

Chapter 8

Secure Color Imaging

RASTISLAV LUKAC and KONSTANTINOS N. PLATANIOTIS

8.1 Introduction

In digital imaging, visual data is massively accessed, distributed, manipulated and stored using communication and multimedia technology. To prevent unauthorized access, illegal copying and distribution, modern communication and multimedia systems utilize digital rights management (DRM) solutions to ensure media integrity, secure its transmission over untrusted communication channels and protect intellectual property rights [1], [2], [3]. Two fundamental DRM frameworks, i.e. watermarking [4] and encryption [5], have been suggested for protecting and enforcing the rights associated with the use of digital content [6].

Watermarking technologies are used for tasks such as identification of the content origin, copy protection, tracing illegal copies, fingerprinting, and disabling unauthorized access to content [7]. The image watermarking process embeds data, the so-called watermark, into the host image. Basically, watermarking can be performed in the spatial or frequency domain of the host image and the visual content can be protected by embedding visible or imperceptible watermarks [1]. Examples of color watermarking solutions which operate on different principles can be found in [8], [9], [10], [11]. Essential secure characteristics can be obtained by additive, multiplicative or quantization embedding. The watermark should be robust to various attacks and attempts for its removal, damage or unauthorized detection. After the transmission of watermarked images, the watermark is extracted using the secret key or blind extraction techniques. Note that most watermarking techniques are symmetric, i.e. the embedding and detection key are identical.

Encryption technologies ensure protection by scrambling the visual data into unrecognizable and meaningless variants [7], [12], [13]. In general, this transformation should be reversible in order to allow for

the perfect recovery of the original content using the secret key. Thus, the security of the encryption solution depends on the secrecy of the encryption and decryption keys. Once the encrypted data has been decrypted, encryption techniques do not offer any protection. To reduce computational overhead, popular image encryption solutions usually perform partial or selective encryption to protect the most important parts of the visual material [14], [15]. Most of partial encryption solutions are secure coders which combine encryption and image coding to overcome the redundancy in the visual data and secure the confidentiality of compressed data by encrypting only a fraction of the total image data. The most significant portion of the data, as dictated by a compression algorithm, is encrypted to disallow decoding without the knowledge of the decryption key. Similarly to the watermarking paradigm, secure characteristics can be obtained by encrypting the visual data in the spatial or frequency domain [16]. Efficient solutions for secure coding of color images can be found in [17], [18], [19], [20].

Apart from the above DRM paradigms, secret sharing schemes have been shown to be sufficiently secure in order to facilitate distributed trust and shared control in various communication applications, such as key management, conditional access, message authentication and content encryption [21], [22], [23], [24]. Due to the proliferation of imaging-enabled consumer electronic devices and the extensive use of digital imaging technologies in networked solutions and services, secret sharing concepts have a great potential to accomplish DRM features for securing the transmission and distribution of personal digital photographs and digital document images in public environments. This makes the secret sharing framework an excellent candidate for filling the gap between watermarking and encryption paradigms in secure imaging applications.

This chapter focuses on visual data protection using secret sharing concepts. Two main frameworks which use either the human visual system or simple logical operations to recover the secret image from the available shares are surveyed in a systematic and comprehensive manner. The presented methods can encrypt the secret image using an array of the existing threshold configurations, thus offering different design and application characteristics.

Section 8.2 starts by surveying the fundamentals of cryptographic solutions based on visual secret sharing or visual cryptography. Encryption and decryption functions are introduced and commented upon, and encryption of natural color images using halftoning and color mixing concepts is discussed. The implication of cost-effective decryption on the visual quality of the decrypted images is demonstrated.

Section 8.3 is devoted to image secret sharing with perfect reconstruction of the original visual data. The framework encrypts the decomposed bit-levels of the secret color image. In the input-agnostic processing mode, the framework produces image shares with representations identical to that of the secret image. Since in practice the end-user may request an increased or reduced pixel depth representation, input-specific solutions can be used to alter the level of protection and computational efficiency. Due to the symmetry between the encryption and decryption function, when the threshold constraint is satisfied during decryption

the framework perfectly reconstructs the input image data.

Section 8.4 introduces a cost-effective variant of the image secret sharing framework. The solution reduces the encryption and decryption operations from the block level to the pixel level, thus allowing significant computational and memory savings and efficient transmission of the shares in public networks. Reducing the number of shares to only two pieces, a private-key cryptosystem is obtained. Because of the symmetry constraint imposed on the encryption and decryption process, the solution satisfies the perfect reconstruction property. This section also includes the discussion on selective encryption, in terms of both bit-levels or color channels of the secret color image.

The chapter concludes with Section 8.5. The section summarizes the ideas behind secret sharing of visual data. The definitions and some properties of the most popular secret sharing configurations are listed in the chapter's appendix.

8.2 Visual Secret Sharing of Color Images

Secret sharing is considered a cost-effective solution which can be used to secure transmission and distribution of visual digital material over untrusted public networks. Most of the existing secret sharing schemes are generalized within the so-called $\{k, n\}$ -threshold framework which confidentially divides the content of a secret message into n shares in the way which requires the presence of at least k , for $k \leq n$, shares for the secret message reconstruction [21], [22]. Thus, the framework can use any of $n!/(k!(n-k)!)$ possible combinations of k shares to recover the secret message, whereas the use of $k - 1$ shares should not reveal the secret message. Amongst numerous possible $\{k, n\}$ configurations, the simplest case is constituted by $\{2, 2\}$ -schemes which are commonly used as a private-key cryptosystem solution [25], [26].

The $\{k, n\}$ -threshold framework has been popularized in the image processing community through visual secret sharing (VSS) or visual cryptography [27], [28], [29]. VSS schemes (Figure 8.1) encrypt the binary or binarized visual data — the so-called secret image — into the shares with the same data representation and uses properties of the human visual system (HVS) to force the recognition of a secret image from available shares. Since each binary data can be displayed either as frosted (for 0 values) or transparent (for 1 values) when printed on transparencies or viewed on the screen, overlapping shares which contain seemingly random information can reveal the secret image without additional computations or any knowledge of cryptographic keys. This makes the approach attractive for various applications. For instance, visual cryptography concepts were recently extended to enhance image watermarking solutions [30], [31], [32].

8.2.1 Visual Cryptography Fundamentals

In the conventional VSS framework [Figure 8.1(a)], the secret image is a $K_1 \times K_2$ binary image $I : Z^2 \rightarrow \{0, 1\}$ with values $I_{(r,s)}$ occupying the pixel locations (r, s) , for $r = 1, 2, \dots, K_1$ and $s = 1, 2, \dots, K_2$. Using a $\{k, n\}$ -scheme operating on I , each binary pixel $I_{(r,s)}$ is encrypted via an encryption function [27]:

$$f_e(I_{(r,s)}) = \begin{cases} [s^{(1)}, s^{(2)}, \dots, s^{(n)}]^T \in C_0 & \text{for } I_{(r,s)} = 0 \\ [s^{(1)}, s^{(2)}, \dots, s^{(n)}]^T \in C_1 & \text{for } I_{(r,s)} = 1 \end{cases} \quad (8.1)$$

to produce a $m_1 \times m_2$ block $s^{(l)} = \{s_{(m_1(r-1)+1, m_2(s-1)+1)}, s_{(m_1(r-1)+1, m_2(s-1)+2)}, \dots, s_{(m_1 r, m_2 s)}\} \in S^{(l)}$, for $l = 1, 2, \dots, n$, of binary values in each of n binary shares $S^{(1)}, S^{(2)}, \dots, S^{(n)}$. The spatial arrangement of bits in $s^{(l)}$ varies from block to block depending on the value of $I_{(r,s)}$ to be encrypted and the choice (usually guided by a random number generator) of the matrix $[s^{(1)}, s^{(2)}, \dots, s^{(n)}]^T$ from the matrices' set C_0 or C_1 . The sets C_0 or C_1 include all matrices obtained by permuting the columns of $n \times m_1 m_2$ basis binary matrices A_0 or A_1 , respectively. The value $m_1 m_2$ is the so-called expansion factor and therefore, the basis matrices are constructed in the way to minimize the expansion factor as much as possible. For example, in $\{2, 2\}$ -VSS configurations the use of 2×2 basis matrices

$$A_0 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad (8.2)$$

implies the following (Figure 8.2):

$$C_0 = \left\{ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \right\} \quad (8.3)$$

$$C_1 = \left\{ \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \right\} \quad (8.4)$$

Repeating (8.1) for $\forall(r, s)$ encrypts the secret image I into the shares $S^{(1)}, S^{(2)}, \dots, S^{(n)}$ with dimensions of $m_1 K_1 \times m_2 K_2$ pixels. The reader can find the definition of basis matrices for other most commonly used $\{k, n\}$ configurations in the Appendix.

