

## Article

# A Lossless-Recovery Secret Distribution Scheme Based on QR Codes

Jeng-Shyang Pan <sup>1,2</sup>, Tao Liu <sup>1</sup>, Bin Yan <sup>3</sup>, Hong-Mei Yang <sup>1</sup> and Shu-Chuan Chu <sup>1,\*</sup>

<sup>1</sup> College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China; jengshyangpan@gmail.com (J.-S.P.); taoliu0201@sdust.edu.cn (T.L.); yhm1998@163.com (H.-M.Y.)

<sup>2</sup> Department of Information Management, Chaoyang University of Technology, Taichung 413310, Taiwan

<sup>3</sup> College of Electronic and Information Engineering, Shandong University of Science and Technology, Qingdao 266590, China; yanbinhit@hotmail.com

\* Correspondence: scchu0803@sdust.edu.cn

**Abstract:** The visual cryptography scheme (VCS) distributes a secret to several images that can enhance the secure transmission of that secret. Quick response (QR) codes are widespread. VCS can be used to improve their secure transmission. Some schemes recover QR codes with many errors. This paper uses a distribution mechanism to achieve the error-free recovery of QR codes. An error-correction codeword (ECC) is used to divide the QR code into different areas. Every area is a key, and they are distributed to  $n$  shares. The loss of any share will make the reconstructed QR code impossible to decode normally. Stacking all shares can recover the secret QR code losslessly. Based on some experiments, the proposed scheme is relatively safe. The proposed scheme can restore a secret QR code without errors, and it is effective and feasible.

**Keywords:** VCS; secure transmission; QR code; ECC

## 1. Introduction

The rapid development of the Internet has prompted the beginning of the information age [1–4]. As a two-dimensional image, the quick response (QR) code carries a lot of information [5]. A decoder can decode its information by scanning it. People can perform operations using QR codes [6], such as visiting a website, obtaining text, using mobile payment systems, and many more. The convenience of QR codes makes them popular. The spread of information across a network requires security [7–10]. The standard for a QR code is public, and its security on the Internet is dependent on methods such as authenticated key exchange [11–14], watermarking technology [15–18], or information hiding [19–21].

Compared to those, the visual cryptography scheme (VCS) [22–26] is suitable to encrypt QR codes.

The  $(n, n)$ -VCS distributes a secret among images [27–30]. Each share is a key. All shares perform the decoding method to reconstruct the secret. The secret will be not constructed if any share is not present. The main idea of the VCS is distribution. The secret is distributed among some images. Every share contains no information about the secret image. Naor and Shamir designed two fundamental matrices for the  $(k, n)$ -VCS ( $k \leq n$ ) [31]. The secret is distributed to  $n$  images. Receivers need to stack no fewer than  $k$  shares to recover the secret. The recovered image requires the human visual system (HVS) for decoding. This scheme is used to encrypt a binary image. Liu et al. proposed an extended VCS [32]. A VCS for gray-level and color images was proposed by Hou using halftone technology and the subtractive model [33]. The two schemes are pixel-expansile. Based on the random grid and halftone technology, Shyu designed a VCS for color images without pixel expansion [34]. Error diffusion was used to design a VCS [35]. Luo et al. designed a VCS to encrypt a continuous tone image [36], mainly relying on halftone



**Citation:** Pan, J.-S.; Liu, T.; Yan, B.; Yang, H.-M.; Chu, S.-C. A Lossless-Recovery Secret Distribution Scheme Based on QR Codes. *Entropy* **2023**, *25*, 653. <https://doi.org/10.3390/e25040653>

Received: 19 March 2023

Revised: 8 April 2023

Accepted: 11 April 2023

Published: 13 April 2023



**Copyright:** © 2023 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/>).

technology. Liu designed a novel  $(t, s, k, n)$ -VCS [37]. Yan et al. designed a robust VCS resistant to noise in shares [38]. A color VCS was proposed for a binary image by the authors in [39]. A QR code was used in a VCS for a single-pixel image [40].

As an image, a QR code can also be encrypted with VCS to enhance its security. Fang designed a VCS for a QR code using two fundamental matrices [41]. The QR code is distributed to two shares with meaningless images. Stacking the two shares can generate a new image. This image is restored to a black-and-white QR code using post-processing. Fang's scheme is pixel-expansible. Chow et al. used XOR to achieve a scheme without pixel expansion [42]. This scheme is an  $(n, n)$ -VCS, improved by Chen to  $(k, n)$ -VCS [43]. Since it has errors, the recovered QR code sacrifices its correction ability. A color VCS for a QR code by [44] Wan et al. used a big version of a QR code as the share to achieve a VCS [45]. The reconstructed QR code with errors was decoded using the error-correction codeword (ECC). Huang et al. designed a high-security VCS for a secret QR code based on the error-correction mechanism [46].

Fang's scheme can restore a QR code without errors. However, it is a pixel-expansible scheme. Some schemes are not pixel-expansible, and they recover a QR code with errors. To design a non-pixel-expansible scheme to restore a QR code without errors, a new VCS is proposed to share the QR code. This paper analyzes the ECC and designs a method to divide the QR code into  $n$  areas. These areas are distributed into  $n$  shares. When all shares are stacked, the secret QR code is reconstructed losslessly. If  $n - 1$  or fewer shares are stacked, the recovered QR code will have many errors. The number of wrong codewords will exceed the capacity of the ECC. The standard decoder will not decode it. It is useless if it does not reveal the secret. The method in this paper is non-pixel-expansible. All versions of the QR code can be encrypted by it, but not other binary images.

The main contributions of this paper are as follows:

1. This paper uses the mechanism of the ECC to divide the QR code into  $n$  areas. Every area is distributed to  $n$  shares to achieve a non-pixel-expansible VCS. Compared to the pixel-expansible VCS (such as [37]), the generated share is the same size as the secret image using the proposed VCS.
2. Compared with [42], this paper can recover a QR code losslessly. The capacity of the ECC is not sacrificed in the recovered QR code.
3. There are no measures to protect itself during the transmission of the QR code. The proposed VCS provides a method to ensure the security of QR codes across the network.

The rest of the paper is structured as follows: Section 2 shows the VCS, the QR code, and the decoding stacking method. The proposed VCS is introduced in Section 3. All experiments are described in Section 4. Section 5 offers some conclusions.

## 2. Preliminaries

This section introduces the VCS (Section 2.1), the QR code (Section 2.2) and the decoding stacking method (Section 2.3). This is an abbreviation, as shown in Table 1.

### 2.1. VCS

The  $(k, n)$ -VCS was presented by Naor and Shamir [31]. A secret can be distributed among images. The secret is reconstructed when no fewer than  $k$  shares are stacked. Secret information about the restored image is recognized using the HVS. The decoding process does not need complex calculations. The user does not need complex computer knowledge.

Let  $\mathbf{R}_\bullet$  (resp.  $\mathbf{R}_o$ ) be a black (resp. white) pixel block in the recovered image. The  $n_\bullet(\cdot)$  represents the number of black pixels. The  $(k, n)$ -VCS satisfies two conditions:

**Security Condition:** Fewer than  $k$  shares being stacked will satisfy  $n_\bullet(\mathbf{R}_\bullet) = n_\bullet(\mathbf{R}_o)$  in the recovered image.

**Contrast Condition:** The image generated by stacking more than  $k - 1$  shares will satisfy  $n_\bullet(\mathbf{R}_\bullet) > n_\bullet(\mathbf{R}_o)$ .

