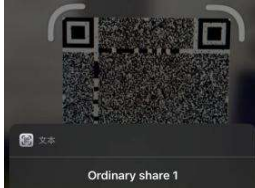
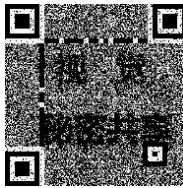
Compression ratio	Shared Part After Attack	Decoded Information	Result of Secret Information Restoration
5%			
10%			
15%			

Table 5. Result of noise attack test.

Type of Noise	Shared Part After Attack	Decoded Information	Result of Secret Information Recovery
Gaussian Noise			
Salt and Pepper Noise			
Multiplicative Noise			

According to the experimental results, it is shown that the information carried by the damaged third-level QR code shares can still be scanned, and after superimposing the secret image, it can still be recovered with high clarity.

(2) Noise attack




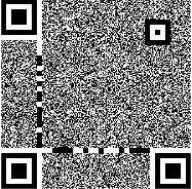
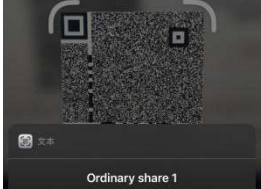

In the process of image transmission, QR code shares may be subject to noise interference or unintentional loss of some pixel information. In order to test the resistance of the proposed scheme to noise attacks, common noise attacks such as Gaussian noise, salt-and-pepper noise, and multiplicative noise were applied to the two-level shared QR code. To verify the effectiveness of the proposed scheme under noise attacks, the results of the shared QR code, decoded information, and recovered secret after noise attacks are shown in Table 5.

According to the experimental results, the information carried by the two-level QR code shares that have been subjected to noise attacks can still be scanned, and the secret image can still be recovered through superposition, with a certain degree of distinguishability.

(3) Rotation attack

During the scanning process of QR codes on mobile devices, the codes may be rotated at arbitrary angles. In order to verify the effectiveness of the proposed scheme under image rotation attacks, we performed rotation tests on the shared QR code shares by rotating them at 45° and 90° using Photoshop. The results of the rotated shared QR codes, decoded information, and recovered secret images are shown in Table 6.

Table 6. Result of rotation attack test.

Rotation Angle	Shared Part After Attack	Decoded Information	Result of Secret Information Recovery
45°			
90°			

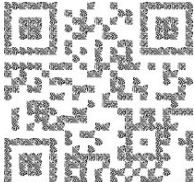
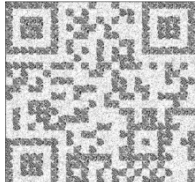
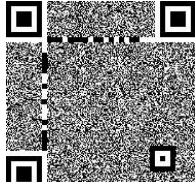
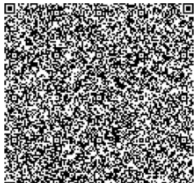
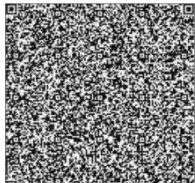

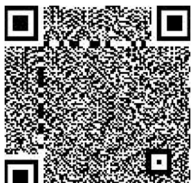

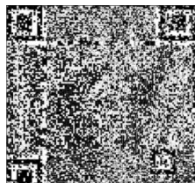
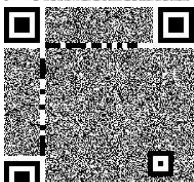
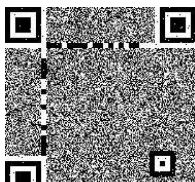
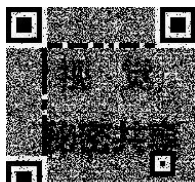
According to the experimental results, no matter how the QR code is rotated, the mobile device can still scan the public information carried by the carrier third-level QR code share, and the hidden information can be determined by the functional structure distribution of the QR code itself to determine the correct angle for superposition, avoiding unnecessary errors.

(4) Robustness experimental comparison

By comparing with other literature, this chapter's approach demonstrates an improved performance. The comparison results are shown in Table 7, where each share image is subjected to 10% pixel perturbation using salt-and-pepper noise to simulate either pixel loss or smudging. The test aims to verify the correct extraction of both the QR code's embedded information and the secret information.

From Table 7, it can be observed that our proposed approach can correctly extract the primary information carried by the QR code. Moreover, even after the recovery of secret information, it still maintains a certain level of recognizability compared to other schemes. Therefore, this approach exhibits good robustness.

Table 7. Robustness comparison experimental results.

	Shared image	Pixel Perturbation Handling	QR Code public information extraction results	Recovery results
[9]			Obtainable	
[10]			Unobtainable	
[21]			Obtainable	
This scheme			Obtainable	