In standard practice, VSS allows for visual recovery of the encrypted images by simply stacking the shares and visually inspecting the resulting message, a feature that makes the operation cost-effective [27], [33], [34]. To understand the reconstruction of the secret image from the shares, VSS decryption can be modelled through the following decryption function [35]:

$$I'_{(u,v)} = f_d(\{s_{(u,v)}^{(l)}; l = 1, 2, \dots, \zeta\}) = \begin{cases} 1 & \text{if } \forall s_{(u,v)}^{(l)} = 1 \\ 0 & \text{if } \exists s_{(u,v)}^{(l)} = 0 \end{cases} \quad (8.5)$$

where $u = 1, 2, \dots, m_1 K_1$ and $v = 1, 2, \dots, m_2 K_2$. The parameter ζ denotes the number of available shares, i.e. $\zeta \leq n$. Due to the utilization of the transparent/frosted concept in (8.1), the VSS decryption process (8.5) recovers the decrypted pixel $I'_{(u,v)}$ as:

- black ($I'_{(u,v)} = 0$) if any of the share pixels $\{s_{(u,v)}^{(l)}, l = 1, 2, \dots, \zeta\}$ corresponding to the same spatial location (u, v) is frosted, or
- white ($I'_{(u,v)} = 1$) if all the share pixels $\{s_{(u,v)}^{(l)}, l = 1, 2, \dots, \zeta\}$ corresponding to (u, v) in the available shares are transparent.

Due to the expansion properties of VSS schemes, the original pixel $I_{(r,s)}$ is transformed by the VSS encryption/decryption process to a $m_1 \times m_2$ block of decrypted pixels:

$$f_d(f_e(I_{(r,s)})) = \{I'_{(m_1(r-1)+1, m_2(s-1)+1)}, I'_{(m_1(r-1)+1, m_2(s-1)+2)}, \dots, I'_{(m_1r, m_2s)}\} \quad (8.6)$$

Through the construction of basis matrices, $\{k, n\}$ -VSS schemes obtain the essential secure characteristics via the contrast properties of decrypted blocks. Since pixels in small spatial neighborhoods are perceived by HVS as a single pixel with the intensity averaged over its neighbors [36], [37], the contrast of the decrypted block $f_d(f_e(I_{(r,s)}))$ can be modelled as $\sum f_d(f_e(I_{(r,s)})) / (m_1 m_2)$. If $\zeta < k$, then the contrast properties of decrypted blocks corresponding to $I_{(r,s)} = 0$ and $I_{(r,s)} = 1$ should be identical. The meaningful information — modelled via the different spatial contrast — can be visually revealed only if $\zeta \geq k$. This forms the following constraint:

$$\begin{aligned} \sum f_d(f_e(0)) &= \sum f_d(f_e(1)) & \text{if } \zeta < k \\ \sum f_d(f_e(0)) &\neq \sum f_d(f_e(1)) & \text{if } \zeta \geq k \end{aligned} \quad (8.7)$$

The graphical interpretation of the matrices listed in (8.3) and (8.4) is given in Figure 8.2. The figure also depicts the decrypted blocks obtained by stacking the share blocks. If only one arbitrary share block is used for the decryption, the spatial contrast of $f_d(f_e(I_{(r,s)}))$ is equal to $1/2$ for both $I_{(r,s)} = 0$ and $I_{(r,s)} = 1$. However, if both shares — as required by the $\{2, 2\}$ -threshold scheme — are available, then the decrypted block $f_d(f_e(I_{(r,s)}))$ has the spatial contrast equal to 0 for $I_{(r,s)} = 0$ and $1/2$ for $I_{(r,s)} = 1$. Note that similar observations can be made for all $\{k, n\}$ -threshold configurations listed in Appendix. Due to the construction of basis matrices, the blocks corresponding to white secret pixels ($I_{(r,s)} = 1$) are recognized as some level of gray, but never white. Similarly, many $\{k, n\}$ -threshold configurations with $k < n$ do not restore the decrypted blocks corresponding to black secret pixels ($I_{(r,s)} = 0$) as purely black. Therefore, a visually decrypted image has shifted intensity (typically darker) compared to the secret input image. An example generated using the $\{2, 2\}$ -VSS scheme is shown in Figure 8.3.

8.2.2 Color Visual Cryptography

Given the binary nature of VSS encryption, the application of a conventional $\{k, n\}$ -VSS solution to a $K_1 \times K_2$ natural, continuous-tone, gray-scale and color images requires their binarization using halftoning [36], [38], [39]. As described in Chapter 7, the halftoning process transforms the input image into a $K_1 \times K_2$

halftone image by using the density of the net dots to simulate the gray or color levels. Note that there are many ways to obtain halftone images and $\{k, n\}$ -VSS solutions can work with all of them. In this work, a simple error-diffusion procedure based on the Floyd-Steinberg filter with the weights [40]:

$$\begin{bmatrix} w_{(r-1,s-1)} & w_{(r-1,s)} & w_{(r-1,s+1)} \\ w_{(r,s-1)} & w_{(r,s)} & w_{(r,s+1)} \\ w_{(r+1,s-1)} & w_{(r+1,s)} & w_{(r+1,s+1)} \end{bmatrix} = \frac{1}{16} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 7 \\ 3 & 5 & 1 \end{bmatrix} \quad (8.8)$$

is used to demonstrate the concept and produce the $I_{(r,s)}$ data suitable for VSS encryption.

Following the application scenario shown in Figure 8.1(b), the input color image [Figure 8.4(a)] is halftoned to reduce the depth of the image representation. Applying the VSS encryption procedure of (8.1) in a component-wise manner to the color halftone image [Figure 8.4(b)], shares — such as those shown in Figures 8.4(c) and (d) for the $\{2, 2\}$ -VSS configuration — can be produced. Figure 8.4(e) depicts the result of stacking two shares together using (8.5). The decrypted color image has reduced visual quality due to color shifts and modified contrast. Not surprisingly, the produced outcome has the familiar form of a halftone image.

Apart from the component-wise solutions, the color $\{k, n\}$ VSS schemes can be constructed using additive or subtractive color mixing principles [36], [37], [41], [42], [43]. In the additive model [Figure 8.5(a)], each color is modelled using Red (R), Green (G) and Blue (B) primaries. This concept is used in most computer monitors. On the other hand, color printers typically use the subtractive model [Figure 8.5(b)] with complementary Cyan (C), Magenta (M) and Yellow (Y) colors to obtain spectrally shifted colors. Additional information on the issue can be found in Chapter 1. By decomposing the color halftone image into its RGB or CMY channels and using either the additive or subtractive model to produce the share blocks, decrypted halftone color pixels are recognizable by HVS for $\zeta \leq k$ as an average color of the corresponding stacked color share blocks of $m_1 \times m_2$ pixels. Similarly to the component-wise VSS solutions, the decryption process deteriorates the visual quality of the output.

8.3 Perfect Reconstruction-Based Image Secret Sharing

To prevent the introduction of visual impairments into the decrypted image, a bit-level processing based image secret sharing (ISS) framework has been introduced (Figure 8.6) [35], [44]. Similarly to the VSS schemes, the ISS framework operates in any $\{k, n\}$ -threshold configuration (see Appendix) to encrypt the secret image into n shares. By operating on the bit-levels of the image array, the ISS framework: i) generates the shares with the enhanced protection, ii) allows for selective encryption of the decomposed bit-levels, and iii) recovers the original secret image when the required amount of shares are available for decryption.