**Table 1.** Denotation of symbols.

Abbreviation	Explanation
H	The secret image
R	The recovered image
$S_i$	Share
•	Black
○	White
k	Black pixel
w	White pixel
$n(\cdot)$	The function of the quantity
n	Natural number (N)
c	The number of total codewords
a	The number of data codewords
r	The capacity of correction in the QR code
b	The block of the QR code
$\rightarrow$	Generation
Area Q	The module collection of QR code
Stacking(·)	The decoding operation of the VCS

When attackers use no more than  $k$  shares to stack, the generated image has no color contrast. It is a useless image, which will not reveal away secrets. When no fewer than  $k$  shares are stacked, the generated image has color contrast. This image is useful.

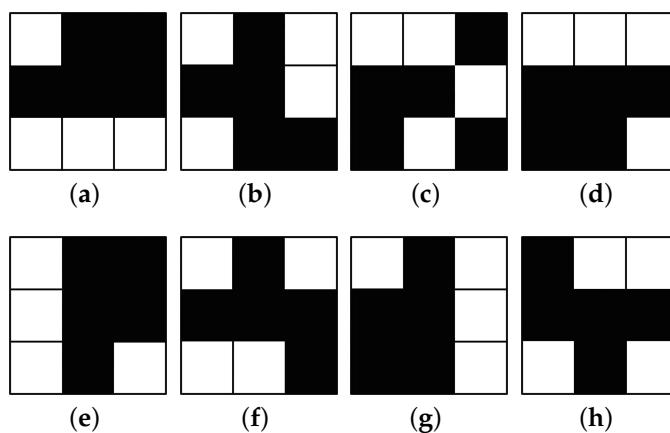
Naor and Shamir's VCS is pixel-expansible. It shares a secret image using two fundamental matrices,  $\mathbf{M}_\bullet$  and  $\mathbf{M}_o$ .  $\mathbf{M}_\bullet$  encrypts the black pixels. The white pixels are encrypted using  $\mathbf{M}_o$ . As an example, a (4, 4)-VC scheme is shown below:

k and the w are the black and white pixels, respectively. In (4, 4)-VC,  $\mathbf{M}_o$  and  $\mathbf{M}_\bullet$  have:

$$\mathbf{M}_o = \begin{bmatrix} w & k & k & k & k & k & w & w & w \\ w & k & w & k & k & w & w & k & k \\ w & w & k & k & k & w & k & w & k \\ w & w & w & k & k & k & k & k & w \end{bmatrix},$$

$$\text{and } \mathbf{M}_\bullet = \begin{bmatrix} w & k & k & w & k & k & w & k & w \\ w & k & w & k & k & k & w & w & k \\ w & k & w & k & k & w & k & k & w \\ k & w & w & k & k & k & w & k & w \end{bmatrix}. \quad (1)$$

Every line of the  $\mathbf{M}_o$  and  $\mathbf{M}_\bullet$  represents a  $3 \times 3$  pixel block, as shown in Figure 1.

**Figure 1.** (a–d): they share the white pixels; (e–h): they share the black pixels.

When no fewer than four shares are stacked,  $n_\bullet(\mathbf{R}_\bullet) = n_o(\mathbf{R}_o)$  is established in the recovered secret. That makes the reconstructed image non-contrasting in color. The

recovered image is useless without any information. When four shares are stacked, the recovered image satisfies  $n_{\bullet}(\mathbf{R}_{\bullet}) = 9 > n_{\bullet}(\mathbf{R}_{\circ}) = 8$ . Its color has contrast, HVS can decode it, and the secret information is obtained.

For the  $(n, n)$ -VCS, only  $n$  shares can restore the secret. If  $n - 1$  or fewer shares are stacked, the secret will not be recovered. The decoding method of VCS is public, and the attacker knows it. If the attacker cannot obtain all shares, they cannot decode the secret. When the VCS is of the  $(k, n)$  type, the attacker needs to obtain  $k$  or more shares to decode the secret.

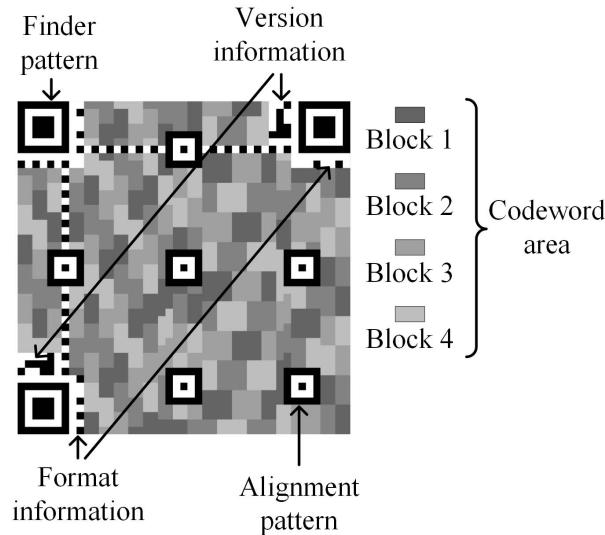
## 2.2. QR Code

QR codes are two-dimensional images carrying information [47] that are encrypted into black and white modules.  $D \times D$  modules are combined into a QR code.  $D$  can be calculated as follows:

$$D = 17 + V \times 4, \quad (2)$$

where  $V$  is the version of the QR code ( $V \leq 40$ ). The  $V$  decides the number of modules. The larger  $V$  is, the larger  $D$  is.

A QR code consists of a finder pattern, version and format information, alignment pattern, and codeword area, as shown in Figure 2. The codeword area is made up of blocks. Every block carries some data and error-correction codewords. Every codeword consists of 8 modules. A module is a square pattern. It is a pixel or  $2 \times 2$  pixels or  $n \times n$  pixels.



**Figure 2.** The QR code structure of version 7 and error-correction level M.

The finder pattern determines the location of the QR code. The ECC corrects the errors. The capacity of the ECC is over four levels, namely L, M, Q, and H. L denotes that the ECC corrects about 7% errors in the QR code. Similarly, M is 15%, Q is 25% and H is 30%. These error-correction capacities are approximations. Different QR codes can correct different numbers of codewords. The capability of each block is different. An example of the capacity of the ECC is shown in Table 2. Any module with errors will make the codeword wrong. A module can cause a codeword error. When a block has wrong codewords, the QR code cannot be decoded correctly. A few wrong modules create many wrong codewords in a block. These errors prevent the QR code from being decoded correctly.

**Table 2.** The correcting ability (part of the ISO [48]).

Version	Error Correction Level	Number of Codewords ( $c, a, r$ )
4	L	(100, 80, 10)
	M	(50, 32, 9)
	Q	(50, 24, 13)
	H	(25, 9, 8)
5	L	(134, 108, 13)
	M	(67, 43, 12)
	Q	(33, 15, 9)
		(34, 16, 9)
	H	(33, 11, 11) (34, 12, 11)

$c, a$  and  $r$  are the numbers of total codewords, data codewords, and the capacity of correction, respectively.

The information is encoded as a 0–1 matrix (this process is reversible). This matrix is rendered as a black-and-white two-dimensional image. The graphical rule is that 0 (resp. 1) represents black (resp. white) modules. QR codes consist of black and white modules. When a decoder decodes QR codes, it scans them to confirm the black and white modules. All modules are then converted into a 0–1 matrix and the information is recovered. If QR codes have many errors, they cannot be decoded. For example, version 4 and error-correction level H of the QR code has nine wrong codewords in a block (it can correct eight wrong codewords;  $r = 8$  in Table 2). Therefore, it will not be decoded, and is meaningless. The QR code will lose its value. The error of the QR code refers to the black (resp. white) modules that are changed to white (resp. black) modules.