6. Conclusions

This paper proposes a two-level QR code scheme based on a region matrix image secret sharing algorithm, which combines visual secret sharing technology with QR code grading technology. To address the security, concern of QR code information leakage, a progressive image secret sharing algorithm with hierarchical priority is introduced. This algorithm achieves the progressive recovery of different priority shares while maintaining a certain level of robustness to prevent the loss of high-priority shares. To tackle the issue of low resolution in the recovered image due to the limited payload of the original QR code grading scheme, an adaptive pixel depth adjustment algorithm is highlighted. This algorithm not only ensures that the scanning functionality of the QR code itself remains unaffected, but also effectively enhances the payload capacity of the secret image, leading to a substantial improvement in the clarity of the embedded image. The results demonstrate that the developed two-level QR code scheme ensures information confidentiality and can withstand conspiracy attacks. The shared images are robust and capable of fully recovering hidden information even under attack. Nevertheless, the contrast and clarity of the recovered image in the image secret sharing algorithm are impacted. Therefore, future work is planned to enhance clarity by emphasizing the contour details of the image through pre-processing.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments. This work was supported by Shaanxi Provincial Department of Science and Technology Youth Project (No. 2021JQ-575 and No.2021JQ-57); Shaanxi Provincial Department of Education Project (No. 19JK0526); Xi'an Science and Technology Plan Project (No.22GXFW0063); Yulin Science and Technology Bureau Project (No. 2016-24-4 and No. 2019-173).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. P. Y. Lin, Y. H. Chen, High payload secret hiding technology for QR codes, *EURASIP J. Image Video Process.*, **1** (2017), 1–8. <https://doi.org/10.1186/s13640-016-0155-0>
2. P. Y. Lin, Y. H. Chen, E. J. L. Lu, P. J. Chen, Secret hiding mechanism using QR barcode, in *2013 International Conference on Signal-Image Technology & Internet-Based Systems*, 2013. <https://doi.org/10.1109/SITIS.2013.15>
3. C. S. Chen, QR code authentication with embedded message authentication code, *Mobile Networks Appl.*, **22** (2017), 383–394. <https://doi.org/10.1007/s11036-016-0772-y>
4. A. Eritza, M. Ramadhan, H. Hafizah, Penerapan digital signature metode SHA dan DSA Pada slip gaji Pegawai, *JURSI TGD*, **1** (2022), 906–914. <https://doi.org/10.53513/jursi.v1i6.6002>
5. Q. B. Kang, K. Li, J. C. Yang, A digital watermarking approach based on DCT domain combining QR code and chaotic theory, in *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2014. <https://doi.org/10.1109/ICCP.2014.6937017>
6. G. H. Li, C. Chen, W. J. Wu, Y. F. Zheng, Y. F. Hong, X. M. Zhou, Method of imbedding and extracting watermark for two-dimensional code images, *Comput. Eng. Appl.*, **55** (2019), 103.
7. Y. W. Chow, W. Susilo, J. Baek, Covert QR codes: How to hide in the crowd, in *Information Security Practice and Experience: 13th International Conference, ISPEC 2017*, (2017), 678–693. https://doi.org/10.1007/978-3-319-72359-4_42
8. P. Y. Lin, Distributed secret sharing approach with cheater prevention based on QR code, *IEEE Trans. Ind. Inf.*, **12** (2016), 384–392. <https://doi.org/10.1109/TII.2015.2514097>
9. I. Tkachenko, W. Puech, C. Destruel, O. Strauss, Two-level QR code for private message sharing and document authentication, *IEEE Trans. Inf. Forensics Secur.*, **11** (2015), 571–583. <https://doi.org/10.1109/TIFS.2015.2506546>
10. S. Wan, Y. Lu, X. Yan, Y. Wang, C. Chang, Visual secret sharing scheme for (k, n) threshold based on QR code with multiple decryptions, *J. Real Time Image Process.*, **14** (2018), 25–40. <https://doi.org/10.1007/s11554-017-0678-3>

11. Y. Cheng, Z. Fu, B. Yu, G. Shen, A new two-level QR code with visual cryptography scheme, *Multimedia Tools Appl.*, **77** (2018), 20629–20649. <https://doi.org/10.1007/s11042-017-5465-4>
12. Y. Y. Liu, Z. X. Fu, Y. W. Wang, Two-level information management scheme based on visual cryptography and QR code, *Appl. Res. Comput.*, **33** (2016), 3460–3463.
13. H. K. Chu, C. S. Chang, R. R. Lee, N. J. Mitra, Halftone QR codes, *ACM Trans. Graphics*, **32** (2013), 1–8. <https://doi.org/10.1145/2508363.2508408>
14. B. Yu, S. J. Liu, Z. X. Fu, Design of gray visual cryptography scheme based on quick response code, *J. Comput. Aided Design Comput. Graphics*, **32** (2020), 635–642.
15. S. J. Liu, Z. X. Fu, B. Yu, A two-level QR code scheme based on polynomial secret sharing, *Multimedia Tools Appl.*, **78** (2019), 21291–21308. <https://doi.org/10.1007/s11042-019-7455-1>
16. Z. X. Fu, L. G. Fang, H. Y. Huang, B. Yu, Distributed three-level QR codes based on visual cryptography scheme, *J. Visual Commun. Image Representation*, **87** (2022), 103567. <https://doi.org/10.1016/j.jvcir.2022.103567>
17. B. Yu, H. Hu, W. P. Cheng, S. Gang, XOR-based region incrementing visual cryptography scheme with share block construction, *J. Electron. Inf. Technol.*, **37** (2015), 1978–1983. <https://doi.org/10.11999/JEIT141385>
18. H. C. Chao, T. Y. Fan, Random-grid based progressive visual secret sharing scheme with adaptive priority, *Digital Signal Process.*, **68** (2017), 69–80. <https://doi.org/10.1016/j.dsp.2017.05.009>
19. R. Sun, Z. X. Fu, X. P. Li, B. Yu, A novel size-invariant visual cryptography scheme based on two-level threshold, *J. Cryptologic Res.*, **8** (2021), 572–581. <https://doi.org/10.13868/j.cnki.jcr.000459>
20. B. Yu, G. Shen, Z. X. Fu, A lossless multi-secret sharing visual cryptography scheme, *J. Electron. Inf. Technol.*, **34** (2012), 2885–2890. <https://doi.org/10.3724/SP.J.1146.2012.00300>
21. L. Zhang, C. Cui, X. Zhang, W. Wu, Adaptive visual cryptography scheme design based on QR codes, *Math. Biosci. Eng.*, **19** (2022), 12160–12179. <https://doi.org/10.3934/mbe.2022566>



AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)