8.3.1 Color Image Secret Sharing

A $K_1 \times K_2$ color image $\mathbf{x} : Z^2 \rightarrow Z^3$ represents a two-dimensional array of three-component vectorial inputs $\mathbf{x}_{(r,s)} = [x_{(r,s)1}, x_{(r,s)2}, x_{(r,s)3}]$ which occupy the spatial position (r, s) with coordinates $r = 1, 2, \dots, K_1$ and $s = 1, 2, \dots, K_2$. Components $x_{(r,s)c}$, for $c = 1, 2, 3$, represent the c -th elements of the color vector $\mathbf{x}_{(r,s)}$. In the case of a Red-Green-Blue (RGB) color image, $c = 1$ denotes the R component whereas $c = 2$ and $c = 3$ correspond to G and B component, respectively. Each color component $x_{(r,s)c}$ is coded with B bits allowing $x_{(r,s)c}$ to take an integer value between 0 and $2^B - 1$. In the case of RGB color images, $x_{(r,s)c}$ is coded by eight bits ($B = 8$) ranging the value of $x_{(r,s)c}$ from 0 to 255. Using a bit-level representation [45], the color vector $\mathbf{x}_{(r,s)}$ can be equivalently expressed in a binary form as follows [44]:

$$\mathbf{x}_{(r,s)} = \sum_{b=1}^B \mathbf{x}_{(r,s)}^b 2^{B-b} \quad (8.9)$$

where $\mathbf{x}_{(r,s)}^b = [x_{(r,s)1}^b, x_{(r,s)2}^b, x_{(r,s)3}^b] \in \{0, 1\}^3$ denotes the binary vector at the bit level b , with $b = 1$ denoting the most significant bit (MSB). Thus each binary component $x_{(r,s)c}^b \in \{0, 1\}$, for $c = 1, 2, 3$, is equal to 1 or 0 corresponding respectively to white and black in the binary representation.

Since components $x_{(r,s)c}^b$ represent binary data, they are ideal for VSS-like encryption. Using the basis matrices of the conventional $\{k, n\}$ -VSS threshold schemes, each $x_{(r,s)c}^b$ can be encrypted as follows:

$$f_e(x_{(r,s)c}^b) = \begin{cases} [s_c^{(1)b}, s_c^{(2)b}, \dots, s_c^{(n)b}]^T \in C_0 & \text{for } x_{(r,s)c}^b = 0 \\ [s_c^{(1)b}, s_c^{(2)b}, \dots, s_c^{(n)b}]^T \in C_1 & \text{for } x_{(r,s)c}^b = 1 \end{cases} \quad (8.10)$$

The set $s_c^{(l)b} = \{s_{(m_1(i-1)+1, m_2(j-1)+1)c}^b, s_{(m_1(i-1)+1, m_2(j-1)+2)c}^b, \dots, s_{(m_1 i, m_2 j)c}^b\} \in S_c^{(l)b}$, for $l = 1, 2, \dots, n$, denotes a $m_1 \times m_2$ block of binary values in each of n binary shares $S_c^{(1)b}, S_c^{(2)b}, \dots, S_c^{(n)b}$ which are generated for the particular bit-level b and the color-channel c .

Due to the use of random selection of $[s_c^{(1)b}, s_c^{(2)b}, \dots, s_c^{(n)b}]^T$ from the matrices' set C_0 or C_1 , component-wise bit-levels $S_c^{(l)b}$ exhibit randomness. Thus, the process modifies both the spatial and spectral correlation between spatially neighboring binary share vectors $\mathbf{s}_{(u,v)}^{(l)b} = [s_{(u,v)1}^{(l)b}, s_{(u,v)2}^{(l)b}, s_{(u,v)3}^{(l)b}] \in S^{(l)b}$, for $u = 1, 2, \dots, m_1 K_1$ and $v = 1, 2, \dots, m_2 K_2$. By repeating the operation in (8.10) for all values of b and c , the generated share bits $s_{(u,v)c}^{(l)b}$ are used to obtain the full-color share vector

$$\mathbf{s}_{(u,v)}^{(l)} = \sum_{b=1}^B \mathbf{s}_{(u,v)}^{(l)b} 2^{B-b} \quad (8.11)$$

where $\mathbf{s}_{(u,v)}^{(l)} = [s_{(u,v)1}^{(l)}, s_{(u,v)2}^{(l)}, s_{(u,v)3}^{(l)}]$ consists of B -bit color components $s_{(u,v)1}^{(l)}, s_{(u,v)2}^{(l)}, s_{(u,v)3}^{(l)}$. Thus, the ISS encryption process splits the full-color secret image \mathbf{x} into seemingly random, full-color shares $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \dots, \mathbf{S}^{(n)}$ with a $m_1 K_1 \times m_2 K_2$ spatial resolution.

Unlike previously proposed VSS solutions, the ISS framework aims to restore the secret image in its original quality. Thus, the framework satisfies the so-called perfect reconstruction property which is considered essential in modern visual communication systems and imaging pipelines [35]. To recover the secret image with perfect reconstruction, encryption and decryption should be symmetric (Figure 8.6). The decryption process first decomposes the color vectors $\mathbf{s}_{(u,v)}^{(1)}, \mathbf{s}_{(u,v)}^{(2)}, \dots, \mathbf{s}_{(u,v)}^{(\zeta)}$ from ζ shares $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \dots, \mathbf{S}^{(\zeta)}$ which are available for decryption. Then, the decryption function [46]:

$$x_{(r,s)c}^b = f_d(\{s_c^{(l)b}; l = 1, 2, \dots, \zeta\}) = \begin{cases} 1 & \text{for } [s_c^{(1)b}, s_c^{(2)b}, \dots, s_c^{(\zeta)b}]^T \in C_0 \\ 0 & \text{for } [s_c^{(1)b}, s_c^{(2)b}, \dots, s_c^{(\zeta)b}]^T \in C_1 \end{cases} \quad (8.12)$$

is applied in the component-wise manner to the set of decomposed share blocks

$$s_c^{(l)b} = \{s_{(u,v)c}^{(l)b}, s_{(u,v+1)c}^{(l)b}, \dots, s_{(u+m_1-1, v+m_2-1)c}^{(l)b}\} \in S_c^{(l)b} \quad (8.13)$$

to recover the individual bits. The determination of the relationship between $\{s_c^{(l)b}, s_c^{(2)b}, \dots, s_c^{(\zeta)b}\} \subseteq \{s_c^{(l)b}, s_c^{(2)b}, \dots, s_c^{(n)b}\}$ for $\zeta \leq n$ and the matrices' sets C_0 and C_1 can be done using the contrast properties of the share blocks $s_c^{(l)b}, s_c^{(2)b}, \dots, s_c^{(\zeta)b}$ stacked together (Figure 8.2). Similarly to the $\{k, n\}$ -VSS schemes, the difference between the stacked blocks $[s_c^{(1)b}, s_c^{(2)b}, \dots, s_c^{(\zeta)b}]^T \in C_0$ and $[s_c^{(1)b}, s_c^{(2)b}, \dots, s_c^{(\zeta)b}]^T \in C_1$ in (8.12) reveals only if $\zeta \geq k$. In this case, the decryption process recovers the corresponding original bit $x_{(r,s)c}^b$ which can be equivalently expressed via the symmetry constraint of ISS encryption/decryption as follows:

$$f_d(f_e(x_{(r,s)c}^b)) = x_{(r,s)c}^b \quad (8.14)$$

By repeating (8.12) with (8.13) for all color channels $c = 1, 2, 3$, bit-levels $b = 1, 2, \dots, B$ and spatial locations $u = 1, 1 + m_1, 1 + 2m_1, \dots, 1 + (K_1 - 1)m_1$ and $v = 1, 1 + m_2, 1 + 2m_2, \dots, 1 + (K_2 - 1)m_2$, the procedure recovers the complete set of bits $x_{(r,s)c}^b$ in binary vectors $\mathbf{x}_{(r,s)}^b = [x_{(r,s)1}^b, x_{(r,s)2}^b, x_{(r,s)3}^b]$ used to represent the original color vector $\mathbf{x}_{(r,s)}$ in (8.9). Completing the bit-level stacking in (8.9) for $r = 1, 2, \dots, K_1$ and $s = 1, 2, \dots, K_2$ recovers the original full-color secret image \mathbf{x} with perfect reconstruction.

Figure 8.7 shows images recorded at different stages of the ISS processing chain. In Figure 8.7, the ISS shares [Figures 8.7(b) and (c)] follow the full-color representation of the original image [Figure 8.7(a)], thus offering better protection compared to the VSS shares shown in Figures 8.4(c) and (d). Moreover, unlike the VSS output shown in Figure 8.4(e), the ISS decrypted output shown in Figure 8.7(d) is perfectly restored in terms of both resolution and color/structural content.

8.3.2 Secret Sharing Solutions for Various Image Formats

Following the nature of the visual data to be encrypted, the ISS framework constitutes an input-agnostic solution [46] which produces shares with the bit representation identical to that of the secret image. In this

case, the ISS solution encrypts [35]: i) the binary secret image into the binary shares, ii) the gray-scale secret image into the gray-scale shares, or iii) the color secret image into the color shares. In all above cases, the decryption process recovers the secret image with perfect reconstruction.