### 2.3. Stacking

This paper uses stacking to denote a specific operation performed on the colors in the image. The process for stacking the two colors is as follows:

**Definition 1.** *Stacking: stacking ( $k, w$ ) →  $k$ , stacking ( $w, w$ ) →  $w$ , stacking ( $k, k$ ) →  $k$ .*

When the two colors are stacked, white and white generate white (other color combinations generate black). This paper uses this operation (stacking) to achieve a secret recovery. All operations are shown in Table 3. The operation of stacking is used to recover the secret [31]. It is also the method used in this paper to restore a secret QR code.

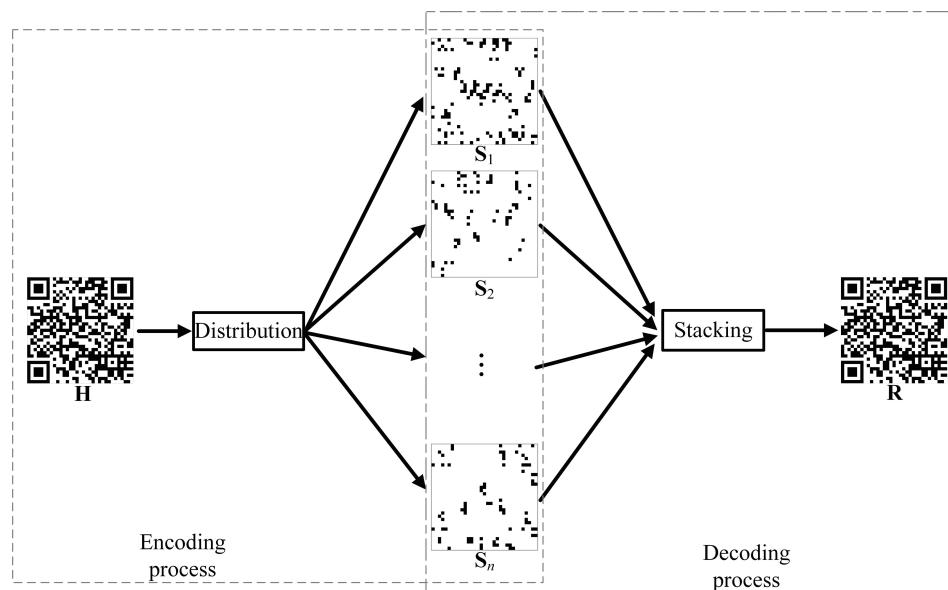
**Table 3.** The operation of stacking.

Pixel Color-1	Pixel Color-2	Stacking

### 3. The Proposed VCS

This paper proposes an  $(n, n)$ -threshold VCS. It can distribute a secret QR code (**H**) to  $n$  meaningless images. When  $n$  shares are stacked, the reconstructed QR code (**R**) can be obtained without errors. All processes are described in Figure 3. When the receiver obtains all shares, it can stack them to recover the secret QR code. The restored QR code can then be decoded by the decoder. The secret QR code is encrypted and decrypted using

a VCS. The security of its transmission across the network is improved. When all shares are obtained, an attacker can decode the QR code. If any share is missed, the attacker will not be able to decode the QR code.



**Figure 3.** The proposed VCS.

### 3.1. Conditions of the Proposed VCS

Let  $n_i^b(\cdot)$  show that block  $b$  has  $n_i^b(\cdot)$  incorrect codewords.  $r^b$  denotes the number of erroneous codewords that ECC can correct in block  $b$ . Every block can correct different numbers of the wrong codeword. The capacities of the different blocks are independent. Let  $n_b(\cdot)$  be the number of blocks. The proposed VCS satisfies **Condition 1** and **Condition 2**:

$$\text{Stacking } (\overbrace{S_1, \dots, S_i}^j) \rightarrow \mathbf{R}. \quad (3)$$

$$\begin{cases} \text{Condition 1 :} \\ \text{When } j < n, \exists b (b \in [1, n_b(\mathbf{H})]) \text{ makes } n_i^b(\mathbf{R}) > r^b. \\ \text{Condition 2 :} \\ \text{When } j = n, \forall b (b \in [1, n_b(\mathbf{H})]) \text{ makes } n_i^b(\mathbf{R}) \leq r^b. \end{cases} \quad (4)$$

When  $\exists b (b \in [1, n_b(\mathbf{H})]), n_i^b(\mathbf{R}) > r^b$  is satisfied in the  $\mathbf{R}$ , the number of the incorrect codeword exceeds  $r^b$ . The  $\mathbf{R}$  is hard to decode. Secret information will not be revealed. If the  $\mathbf{R}$  satisfies  $\forall b (b \in [1, n_b(\mathbf{H})]), n_i^b(\mathbf{R}) \leq r^b$ , it is decoded using the decoder, and the secret is obtained.

Every share is a key to restoring the secret QR code. If any share is missing, the restored QR code will have many errors. The number of wrong codewords is exceeded by the capacity of the ECC. This wrong QR code cannot be decoded using the standard decoder. The security of the VCS for the QR code is therefore ensured.

### 3.2. Encryption and Decryption Processes

To restore a QR code without errors, the white module in the QR code must also be the white module in the share. The black module of the QR code is a black module or white module in the share. Therefore, only the black modules are processed in the VCS.

Every block has error-correction codewords, which correct incorrect codewords. A QR code with more than  $r^b$  wrong codewords will not be decoded. Therefore, any two shares

need to recover more than  $r^b$  codewords. If any share is not obtained, the recovered QR code will have more than  $r^b$  wrong codewords. ECC can correct up to about 30%. To be more secure, when a share is missing, about half of the codeword will be wrong in a certain block of the restored QR code. Any two shares can restore modules of at least  $d$  codewords (the codeword is restored or the part of the codeword is restored) in every block. The  $d$  can be calculated by:

$$d = \left\lfloor \frac{c}{2} \right\rfloor, \quad (5)$$

where a block has  $c$  codewords of the data and error correction. When  $c$  has different values in a QR code, it takes the minimum value. We use  $d$  and ECC to divide the QR code into area  $\mathbf{Q}_\bullet^l$  and area  $\hat{\mathbf{Q}}_\bullet^l$  ( $l = 1, 2, \dots, n$ ).

Every area should include modules that belong to many codewords. For example, one or two modules per codeword are in each area. The module of a codeword is wrong, and this codeword is wrong. Therefore, an area is missed that will cause many codeword errors. If these wrong codewords are in a block, the QR code will not be decoded.

### 3.2.1. The Generation of $\mathbf{Q}_\bullet^l$

The area of the codeword is divided into area  $\mathbf{Q}_\bullet^l$  ( $l = 1, 2, \dots, n$ ). The area  $\mathbf{Q}_\bullet^l$  is generated in two cases as follows:

**Case 1**, When  $n \in [2, 2n_b(\mathbf{H})]$ :

If  $n \leq n_b(\mathbf{H})$ , the area  $\mathbf{Q}_\bullet^l$  is generated as follows:

**a.** We select  $n$  blocks randomly. The black modules of every selected block form the area  $\mathbf{Q}_\bullet^l$ , where  $l = 1, 2, \dots, n$ .