Since the input-agnostic ISS framework can be directly applied to any B -bit image, it can be also used to process the binary image shown in Figure 8.8(a). Visual inspection of the results shown in Figures 8.8(b) and (c) and Figures 8.3(b) and (c) reveals that for the binary data both VSS and ISS frameworks produce equivalent shares. However, due to the different decryption concept, the ISS decrypted output [Figure 8.8(d)] is identical with the original image whereas the VSS output [Figure 8.3(d)] has an expanded size and suffer from various visual impairments.

The randomness of the encryption operations in (8.10) fortified by the depth of the B -bit representation of the secret image introduces significant variations between the original and share pixels. The degree of protection, obtained here through the depth of cryptographic noise generated by the ISS framework, increases with the number of bits used to represent the image pixel (Figure 8.9). Assuming that N denotes the number of unique matrices either in C_0 or C_1 , the B -bit color component $x_{(r,s)c}$ is encrypted using one of N^B unique share blocks of B -bit values instead of one of only N unique share blocks of binary values used in the traditional and halftoning-based VSS. It is not difficult to see that even for a simple $\{2, 2\}$ -ISS scheme with six ($N = 6$) matrices listed in (8.3) or (8.4), there exist 6^{24} unique full-color share blocks which can be used for encryption of color RGB vectors $\mathbf{x}_{(r,s)}$ with $B = 3 \times 8$. This suggests that the ISS framework can offer the higher protection of the visual data compared to the conventional VSS solutions.

In many practical applications, the user can request different protection levels during the encryption process and/or encrypt the visual data in the pre-determined format. This can be done using the input-specific ISS solutions [46]. As part of the processing pipeline, the input-specific solution can require to convert: i) the binary or gray-scale input image into the color image when the solution is color image specific to produce color shares, ii) the binary or color input image into the gray-scale image when the solution is gray-scale image specific to produce gray-scale shares, and iii) the color or gray-scale input image into the binary image when the solution is binary image specific to produce binary shares.

The input-specific paradigm requires format conversion such as the replication of the input (for binary-to-gray-scale, binary-to-color and gray-scale-to-color) or reduction of image representation (for color-to-gray-scale, color-to-binary, and gray-scale-to-binary) in order to meet the requirements for the input. Depending on the format conversion, the procedure requires to transmit more or less share information compared to the shares produced by the input-agnostic ISS solution. Note that inverse format conversion is necessary to recover the secret image. In the data-replication encryption mode, the decryption recovers the original image. In the data-reduction encryption mode, the procedure results in the approximated secret image due to the loss in input format conversion. The reader can find additional information on input-agnostic

and input-specific solutions in [46], [47].

Finally, it should be noted that both input-agnostic and input-specific ISS solutions can be used to process the secret image using an arbitrary $\{k, n\}$ -threshold configuration and expansion factor. Since expanded dimensionality of the shares suggests increased requirements for their transmission, the next Section will discuss the design of non-expandable ISS solution.

8.4 Cost-Effective Private-Key Solution

In practice, $\{2, 2\}$ -threshold configurations are often used as the private-key cryptosystem solution [25]. In this application scenario, each of the two generated shares serves to the other as the decryption key. However, encrypting the color image using basis matrices defined in (8.2) expands the spatial resolution of shares four-fold, i.e. the expansion factor is $m_1 m_2 = 4$. To reduce the complexity and allow for cost-effective transmission of the shares via public networks, the ISS solution (Figure 8.6) proposed in [26] encrypts each binary component of the decomposed original vectors with a single output bit instead of the usual block of $m_1 m_2$ bits (for $m_1 \geq 2$ and $m_2 \geq 2$). Since the solution admits the non-expansion factor $m_1 m_2 = 1$, the produced shares have the same spatial resolution as that of the original image (see Figure 8.10). This suggests that the encrypted visual information can be transmitted over untrusted channels at a reasonable cost (overhead).

Based on decomposed binary vectors $\mathbf{x}_{(r,s)}^b = [x_{(r,s)1}^b, x_{(r,s)2}^b, x_{(r,s)3}^b]$ obtained from the original color vector $\mathbf{x}_{(r,s)}$ in (8.9), the solution generates two binary share vectors $\mathbf{s}_{(r,s)}^{(l)b} = [s_{(r,s)1}^{(l)b}, s_{(r,s)2}^{(l)b}, s_{(r,s)3}^{(l)b}]$, for $l = 1, 2$, as follows [26]:

$$f_e(x_{(r,s)c}^b) = [s_{(r,s)c}^{(1)b} \ s_{(r,s)c}^{(2)b}]^T \in \begin{cases} \{[0 \ 1]^T, [1 \ 0]^T\} & \text{for } x_{(r,s)c}^b = 1 \\ \{[0 \ 0]^T, [1 \ 1]^T\} & \text{for } x_{(r,s)c}^b = 0 \end{cases} \quad (8.15)$$

where the binary sets $[s_{(r,s)c}^{(1)b} \ s_{(r,s)c}^{(2)b}]^T$ are obtained from the basis elements 0 and 1. For simulation purposes any conventional 'rand' programming routine, which implements a random number generator seeded using the computer system clock state, can be used in (8.15) to guide the encryption. However, solutions implemented in hardware may use electronic noise sources or radioactive decay [48]. The generated share bits are used in (8.11) to produce the full-color share vectors $\mathbf{s}_{(r,s)}^{(1)}$ and $\mathbf{s}_{(r,s)}^{(2)}$ located at spatial position (r, s) of the $K_1 \times K_2$ color shares $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$, respectively.

Although each binary component $x_{(r,s)c}^b$ is encrypted by one of only two different configurations, the formation of the binary vectors $\mathbf{s}_{(r,s)}^{(l)b}$ increases the degree of protection to 2^3 . Moreover, due to bit-level stacking in (8.11), the encryption process can generate the full-color shares from the set of 2^{3B} possible vectors. It is evident that the maximum confidentiality of the encrypted information can be obtained by repeating the encryption process (8.15) for $b = 1, 2, \dots, B$ and $c = 1, 2, 3$.

During decryption, the original color/structural information is recovered by processing the share vectors at the decomposed bit-level. The decryption procedure classifies the original binary components $x_{(r,s)c}^b$ under the constraint in (8.14) as follows [26]:

$$x_{(r,s)c}^b = \begin{cases} 0 & \text{for } s_{(r,s)c}^{(1)b} = s_{(r,s)c}^{(2)b} \\ 1 & \text{for } s_{(r,s)c}^{(1)b} \neq s_{(r,s)c}^{(2)b} \end{cases} \quad (8.16)$$

and recovers the original color vector $\mathbf{x}_{(r,s)}$ using (8.9). Due to the symmetry between (8.15) and (8.16), as indicated in (8.14), the solution satisfies the perfect reconstruction property (Figure 8.10).

Since the above approach holds the perfect reconstruction property and is non-expansive and easy to implement, it has been recently used to encrypt the metadata information in digital camera images [49]. In this way, the acquired images can be indexed directly in the capturing device by embedding metadata information using the simple $\{2, 2\}$ -scheme. The concept described in this section has also been extended in the JPEG domain to enable shared key image encryption for a variety of applications. The scheme proposed in [50] directly works on the quantized DCT coefficients and the shares are stored in the JPEG format. Following the symmetry of (8.15) and (8.16), the decryption process preserves the generated JPEG data.

To understand the importance of bit-level encryption of color images, Figure 8.11 allows for the visual comparison of the color shares when cryptographic processing is applied to a subset of binary levels. Applying the cryptographic operations for the MSB [Figure 8.11(a)] or the two most significant bits [Figure 8.11(b)] only, fine details are sufficiently encrypted, however, large flat regions can be partially visually revealed. As shown in Figure 8.11(c), a sufficient level of protection of the whole visual information is achieved by applying (8.15) to the first three most significant bits ($b = 1, 2, 3$). The remaining bits of the original image vectors can be simply copied into the shares unchanged.

Other important factor in color image encryption is the color information. Figures 8.12 and 8.13 depict share images generated when the encryption operations are selectively applied to the particular color channels. As it can be seen from the presented examples, encrypting either one (Figure 8.12) or two color channels (Figure 8.13) does not completely obscure the actual input. This suggests that for secure ISS encryption all the channels of the color RGB image should be encrypted.

8.5 Conclusion

Secret sharing technology was used in this chapter as the means of ensuring protection of color images intended for distribution over untrusted public networks. Using the popular $\{k, n\}$ -threshold framework, secret sharing solutions encrypt color images into n seemingly-random, noise-like, shares and recover the input image when at least k shares are available for decryption. The chapter provided an overview of the

$\{k, n\}$ -threshold solutions which decrypt the visual data using either the properties of the human visual system or simple logical operations.

In the first application scenario, the nature of visual secret sharing solutions requires that the color image to be encrypted should be transformed to a halftone image prior to its encryption. The generated shares are commonly printed on transparencies or viewed on the special screen. Due to the ability of the human visual system to sense small image neighborhoods by averaging color information over spatially neighboring pixels, overlapping k or more shares readily reveals the secret image without the need for additional computations or knowledge of cryptographic keys. However, the simplicity of decryption is obtained at expense of reduced visual quality of the decrypted color image.

However, the availability of decrypted images in quality and representation identical to that of the original is essential in modern visual communication and multimedia systems. For that reason, $\{k, n\}$ -threshold configurations were combined with bit-level processing and simple logical operations to provide perfect reconstruction of the original color input. Building on the bit representation of the secret image, the framework can be used to design various input-agnostic and input-specific image encryption tools. These $\{k, n\}$ image secret sharing solutions differ in their design characteristics and complexity and may secure the visual content at different protection levels and with different expansion or data reduction modes.

This overview suggests that secret sharing of color images constitutes a modern and powerful cryptographic tool which complements existing watermarking and encryption technology. It can be used to efficiently protect visual communication over untrusted public networks, and is well suited to support value-additive services for the next generation of applications, such as secure wireless video-conferencing, online collaboration, and secure distribution/sharing of digital image materials.

Appendix: Basis Matrices of Some Popular Threshold Configurations

Apart from the most popular $\{2, 2\}$ -threshold scheme with the basis matrices defined in (8.2), other popular $\{k, n\}$ -threshold configurations are used to encrypt the secret image into three, four, or more shares. For example, the basis matrices

$$A_0 = \begin{bmatrix} 1, 0, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1, 0, 0, 0 \\ 1, 0, 0, 0 \\ 1, 0, 0, 0 \end{bmatrix} \quad (8.17)$$

of the $\{2, 3\}$ scheme achieves the required secure characteristics by splitting the content of the input image into three shares. The use of any two ($k = 2$) of three ($n = 3$) generated shares produces the spatial contrast equal to 0 for secret zero bits and 1/4 for secret unit bits. The same spatial contrast of the stacked share blocks is obtained when the decryption is performed over all three shares generated using the following

$\{3, 3\}$ -threshold configuration:

$$A_0 = \begin{bmatrix} 0, 0, 1, 1 \\ 0, 1, 0, 1 \\ 0, 1, 1, 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1, 1, 0, 0 \\ 1, 0, 1, 0 \\ 1, 0, 0, 1 \end{bmatrix} \quad (8.18)$$

If the secret image is to be encrypted into four shares, three different $\{k, 4\}$ configurations (for $k = 2, 3, 4$) are possible. The $\{2, 4\}$ -threshold scheme:

$$A_0 = \begin{bmatrix} 1, 0, 0, 0 \\ 0, 1, 0, 0 \\ 0, 0, 1, 0 \\ 0, 0, 0, 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1, 0, 0, 0 \\ 1, 0, 0, 0 \\ 1, 0, 0, 0 \\ 1, 0, 0, 0 \end{bmatrix} \quad (8.19)$$

operates on 2×2 blocks and for two stacked shares has the spatial contrast properties similar to the configurations in (8.17) and (8.18). However, the construction of $\{3, 4\}$ and $\{4, 4\}$ basis matrices is more complex, necessitating 3×3 blocks to preserve the ratios $m_1 K_1 / (m_2 K_2)$ and K_1 / K_2 identical. Thus, the $\{3, 4\}$ -threshold scheme is defined using

$$A_0 = \begin{bmatrix} 0, 1, 1, 1, 1, 1, 1, 0, 0 \\ 0, 1, 1, 1, 1, 0, 0, 1, 1 \\ 0, 1, 1, 0, 0, 1, 1, 1, 1 \\ 0, 0, 0, 1, 1, 1, 1, 1, 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1, 1, 1, 1, 1, 1, 0, 0, 0 \\ 1, 1, 1, 1, 0, 0, 1, 1, 0 \\ 1, 1, 1, 0, 1, 0, 1, 0, 1 \\ 1, 1, 1, 0, 0, 1, 0, 1, 1 \end{bmatrix} \quad (8.20)$$

which implies that three stacked shares produce the spatial contrast which is equal to $2/9$ for secret zero bits and $1/3$ for secret unit bits. By stacking four shares in the $\{4, 4\}$ -scheme given by

$$A_0 = \begin{bmatrix} 1, 0, 0, 1, 0, 0, 1, 0, 1 \\ 1, 0, 1, 0, 0, 0, 1, 1, 0 \\ 1, 0, 1, 0, 0, 1, 0, 0, 1 \\ 0, 1, 1, 0, 0, 0, 1, 0, 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1, 0, 0, 0, 0, 0, 1, 1, 1 \\ 1, 0, 1, 0, 0, 1, 1, 0, 0 \\ 1, 1, 0, 0, 0, 1, 0, 1, 0 \\ 1, 1, 1, 0, 0, 0, 0, 0, 1 \end{bmatrix} \quad (8.21)$$

the secret zero and unity bits are respectively represented in the stacked shares by spatial contrast values 0 and $1/9$.

Finally, the basis matrices of the $\{2, 6\}$ -threshold scheme:

$$A_0 = \begin{bmatrix} 0, 1, 0, 1 \\ 1, 0, 1, 0 \\ 1, 1, 0, 0 \\ 0, 0, 1, 1 \\ 1, 0, 0, 1 \\ 0, 1, 1, 0 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 0, 1, 0, 1 \\ 0, 1, 0, 1 \\ 0, 1, 0, 1 \\ 0, 1, 0, 1 \\ 0, 1, 0, 1 \\ 0, 1, 0, 1 \end{bmatrix} \quad (8.22)$$

are defined using 2×2 blocks. The decrypted block is recognized with the spatial contrast value $1/2$ for secret unity bits while decryption of secret zero bits can result in the contrast value 0 or $1/4$ depending on which two of six generated shares are available for decryption.

The interested reader can find the guidelines for the construction of basis matrices corresponding to higher-order $\{k, n\}$ -threshold configurations in [25], [34]. However, it should be noted that expanding share dimensions may be a limiting factor in practical applications. Therefore, some recent research effort has been devoted to reduction and/or minimization of share blocks used for encryption/decryption [26], [51], [52].

Bibliography

- [1] F. Bartolini, M. Barni, A. Tefas, and I. Pitas, "Image authentication techniques for surveillance applications," *Proceedings of the IEEE*, vol. 89, pp. 1403–1418, October 2001.
- [2] M. Wu, W. Trappe, Z. J. Wang, and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," *IEEE Signal Processing Magazine*, vol. 21, pp. 15–27, March 2004.
- [3] D. C. Lou and J. L. Liu, "Steganographic method for secure communications," *Computers and Security*, vol. 21, pp. 449–460, October 2002.
- [4] I. Cox, M. Miller, and J. Bloom, *Digital Watermarking*. San Francisco: Morgan Kaufmann Publishers, 2001.
- [5] A. Menezes, P. V. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.
- [6] E. T. Lin, A. M. Eskicioglu, R. L. Lagendijk, and E. J. Delp, "Advances in digital video content protection," *Proceedings of the IEEE*, vol. 93, pp. 171–183, January 2005.
- [7] A. M. Eskicioglu and E. J. Delp, "An overview of multimedia content protection in consumer electronics devices," *Signal Processing: Image Communication*, vol. 16, pp. 681–699, April 2001.
- [8] G. W. Braudaway, K. A. Magerlein, and F. Mintzer, "Protecting publicly-available images with a visible image watermark," *Proceedings of SPIE*, vol. 2659, pp. 126–133, February 2659.
- [9] M. Barni, F. Bartolini, and A. Piva, "Multichannel watermarking of color images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 142–156, March 2000.
- [10] C. H. Tzeng, Z. F. yang, and W. H. Tsai, "Adaptive data hiding in palette images by color ordering and mapping with security protection," *IEEE Transactions on Communications*, vol. 52, pp. 791–800, May 2004.
- [11] C. S. Chan, C. C. Chang, and Y. C. Hu, "Color image hiding scheme using image differencing," *Optical Engineering*, vol. 44, p. 017003, January 2005.
- [12] J. Wen, M. Severa, W. J. Zeng, M. Luttrell, and W. Jin, "A format-compliant configurable encryption framework for access control of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 545–557, June 2002.
- [13] C. P. Wu and C. C. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Transactions on Multimedia*, vol. 7, pp. 828–839, October 2005.
- [14] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Transactions on Signal Processing*, vol. 48, pp. 2439–2451, August 2000.
- [15] T. Lookabaugh and D. C. Sicker, "Selective encryption for consumer applications," *IEEE Communication Magazine*, pp. 124–129, May 2004.
- [16] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, pp. 118–129, March 2003.