**b.** The area of all unselected black modules in each block is added to area  $\mathbf{Q}_\bullet^n$ .

For example,  $n = 3$ ,  $n_b(\mathbf{H}) = 4$  and selected blocks are block 1-block 3. Area  $\mathbf{Q}_\bullet^1$ ,  $\mathbf{Q}_\bullet^2$  and  $\mathbf{Q}_\bullet^3$  are the area of all black modules in block 1, block 2 and (block 3 and block 4), respectively.

If  $n > n_b(\mathbf{H})$ , the area  $\mathbf{Q}_\bullet^l$  is generated as follows:

**a.** We select  $2n_b(\mathbf{H}) - n$  blocks randomly. The black modules of every selected block form the area  $\mathbf{Q}_\bullet^{l_1}$ , where  $l_1 = 1, 2, \dots, 2n_b(\mathbf{H}) - n$ .

For example,  $2n_b(\mathbf{H}) - n = 3$  and selected blocks are block 1-block 3. Area  $\mathbf{Q}_\bullet^1 - \mathbf{Q}_\bullet^3$  is the area of all black modules in block 1-block 3, respectively.

**b.** The area of all black modules ( $d$  different unselected codewords in every unselected block) forms the area  $\mathbf{Q}_\bullet^{l_2}$  ( $l_2 = 2n_b(\mathbf{H}) - n + 1, 2n_b(\mathbf{H}) - n + 2, \dots, n$ ). If a block does not have  $d$  different codewords, it will select the next block. There is no intersection for any subset in area  $\mathbf{Q}_\bullet^{l_2}$ .

For example,  $n_b(\mathbf{H}) - (2n_b(\mathbf{H}) - n) = 1$  and an unselected block is block 4. Area  $\mathbf{Q}_\bullet^4$  and area  $\mathbf{Q}_\bullet^5$ , respectively, consist of all black modules in  $d$  different codewords. They satisfy  $\text{area } \mathbf{Q}_\bullet^4 \cap \text{area } \mathbf{Q}_\bullet^5 = \emptyset$ . These codewords are all from block 4.

**c.** The area of all unselected black modules (every block) is added to area  $\mathbf{Q}_\bullet^n$ . The area  $\mathbf{Q}_\bullet^l = \text{area } \mathbf{Q}_\bullet^{l_1} \cup \text{area } \mathbf{Q}_\bullet^{l_2}$ , which  $l = 1, 2, \dots, n$ .

**Case 2**, When  $n > 2n_b(\mathbf{H})$ :

We sort the number of black modules of each codeword from small to large in block  $b$ . Area  $\mathbf{A}_\bullet^b$  consists of the last  $d$  codewords in the sequence in block  $b$ . Area  $\hat{\mathbf{A}}_\bullet^b$  is made up of the rest of the codewords in block  $b$ . The minimum number of black modules for every codeword in area  $\mathbf{A}_\bullet^b$  (resp.  $\hat{\mathbf{A}}_\bullet^b$ ) is denoted by  $n_1$  (resp.  $n_2$ ). The value of  $b$  is  $b = 1, 2, \dots, n_b(\mathbf{H})$ . The biggest value ( $n_m$ ) of  $n$  is:

$$n_m = \sum_{b=1}^{n_b(\mathbf{H})} (n_1 + n_2). \quad (6)$$

When  $n > n_m$ , the proposed scheme is not suitable and  $n \in [2, n_m]$ .

Area  $\mathbf{Q}_\bullet^l$  generates rules as follows:

*Rule 1:*

$$\begin{cases} \text{Area } \mathbf{Q}_\bullet^{l_1} = \text{Area } \hat{\mathbf{A}}_\bullet^{l_1}, & \text{if } l_1 = 1, 2, \dots, n_b(\mathbf{H}), \\ \text{Area } \mathbf{Q}_\bullet^{l_1} = \text{Area } \mathbf{A}_\bullet^{l_1 - n_b(\mathbf{H})}, & \text{if } l_1 = n_b(\mathbf{H}) + 1, n_b(\mathbf{H}) + 2, \dots, 2n_b(\mathbf{H}). \end{cases} \quad (7)$$

*Rule 2:* Select a black module from each codeword (these codewords are from area  $\mathbf{Q}_\bullet^{l_1}$ , where  $l_1$  is a random number from 1 to  $2n_b(\mathbf{H})$ ) to perform  $n - 2n_b(\mathbf{H})$  times to form area  $\mathbf{Q}_\bullet^{l_3}$ , where  $m_\bullet^b, \hat{m}_\bullet^b > 1$  and  $l_3 = 2n_b(\mathbf{H}) + 1, 2n_b(\mathbf{H}) + 2, \dots, n$ . The selected modules are deleted in area  $\mathbf{Q}_\bullet^{l_1}$  each time.

For example,  $n - 2n_b(\mathbf{H}) = 3$ . Areas  $\mathbf{Q}_\bullet^{2n_b(\mathbf{H})+1}, \mathbf{Q}_\bullet^{2n_b(\mathbf{H})+2}$  and  $\mathbf{Q}_\bullet^{2n_b(\mathbf{H})+3}$  are made up of all black modules in the first, second, and third to be chosen, respectively. The selected modules are deleted in the area  $\mathbf{Q}_\bullet^{l_1}$ .

*Rule 3:* Area  $\mathbf{Q}_\bullet^l = \text{area } \mathbf{Q}_\bullet^{l_1} \cup \text{area } \mathbf{Q}_\bullet^{l_2} \cup \text{area } \mathbf{Q}_\bullet^{l_3}$ , which  $l = 1, 2, \dots, n$ .

### 3.2.2. The Generation of Area $\hat{\mathbf{Q}}_\bullet^l$

The remaining black module (except for the area of the codeword) is divided into area  $\hat{\mathbf{Q}}_\bullet^l$ . The process is as follows:

*a.* Let area  $\mathbf{H}_\bullet$  (resp. area  $\mathbf{H}_o$ ) be black (resp. white) modules of  $\mathbf{H}$ . All black modules (except for the area  $\mathbf{Q}_\bullet^l$ ) form the area  $\hat{\mathbf{Q}}_\bullet$  in area  $\mathbf{H}_\bullet$ . Area  $\mathbf{H}_\bullet = \text{area } \hat{\mathbf{Q}}_\bullet \cup \text{area } \mathbf{Q}_\bullet^l$ . The  $n_\bullet(\cdot)$  is the number of black modules.

*b.* When  $n \geq n_\bullet(\hat{\mathbf{Q}}_\bullet)$ , select  $n_\bullet(\hat{\mathbf{Q}}_\bullet)$  different modules of the area  $\hat{\mathbf{Q}}_\bullet$  to form area  $\hat{\mathbf{Q}}_\bullet^{l_1}$ , where  $l_1 = 1, 2, \dots, n_\bullet(\hat{\mathbf{Q}}_\bullet)$ . Select  $n - n_\bullet(\hat{\mathbf{Q}}_\bullet)$  modules of the area  $\hat{\mathbf{Q}}_\bullet$  to form area  $\hat{\mathbf{Q}}_\bullet^{l_2}$ , where  $l_2 = n_\bullet(\hat{\mathbf{Q}}_\bullet) + 1, n_\bullet(\hat{\mathbf{Q}}_\bullet) + 2, \dots, n$ . The selected modules should not be repeated where possible. Area  $\hat{\mathbf{Q}}_\bullet^l = \text{area } \hat{\mathbf{Q}}_\bullet^{l_1} \cup \text{area } \hat{\mathbf{Q}}_\bullet^{l_2}$ , where  $l = 1, 2, \dots, n$ . Every area is made up of a black module in the area  $\hat{\mathbf{Q}}_\bullet^l$ .

When  $n < n_\bullet(\hat{\mathbf{Q}}_\bullet)$ , we select  $\left\lfloor \frac{n_\bullet(\hat{\mathbf{Q}}_\bullet)}{n} \right\rfloor$  black modules (not repeating) from area  $\hat{\mathbf{Q}}_\bullet$  for  $n$  times to form area  $\hat{\mathbf{Q}}_\bullet^l$  which  $l = 1, 2, \dots, n$ . The unselected area of all the black modules from the area  $\hat{\mathbf{Q}}_\bullet$  is added to area  $\hat{\mathbf{Q}}_\bullet^n$ . For example, area  $\hat{\mathbf{Q}}_\bullet^1$  is the area of  $\left\lfloor \frac{n_\bullet(\hat{\mathbf{Q}}_\bullet)}{n} \right\rfloor$  black modules. Area  $\hat{\mathbf{Q}}_\bullet^n$  is the area of several black modules (the number is no less than  $\left\lfloor \frac{n_\bullet(\hat{\mathbf{Q}}_\bullet)}{n} \right\rfloor$ ).

### 3.2.3. The Encryption Process of the Proposed VCS

$\mathbf{H}$  is distributed to  $n$  shares. The encoding of the proposed VCS for  $\mathbf{H}$  is:

**Step 1:**  $\mathbf{H}$  is divided into area  $\mathbf{H}_o$ , area  $\hat{\mathbf{Q}}_\bullet^l$  and area  $\mathbf{Q}_\bullet^l$  by Sections 3.2.1 and 3.2.2.

**Step 2:** Generate  $n$  shares as follows:

$$\begin{cases} \mathbf{S}_l(x, y) \leftarrow k, \text{ if } \mathbf{H}(x, y) \in \text{area } \hat{\mathbf{Q}}_\bullet^l, \\ \mathbf{S}_l(x, y) \leftarrow k, \text{ if } \mathbf{H}(x, y) \in \text{area } \mathbf{Q}_\bullet^l, \\ \mathbf{S}_l(x, y) \leftarrow w, \text{ others.} \end{cases} \quad (8)$$

Here,  $\mathbf{S}_l(x, y) \leftarrow k$  (resp.  $\mathbf{S}_l(x, y) \leftarrow w$ ) represents the color of  $\mathbf{S}_l(x, y)$  is modified by black (resp. white).

All processes are shown in Algorithm 1. The secret QR code is divided into several areas. Every area has some black modules in a certain block. If this area is missed or is not used to restore the QR code, this restored QR code will have errors. These errors will cause the restored QR code not to be decoded by the standard decoder. All areas are distributed by  $n$  shares to share the secret QR code.

**Algorithm 1** The distribution algorithm**Input:**

$H$ , the threshold  $n$ , area  $\hat{Q}_\bullet^l$ , and area  $Q_\bullet^l$  ( $l = 1, 2, \dots, n$ ).

**Output:**

These  $n$  shares:  $S_1, S_2, \dots, S_n$ .

```

1: for  $x$  from 1 to  $D$  do
2:   for  $y$  from 1 to  $D$  do
3:     /*  $H$  consists of  $D \times D$  modules.*/
4:     if  $H(x, y) \in$  area  $\hat{Q}_\bullet^1$  or area  $Q_\bullet^1$  then
5:        $S_1(x, y) \leftarrow k$ .
6:     else
7:        $S_1(x, y) \leftarrow w$ .
8:     end if
9:     if  $H(x, y) \in$  area  $\hat{Q}_\bullet^2$  or area  $Q_\bullet^2$  then
10:       $S_2(x, y) \leftarrow k$ .
11:    else
12:       $S_2(x, y) \leftarrow w$ .
13:    end if
14:    :
15:    if  $H(x, y) \in$  area  $\hat{Q}_\bullet^n$  or area  $Q_\bullet^n$  then
16:       $S_n(x, y) \leftarrow k$ .
17:    else
18:       $S_n(x, y) \leftarrow w$ .
19:    end if
20:   end for
21: end for

```

**Output:**  $S_1, S_2, \dots, S_n$

### 3.2.4. The Decryption of the Proposed VCS

The decryption operation of this paper is as follows: the receiver needs to obtain  $n$  shares. These  $n$  shares are stacked to reconstruct the QR code. Its operation is:

$$\text{Stacking } (S_1, S_2, \dots, S_n) \rightarrow R. \quad (9)$$

$R$  is completely recovered. Its information is obtained using the decoder. When the receiver obtains all the shares, it needs to stack all the shares to recover the secret QR code. Each share contains nothing. They are meaningless.

### 3.3. Analysis of the Proposed VCS

This paper proposes an  $(n, n)$ -VCS. Use the mechanism of the ECC to ensure that no more than  $n$  shares can reconstruct the secret image losslessly. The number recovering the codeword exceeds the capacity of the ECC by every share. There is about half the number of the wrong codewords in the restored image in a specific block when the number is less than  $n$ , using Equation (5). The ECC of the QR code cannot correct half the codeword in every block. Therefore, the proposed scheme is relatively safe. When the attacker obtains  $n - 1$  or fewer shares, they can know that the secret image is a QR code. However, they cannot decode the secret QR code. The attacker does not obtain the secret.

If the attacker obtains  $n - 1$  shares, they will stack them to generate an image that resembles a QR code. However, it will not be decoded. Some codewords have one or many wrong modules that will cause these codewords to be wrong. The errors refer to some black modules being white. If a module is wrong, it makes a codeword wrong. It is hard to confirm which white module is wrong in a recovered image. The security is mainly reflected in the following points:

1. This new image has more wrong codewords than it can correct ( $n - 1$  shares). Therefore, it cannot be decoded. The attacker will not obtain the secret.
2. The attacker knows some white modules are wrong. However, they cannot determine which areas are wrong. The errors are scattered throughout the QR code.

Due to the particularity of the QR code, this proposed scheme is secure. These  $n - 1$  shares are used to recover the wrong QR code, which cannot be decoded. All recovered black modules are right, no matter how many shares. The number of wrong modules is very small, but they are mixed with the correct ones. There are many white codewords in the QR code. The right white modules and the wrong white modules are mixed, and are difficult to correct by any means.

This paper uses the mechanism of the ECC to design a QR code. Every share can restore a certain number of codewords. This number will exceed the number that the ECC can correct. Therefore, the number of shares is connected to the number of codewords. The  $n$  is limited in the proposed  $(n, n)$ -VCS and  $n \in [2, n_m]$ .

This paper designs a scheme for a QR code. When a QR code is transmitted across a network, it can be obtained by an attacker. The standard of the QR code is public. Therefore, this QR code can easily be decoded. When using the proposed scheme to transmit the QR code, the attacker needs to obtain all the shares to restore the QR code. The proposed scheme enhances the security of the QR code on the Internet.

#### 4. Experiments

The designed QR code follows ISO standard [48] and the Zxing library [49]. All experiments are introduced in this section.

##### 4.1. The Simulated Experiment of the Proposed VCS

Version 4 and the error-correction level H (4-H) of the QR code are adopted in this section. The experimental results are shown in Figure 4.