- [17] K. Martin, R. Lukac, and K. N. Plataniotis, "Efficient encryption of wavelet-based coded color images," *Pattern Recognition*, vol. 38, pp. 1111–1115, July 2005.
- [18] S. Lian, J. Sun, D. Zhang, and Z. Wang, "A selective image encryption scheme based on JPEG2000 codec," *Lecture Notes in Computer Science*, vol. 3332, pp. 65–72, 2004.
- [19] A. Sinha and K. Singh, "Image encryption by using fractional Fourier transform and jigsaw transform in image bit planes," *Optical Engineering*, vol. 44, p. 057001, May 2005.
- [20] Y. Sadourny and V. Conan, "A proposal for supporting selective encryption in JPSEC," *IEEE Transactions on Consumer Electronics*, vol. 49, pp. 864–849, November 2003.
- [21] A. M. Eskicioglu, E. J. Delp, and M. R. Eskicioglu, "New channels for carrying copyright and usage rights data in digital multimedia distribution," in *Proc. International Conference on Information Technology: Research and Education (ITRE'03)*, pp. 94–98, August 2003.
- [22] W. Lou, W. Liu, and Y. Fang, "A simulation study of security performance using multipath routing in ad hoc networks," in *Proc. IEEE Vehicular Technology Conference (VTC'03)*, vol. 3, pp. 2142–2146, October 2003.
- [23] D. C. Lou, J. M. Shieh, and H. K. Shieh, "Copyright protection scheme based on chaos and secret sharing techniques," *Optical Engineering*, vol. 44, p. 117004, November 2005.
- [24] C. Padró and G. Sáez, "Lower bounds on the information rate of secret sharing schemes with homogeneous access structure," *Information Processing Letters*, vol. 83, pp. 345–351, September 2002.
- [25] G. Ateniese, C. Blundo, A. de Santis, and D. R. Stinson, "Visual cryptography for general access structures," *Information and Computation*, vol. 129, pp. 86–106, September 1996.
- [26] R. Lukac and K. N. Plataniotis, "A cost-effective encryption scheme for color images," *Real-Time Imaging, Special Issue on Multi-Dimensional Image Processing*, vol. 11, pp. 454–464, October-December 2005.
- [27] M. Naor and A. Shamir, "Visual cryptography," *Lecture Notes in Computer Science*, vol. 950, pp. 1–12, 1994.
- [28] C. C. Chang and J. C. Chuang, "An image intellectual property protection scheme for gray-level images using visual secret sharing strategy," *Pattern Recognition Letters*, vol. 23, pp. 931–941, June 2002.
- [29] C. N. Yang, "New visual secret sharing schemes using probabilistic method," *Pattern Recognition Letters*, vol. 25, pp. 481–494, March 2004.
- [30] G. C. Tai and L. W. Chang, "Visual cryptography for digital watermarking in still images," *Lecture Notes in Computer Science*, vol. 3332, pp. 50–57, December 2004.
- [31] C. S. Tsai and C. C. Chang, "A new repeating color watermarking scheme based on human visual model," *Eurasip Journal on Applied Signal Processing*, vol. 2004, pp. 1965–1972, October 2004.
- [32] H. Guo and N. D. Georganas, "A novel approach to digital image watermarking based on a generalized secret sharing schemes," *Multimedia Systems*, vol. 9, pp. 249–260, September 2003.
- [33] T. Hofmeister, M. Krause, and H. Simon, "Contrast optimal k out of n secret sharing schemes in visual cryptography," *Theoretical Computer Science*, vol. 240, pp. 471–485, June 2000.

- [34] P. A. Eisen and D. R. Stinson, "Threshold visual cryptography schemes with specified levels of reconstructed pixels," *Design, Codes and Cryptography*, vol. 25, pp. 15–61, January 2002.
- [35] R. Lukac and K. N. Plataniotis, "Bit-level based secret sharing for image encryption," *Pattern Recognition*, vol. 38, pp. 767–772, May 2005.
- [36] J. C. Hou, "Visual cryptography for color images," *Pattern Recognition*, vol. 36, pp. 1619–1629, July 2003.
- [37] T. Ishihara and H. Koga, "A visual secret sharing scheme for color images based on meanvalue-color mixing," *IEICE Transactions on Fundamentals*, vol. E86-A, pp. 194–197, January 2003.
- [38] C. C. Lin and W. H. Tsai, "Visual cryptography for gray-level images by dithering techniques," *Pattern Recognition Letters*, vol. 24, pp. 349–358, January 2003.
- [39] P. W. Wong and N. S. Memon, "Image processing for halftones," *IEEE Signal Processing Magazine*, vol. 20, pp. 59–70, July 2003.
- [40] R. A. Ulichney, "Dithering with blue noise," *Proceedings of the IEEE*, vol. 76, pp. 56–79, January 1988.
- [41] T. Ishihara and H. Koga, "New constructions of the lattice-based visual secret sharing scheme using mixture of colors," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E85-A, pp. 158–166, January 2002.
- [42] H. Koga, M. Iwamoto, and H. Yamamoto, "An analytic construction of the visual secret sharing scheme for color images," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 51, pp. 262–272, January 2001.
- [43] A. Adhikari and S. Sikdar, "A new $(2, n)$ visual threshold scheme for color images," *Lecture Notes in Computer Science*, vol. 2904, pp. 148–161, December 2003.
- [44] R. Lukac and K. N. Plataniotis, "Colour image secret sharing," *IEE Electronics Letters*, vol. 40, pp. 529–530, April 2004.
- [45] S. Ramprasad, N. R. Shanbha, and I. N. Hajj, "Analytical estimation of signal transition activity from word-level statistics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, pp. 718–733, July 1997.
- [46] R. Lukac and K. N. Plataniotis, "Image representation based secret sharing," *Communications of the CCISA (Chinese Cryptology Information Security Association), Special Issue on Visual Secret Sharing*, vol. 11, pp. 103–114, April 2005.
- [47] R. Lukac, K. N. Plataniotis, and C. N. Yang, *Encyclopedia of Multimedia*, ch. Image secret sharing. Springer, 2005.
- [48] C. S. Petrie and J. A. Connelly, "A noise-based ic random number generator for applications in cryptography," *IEEE Transactions on Circuits and Systems I*, vol. 47, pp. 615–621, May 2000.

- [49] R. Lukac and K. N. Plataniotis, "Digital image indexing using secret sharing schemes: A unified framework for single-sensor consumer electronics," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 908–916, August 2005.
- [50] S. Sudharsan, "Shared key encryption of JPEG color images," *IEEE Transactions on Consumer Electronics*, vol. 51, pp. 1204–1211, November 2005.
- [51] C. N. Yang and T. S. Chen, "Aspect ratio invariant visual secret sharing schemes with minimum pixel expansion," *Pattern Recognition Letters*, vol. 26, pp. 193–206, January 2005.
- [52] C. C. Lin and W. H. Tsai, "Secret image sharing with capability of share data reduction," *Optical Engineering*, vol. 42, pp. 2340–2345, August 2005.

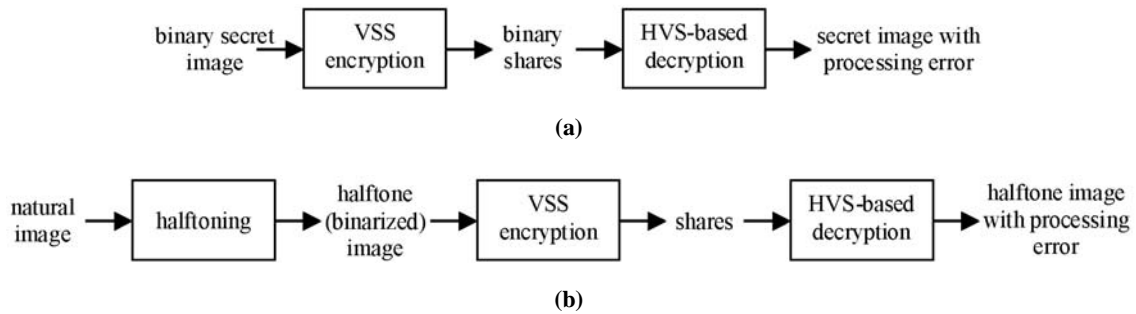


Figure 8.1: Visual secret sharing for: (a) binary images, (b) natural continuous-tone images.