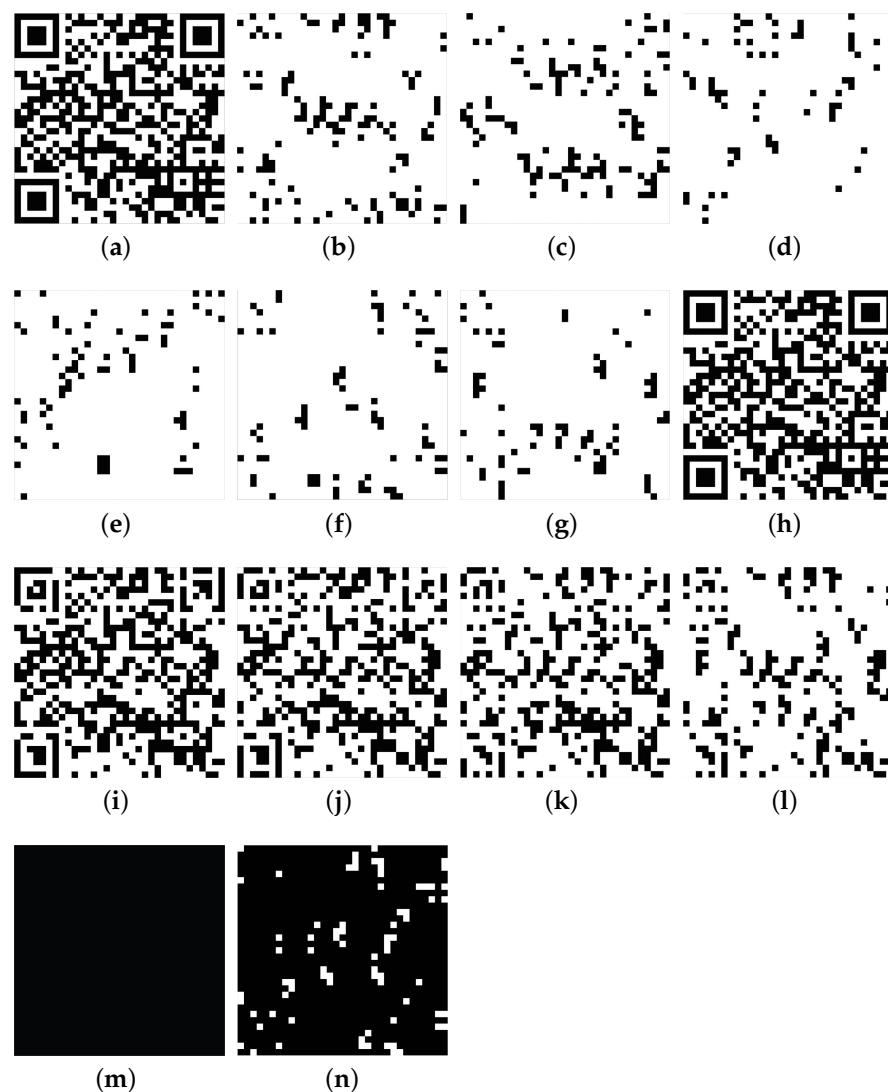
The H (Figure 4a) can be distributed to six shares (Figure 4b–g) when  $n = 6$ . When six shares are stacked, the QR code will be recovered without errors (Figure 4h). Five or fewer shares can restore an image that resembles a QR code by HVS (Figure 4i–l). However, a standard decoder cannot decode them, and they are all useless images.

Table 4 shows some experimental results. The experiment is tested with Figure 4h–l. The decoder is a standard decoder. The results show that Figure 4 is decoded and Figure 4i–l cannot be decoded. These results verify that the proposed scheme is safe.

When stacking different numbers of shares, the reconstructed QR code has different numbers of wrong codewords, as shown in Table 5. Six shares can fully recover the QR code. If a QR code is 4-H, it can correct eight codewords in every block. If the number of stacked shares is fewer than six, the recovered QR code has some errors. The number of wrong codewords is more than eight in a certain block. The restored QR code will not be decoded by the standard decoder. When the number is less than six, the number of incorrect codewords is greater than the capacity of ECC. A standard decoder cannot decode this recovered QR code.

**Table 4.** The decoding test of the restored image.

The QR Code	Correct Decoding with a Standard Decoder
Figure 4h	yes
Figure 4i	no
Figure 4j	no
Figure 4k	no
Figure 4l	no



**Figure 4.** (a):  $\mathbf{H}$ ; (b):  $\mathbf{S}_1$ ; (c):  $\mathbf{S}_2$ ; (d):  $\mathbf{S}_3$ ; (e):  $\mathbf{S}_4$ ; (f):  $\mathbf{S}_5$ ; (g):  $\mathbf{S}_6$ ; (h): Stacking ( $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4, \mathbf{S}_5, \mathbf{S}_6$ ); (i): Stacking ( $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4, \mathbf{S}_6$ ); (j): Stacking ( $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_6$ ); (k): Stacking ( $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_6$ ); (l): Stacking ( $\mathbf{S}_1, \mathbf{S}_6$ ); (m): The same between  $\mathbf{H}$  and Figure 4h (R), where black modules denote the same; (n): The difference between Figure 4a,i.

The proposed scheme can fully reconstruct the QR code. It is an  $(n, n)$ -VCS. We stack fewer than  $n$  shares to recover the image without decoding it (similar to a QR code). The error exceeds the capacity of ECC. The reconstructed QR code is a useless image. The secret will not be revealed. The restored image does not have helpful information from HVS. These experiments show that the proposed VCS is relatively safe.

**Table 5.** The number of the incorrect codeword.

Operation	Number of Wrong Codewords (Block 1, Block 2, Block 3, Block 4)
Stacking ( $\mathbf{S}_1, \dots, \mathbf{S}_6$ )	(0, 0, 0, 0)
Stacking ( $\mathbf{S}_1, \dots, \mathbf{S}_4, \mathbf{S}_6$ )	(0, 13, 0, 0)
Stacking ( $\mathbf{S}_1, \dots, \mathbf{S}_3, \mathbf{S}_6$ )	(0, 13, 0, 12)
Stacking ( $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_6$ )	(0, 25, 0, 12)
Stacking ( $\mathbf{S}_1, \mathbf{S}_6$ )	(0, 5, 25, 12)

Figure 4i and the secondary data in Table 5 (Stacking ( $S_1, \dots, S_4, S_6, \dots$ ) show that the proposed scheme is relatively safe. Figure 4i can be observed as resembling a QR code. It has more wrong codewords than can be corrected in block 2 (13 codewords are wrong in Table 5). This wrong QR code can not be decoded using the decoder. The secret will not be revealed. It is hard to determine what areas are wrong in Figure 4i except for the finder pattern. Therefore, it is hard to do brute force attacks to reveal the secret.

This paper designs an  $(n, n)$ -VCS for the QR code. When  $n$  shares are used to restore the secret, the secret QR code can be recovered losslessly. This can be decoded by a standard decoder. If  $n - 1$  or fewer shares are used to recover the secret, the restored QR code will have errors. A few wrong modules cause many codeword errors. The recovered QR code cannot be decoded by the standard decoder. The proposed scheme is relatively secure. This recovered QR code is deserved by HVS. However, it cannot be decoded and will not reveal anything.

Figure 4, Tables 4, and 5 show that all shares can recover the QR code without errors. The restored QR code with  $n - 1$  shares cannot be decoded by the standard decoder. These results prove that the proposed scheme is feasible and secure.

#### 4.2. Comparison of Different Schemes

The proposed VCS is non-pixel-expansive. It can completely reconstruct QR codes. Table 6 introduces the comparison result. Compared with the proposed VCS, this is a pixel-expansive scheme [41]. The two schemes belong to two non-pixel-expansive VCSs [42,43]. However, they can recover the secret image with errors. Compared with these two VCSs, the proposed VCS cannot reconstruct a QR code losslessly. Wan et al. used a QR code to share a secret QR code [45]. The secret QR code version met certain conditions. This scheme does not apply to all versions of QR codes. The proposed scheme can encrypt all versions of QR codes. This is a non-pixel-expansive scheme [46]. It can produce meaningful shares. Compared to the proposed VCS, the restored image has some errors. Compared with the pixel-expansive scheme ([41]), the proposed VCS has low time complexity.

**Table 6.** The comparison result.

	Pixel Expansion	Meaningful Shares	Recovered Image with Errors
[41]	yes	no	no
[42]	no	yes	yes
[43]	no	yes	yes
[45]	yes	yes	yes
[46]	no	yes	yes
Proposed scheme	no	no	no

The proposed scheme has some advantages. It can reconstruct a QR code losslessly. Moreover, its size is the same in both secret images and shares. This paper obtains a VCS for a QR code without pixel expansion. However, the proposed VCS has the shortcoming that the generated shares are meaningless images.

The scheme belongs to a  $(2, 2)$ -threshold scheme [41]. It can recover a QR code without errors. It is hard to extend the  $(n, n)$ -threshold scheme. These schemes are  $(n, n)$  schemes [42,45,46]. However, they cannot restore the secret QR code with errors. To solve the above disadvantages, this paper proposes a new VCS for the QR code. The proposed scheme can recover the QR code without errors. It belongs to a  $(2, 2)$  scheme. This scheme is a scheme without pixel expansion. The time complexity is  $\mathcal{O}(2D \times 2D)$  in [41]. The time complexity of the proposed scheme is  $\mathcal{O}(D \times D)$ . The proposed scheme is lower than Fang's scheme in terms of time complexity.

#### 4.3. Analysis and Discussion

This paper uses the mechanism of the ECC to design a VCS for a QR code. A secret QR code can be recovered without errors using all shares. The proposed scheme is an  $(n, n)$ -VCS. It will be improved to design a  $(k, n)$ -VCS in the future. Moreover, the proposed scheme generates meaningless shares. That would be difficult for a receiver and sender to manage. In the future, the authors will design a new VCS that can generate meaningful shares for a QR code.

#### 5. Conclusions

Using the characteristics of the ECC, this paper proposes a method for regional partitioning. A QR code is divided into  $n$  areas. These areas are distributed to  $n$  shares in a non-pixel-expansible method. If a share is missing, many wrong codewords will exceed the amount that the ECC can correct. A recovered QR code is not decoded, and the secret will not be revealed. Stacking all the shares can completely reconstruct the QR code, and a standard decoder can decode it. The capacity of the ECC will not be sacrificed. The proposed VCS is an  $(n, n)$ -VCS, which will be modified to a  $(k, n)$ -VCS later. The share is meaningless in this paper. In the future, a new scheme will generate meaningful shares. The proposed scheme is not resistant to outside attacks. A scheme that can resist external attacks will also be designed.

**Author Contributions:** Conceptualization, T.L. and B.Y.; software, T.L.; Formal analysis, T.L., B.Y., H.-M.Y., S.-C.C. and J.-S.P.; Methodology, T.L., B.Y. and J.-S.P.; Writing—original draft, T.L.; Writing—review and editing, T.L., B.Y., H.-M.Y., S.-C.C. and J.-S.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used to support the findings of this study are included in the article.

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

#### References

1. Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243. [[CrossRef](#)]
2. Saha, R.; Kumar, G.; Conti, M.; Devgun, T.; Kim, T.h.; Alazab, M.; Thomas, R. DHACS: Smart Contract-Based Decentralized Hybrid Access Control for Industrial Internet-of-Things. *IEEE Trans. Ind. Inform.* **2021**, *18*, 3452–3461. [[CrossRef](#)]
3. Kumar, A.; Saha, R.; Conti, M.; Kumar, G.; Buchanan, W.J.; Kim, T.H. A comprehensive survey of authentication methods in Internet-of-Things and its conjunctions. *J. Netw. Comput. Appl.* **2022**, *204*, 103414. [[CrossRef](#)]
4. Nowroozi, E.; Mohammadi, M.; Savas, E.; Conti, M.; Mekdad, Y. SPRITZ-1.5 C: Employing Deep Ensemble Learning for Improving the Security of Computer Networks against Adversarial Attacks. *arXiv* **2022**, arXiv:2209.12195.
5. Weng, S.; Chen, Y.; Hong, W.; Pan, J.S.; Chang, C.C.; Liu, Y. An improved integer transform combining with an irregular block partition. *Symmetry* **2019**, *11*, 49. [[CrossRef](#)]
6. Tiwari, S. An introduction to QR code technology. In Proceedings of the 2016 International Conference on Information Technology (ICIT), Bhubaneswar, India, 22–24 December 2016; pp. 39–44. [[CrossRef](#)]
7. Shin, S.; Gu, G. CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?). In Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP), Austin, TX, USA, 30 October–2 November 2012; pp. 1–6. [[CrossRef](#)]
8. Wang, K.C.; Brooks, R.R.; Barrineau, G.; Oakley, J.; Yu, L.; Wang, Q. Internet security liberated via software defined exchanges. In Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, Tempe, AZ, USA, 21 March 2018; pp. 19–22. [[CrossRef](#)]
9. Tewari, A.; Gupta, B.B. Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework. *Future Gener. Comput. Syst.* **2020**, *108*, 909–920. [[CrossRef](#)]
10. Song, P.C.; Chu, S.C.; Pan, J.S.; Wu, T.Y. An adaptive stochastic central force optimisation algorithm for node localisation in wireless sensor networks. *Int. J. Hoc Ubiquitous Comput.* **2022**, *39*, 1–19. [[CrossRef](#)]