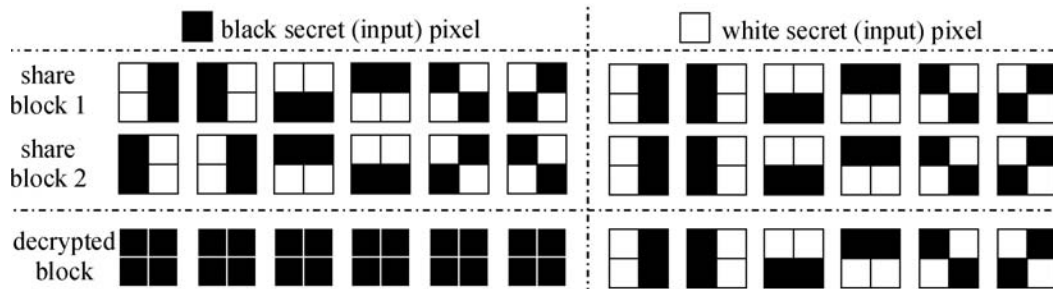


Figure 8.2: VSS concept demonstrated using a {2, 2}-threshold scheme.

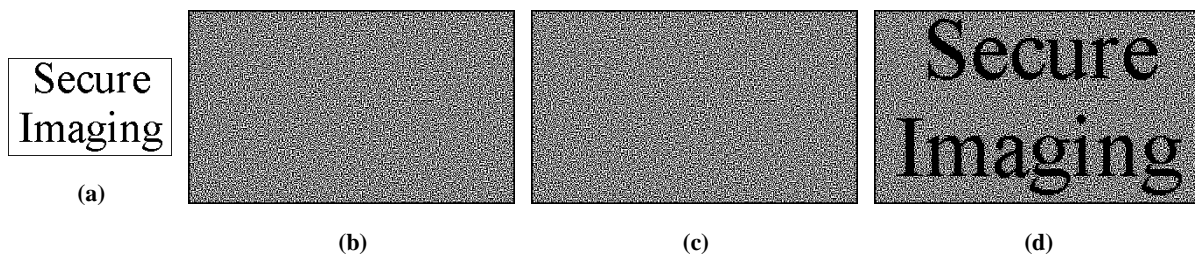


Figure 8.3: Secure binary imaging using {2, 2}-VSS scheme: (a) 111 × 187 binary secret image, (b,c) 222 × 374 binary shares, (d) 222 × 374 binary decrypted image.

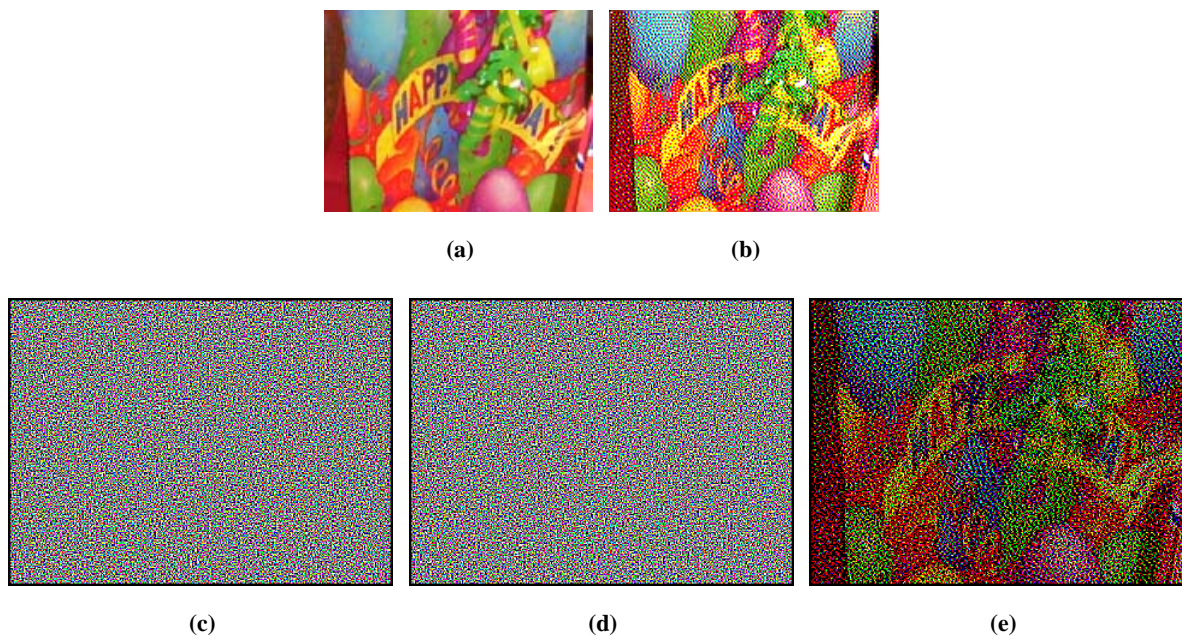


Figure 8.4: Secure color imaging using $\{2, 2\}$ -VSS scheme: (a) 120×160 color secret image, (b) 120×160 halftone image produced using Floyd-Steinberg filter, (c,d) 240×320 binarized color shares, (e) 240×320 decrypted color image.

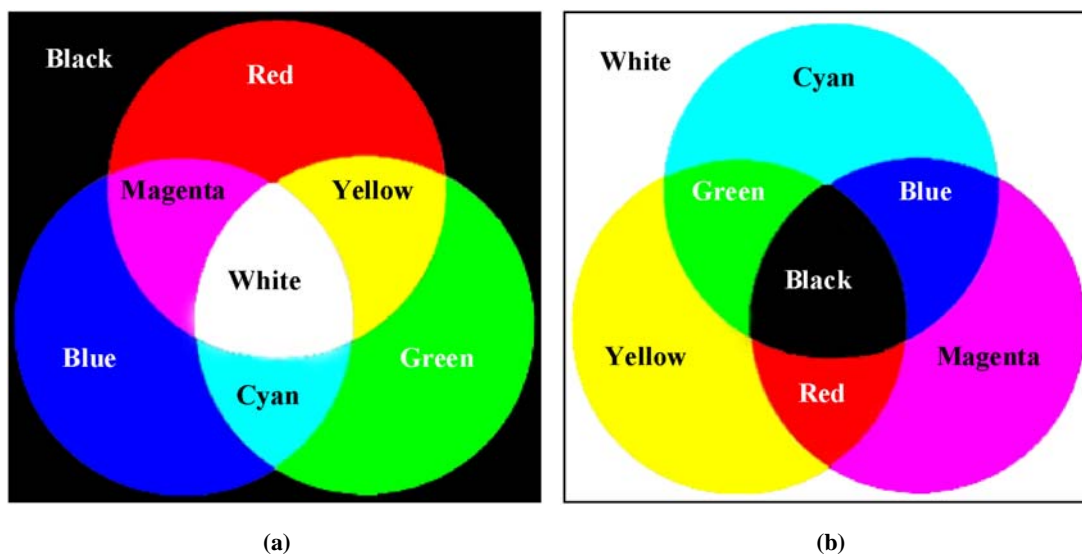


Figure 8.5: Two most common color models: (a) additive model, (b) subtractive model.

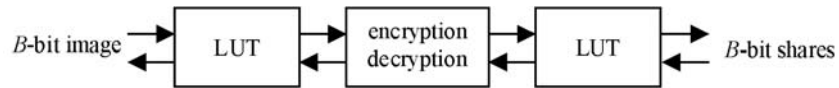


Figure 8.6: Bit-level processing based image secret sharing. Both bit-level decomposition and stacking can be realized using look-up tables (LUT).

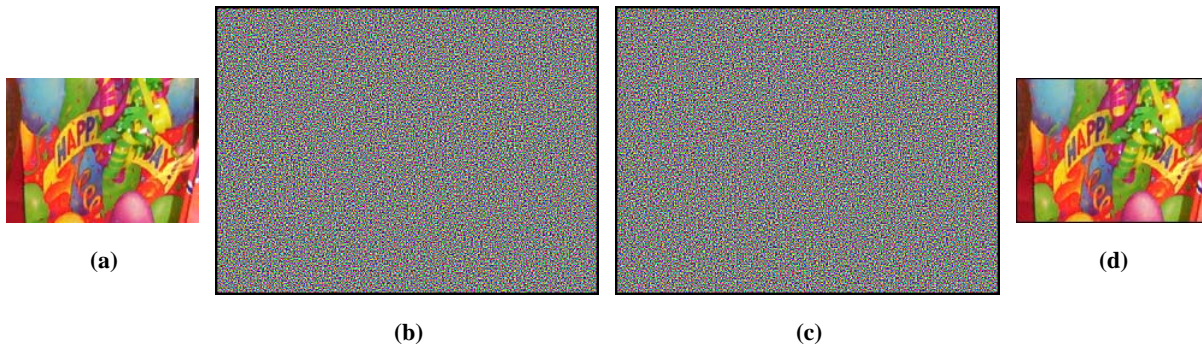


Figure 8.7: Secure color imaging using $\{2, 2\}$ -ISS scheme: (a) 120×160 color secret image, (b,c) 240×320 full-color shares, (d) 120×160 decrypted color image.