11. Feng, Q.; He, D.; Zeadally, S.; Khan, M.K.; Kumar, N. A survey on privacy protection in blockchain system. *J. Netw. Comput. Appl.* **2019**, *126*, 45–58. [[CrossRef](#)]
12. Wu, T.Y.; Lee, Z.; Obaidat, M.S.; Kumari, S.; Kumar, S.; Chen, C.M. An authenticated key exchange protocol for multi-server architecture in 5G networks. *IEEE Access* **2020**, *8*, 28096–28108. [[CrossRef](#)]
13. Wu, T.Y.; Yang, L.; Lee, Z.; Chen, C.M.; Pan, J.S.; Islam, S. Improved ECC-based three-factor multiserver authentication scheme. *Secur. Commun. Netw.* **2021**, *2021*, 1–14. [[CrossRef](#)]
14. Tsai, T.T.; Huang, S.S.; Tseng, Y.M.; Chuang, Y.H.; Hung, Y.H. Leakage-Resilient Certificate-Based Authenticated Key Exchange Protocol. *IEEE Open J. Comput. Soc.* **2022**, *3*, 137–148. [[CrossRef](#)]
15. Huang, H.C.; Chu, S.C.; Pan, J.S.; Huang, C.Y.; Liao, B.Y. Tabu search based multi-watermarks embedding algorithm with multiple description coding. *Inf. Sci.* **2011**, *181*, 3379–3396. [[CrossRef](#)]
16. Weng, C.J.; Pan, J.S.; Liu, S.J.; Wang, M.J. A Watermarking method for printed QR code based on module expansion. In Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Sendai, Japan, 26–28 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 124–133. [[CrossRef](#)]
17. Pan, J.S.; Sun, X.X.; Chu, S.C.; Abraham, A.; Yan, B. Digital watermarking with improved SMS applied for QR code. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104049. [[CrossRef](#)]
18. Lukas, N.; Jiang, E.; Li, X.; Kerschbaum, F. Sok: How robust is image classification deep neural network watermarking? In Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–26 May 2022; pp. 787–804. [[CrossRef](#)]
19. Mustafa, E.M.; Elshafey, M.A.; Fouad, M.M. Enhancing CNN-based Image Steganalysis on GPUs. *J. Inf. Hiding Multim. Signal Process.* **2020**, *11*, 138–150.
20. Weng, S.; Tan, W.; Ou, B.; Pan, J.S. Reversible data hiding method for multi-histogram point selection based on improved crisscross optimization algorithm. *Inf. Sci.* **2021**, *549*, 13–33. [[CrossRef](#)]
21. Li, M.; Shi, W.; Zhu, F.; Tian, Z.; Shafiq, M.; Luo, Z.; Shan, C. Large-Capacity Local Multi-Dimensional Information Hiding Method for 6G Networks. *IEEE Netw.* **2022**, *36*, 160–165. [[CrossRef](#)]
22. Chang, C.C.; Chiu, Y.P.; Lin, C.C.; Chen, Y.H. Distortion Free Progressive BTC based Secret Image Sharing. *J. Netw. Intell.* **2018**, *3*, 195–213.
23. Yan, B.; Xiang, Y.; Hua, G. Improving the visual quality of size-invariant visual cryptography for grayscale images: An analysis-by-synthesis (AbS) approach. *IEEE Trans. Image Process.* **2018**, *28*, 896–911. [[CrossRef](#)]
24. Chen, C.C.; Tsai, Y.H. An Expandable Essential Secret Image Sharing Structure. *J. Inf. Hiding Multim. Signal Process.* **2016**, *7*, 135–144.
25. Yan, X.; Liu, X.; Yang, C.N. An enhanced threshold visual secret sharing based on random grids. *J. Real-Time Image Process.* **2018**, *14*, 61–73. [[CrossRef](#)]
26. Wu, X.; An, N.; Xu, Z. Sharing multiple secrets in XOR-based visual cryptography by non-monotonic threshold property. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *33*, 88–103. [[CrossRef](#)]
27. Ateniese, G.; Blundo, C.; De Santis, A.; Stinson, D.R. Visual cryptography for general access structures. *Inf. Comput.* **1996**, *129*, 86–106. [[CrossRef](#)]
28. Droste, S. New results on visual cryptography. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 1996; Springer: Berlin/Heidelberg, Germany, 1996; pp. 401–415. [[CrossRef](#)]
29. Zhou, Z.; Arce, G.R.; Di Crescenzo, G. Halftone visual cryptography. *IEEE Trans. Image Process.* **2006**, *15*, 2441–2453. [[CrossRef](#)] [[PubMed](#)]
30. Wu, X.; Yang, C.N. Probabilistic color visual cryptography schemes for black and white secret images. *J. Vis. Communun. Image Represent.* **2020**, *70*, 102793. [[CrossRef](#)]
31. Naor, M.; Shamir, A. Visual cryptography. In Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, 9–12 May 1994; Springer: Berlin/Heidelberg, Germany, 1994; pp. 1–12. [[CrossRef](#)]
32. Liu, F.; Wu, C. Embedded extended visual cryptography schemes. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 307–322. [[CrossRef](#)]
33. Hou, Y.C. Visual cryptography for color images. *Pattern Recognit.* **2003**, *36*, 1619–1629. [[CrossRef](#)]
34. Shyu, S.J. Image encryption by random grids. *Pattern Recognit.* **2007**, *40*, 1014–1031. [[CrossRef](#)]
35. Wang, Z.; Arce, G.R.; Di Crescenzo, G. Halftone visual cryptography via error diffusion. *IEEE Trans. Inf. Forensics Secur.* **2009**, *4*, 383–396. [[CrossRef](#)]
36. Luo, H.; Chen, H.; Shang, Y.; Zhao, Z.; Zhang, Y. Color transfer in visual cryptography. *Measurement* **2014**, *51*, 81–90. [[CrossRef](#)]
37. Liu, Z.; Zhu, G.; Wang, Y.G.; Yang, J.; Kwong, S. A novel (t, s, k, n)-threshold visual secret sharing scheme based on access structure partition. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2020**, *16*, 1–21. [[CrossRef](#)]
38. Yan, X.; Liu, L.; Li, L.; Lu, Y. Robust secret image sharing resistant to noise in shares. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2021**, *17*, 1–22. [[CrossRef](#)]
39. Liu, Z.; Zhu, G.; Ding, F.; Luo, X.; Kwong, S.; Li, P. Contrast-Enhanced Color Visual Cryptography for (k, n) Threshold Schemes. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2022**, *18*, 1–16. [[CrossRef](#)]
40. Jiao, S.; Feng, J.; Gao, Y.; Lei, T.; Yuan, X. Visual cryptography in single-pixel imaging. *Opt. Express* **2020**, *28*, 7301–7313. [[CrossRef](#)]

41. Fang, W.P. Offline QR code authorization based on visual cryptography. In Proceedings of the 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Dalian, China, 14–16 October 2011; pp. 89–92. [[CrossRef](#)]
42. Chow, Y.W.; Susilo, W.; Yang, G.; Phillips, J.G.; Pranata, I.; Barmawi, A.M. Exploiting the error correction mechanism in QR codes for secret sharing. In Proceedings of the Australasian Conference on Information Security and Privacy, Melbourne, VIC, Australia, 4–6 July 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 409–425. [[CrossRef](#)]
43. Cheng, Y.; Fu, Z.; Yu, B. Improved visual secret sharing scheme for QR code applications. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2393–2403. [[CrossRef](#)]
44. Tan, L.; Liu, K.; Yan, X.; Wan, S.; Chen, J.; Chang, C. Visual secret sharing scheme for color QR code. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 961–965. [[CrossRef](#)]
45. Wan, S.; Qi, L.; Yang, G.; Lu, Y.; Yan, X.; Li, L. Visual secret sharing scheme with  $(n, n)$  threshold for selective secret content based on QR codes. *Multimed. Tools Appl.* **2020**, *79*, 2789–2811. [[CrossRef](#)]
46. Huang, P.C.; Chang, C.C.; Li, Y.H.; Liu, Y. Enhanced  $(n, n)$ -threshold QR code secret sharing scheme based on error correction mechanism. *J. Inf. Secur. Appl.* **2021**, *58*, 102719. [[CrossRef](#)]
47. DENSO WAVE. QR Code.com. 2003. Available online: <http://www.qrcode.com/en/> (accessed on 18 March 2003).
48. ISO/IEC 18004:2006; Information Technology—Automatic Identification and Data Capture Techniques—QR Code 2005 Bar Code Symbology Specification. BS ISO/IEC: Geneva, Switzerland, 2006.
49. Github. Zxing Library. 2021. Available online: <https://github.com/zxing/zxing> (accessed on 30 September 2015).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.