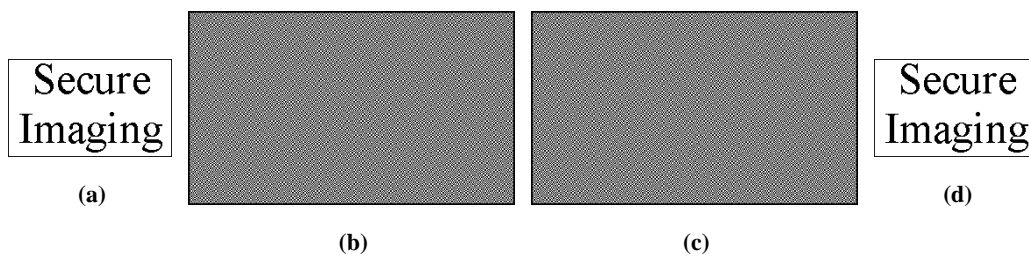


Figure 8.8: Secure binary imaging using $\{2, 2\}$ -ISS scheme: (a) 111×187 binary secret image, (b,c) 222×374 binary shares, (d) 111×187 binary decrypted image.

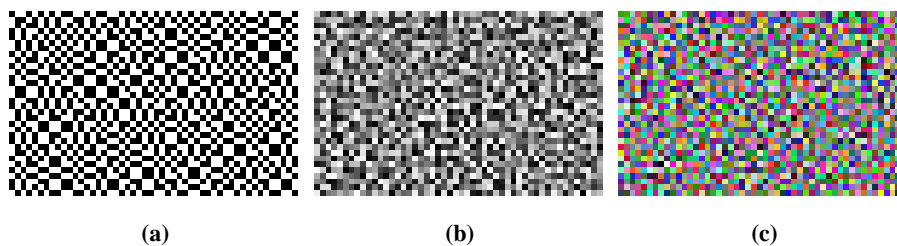


Figure 8.9: ISS share formats generated for: (a) binary secret image, (b) gray-scale secret image, (c) full-color secret image.

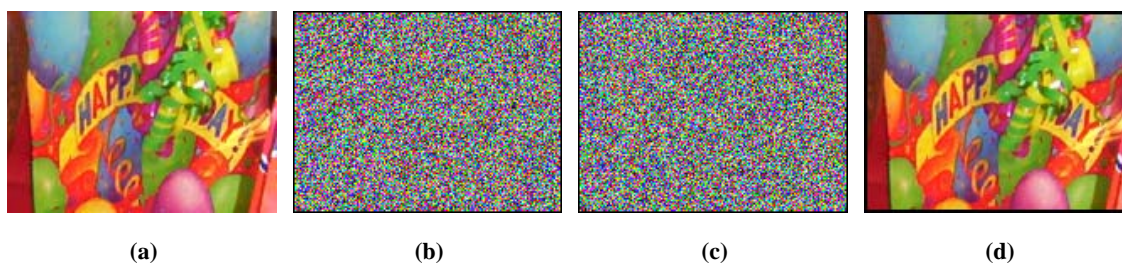


Figure 8.10: Secure color imaging using $\{2,2\}$ -ISS scheme: (a) 120×160 color secret image, (b,c) 120×160 full-color shares, (d) 120×160 decrypted color image.

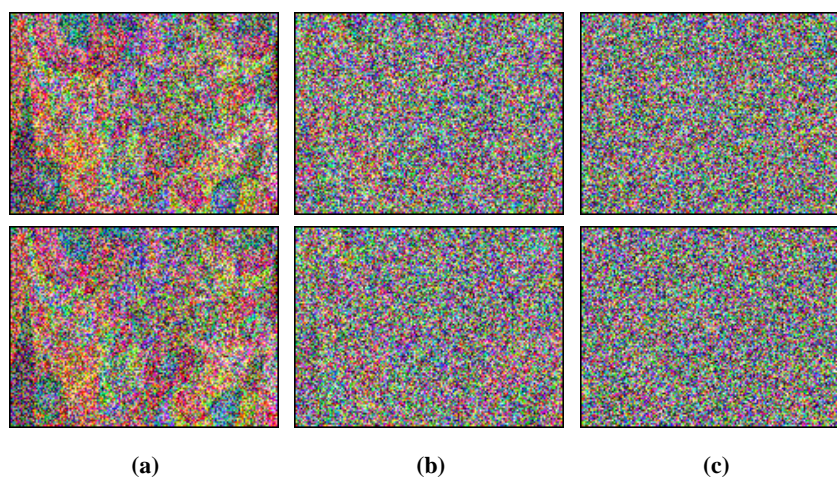


Figure 8.11: Color shares obtained using the non-expansive ISS solution when cryptographic processing is performed for reduced set of binary levels: (a) $b = 1$, (b) $b = 1, 2$, (c) $b = 1, 2, 3$.

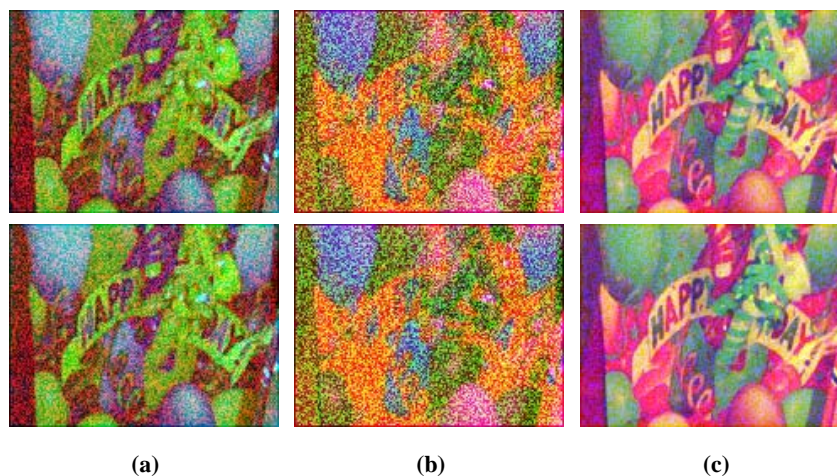


Figure 8.12: Color shares obtained using the non-expansive ISS solution when cryptographic processing is performed for a single color channel: (a) R channel with $c = 1$, (b) G channel with $c = 2$, (c) B channel with $c = 3$.

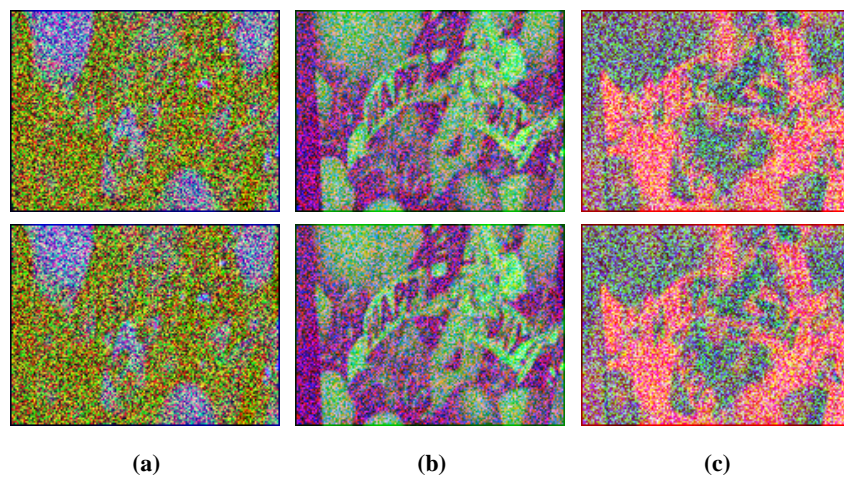


Figure 8.13: Color shares obtained using the non-expansive ISS solution when cryptographic processing is performed for two color channels: **(a)** RG channels with $c = 1$ and $c = 2$, **(b)** RB channels with $c = 1$ and $c = 3$, **(c)** GB channels with $c = 2$ and $c = 3$.