PERGAMON

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition 38 (2005) 767–772

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Rapid and brief communication

Bit-level based secret sharing for image encryption

Rastislav Lukac*,[1], Konstantinos N. Plataniotis

Bell Canada Multimedia Laboratory, Room BA 4157, The Edward S. Rogers Sr. Department of ECE, University of Toronto, 10 King's College Road, Toronto, Ont., Canada M5S 3G4

Received 25 October 2004; accepted 3 November 2004

Abstract

A new secret sharing scheme capable of protecting image data coded with B bits per pixel is introduced and analyzed in this paper. The proposed input-agnostic encryption solution generates B-bit shares by combining bit-level decomposition/stacking with a $\{k, n\}$-threshold sharing strategy. Perfect reconstruction is achieved by performing decryption through simple logical operations in the decomposed bit-levels without the need for any postprocessing operations. The framework allows for cost-effective cryptographic image processing of B-bit images over the Internet.

Keywords: Secret sharing; Image encryption; Bit-level processing; Visual cryptography

1. Introduction

Secret sharing-based image encryption technologies [1] can be utilized to secure data transmission in multimedia networks and mobile public networks which are used for exchange of private images such as scanned (e.g. financial) documents and digital personal photographs. Visual cryptography [2] is a secret sharing procedure for image data, which uses the properties of the human visual system to force the recognition of a secret message from overlapping encrypted images (shares) without additional computations and any knowledge of cryptography. In existing schemes a well-known $\{k, n\}$-threshold procedure is used to encrypt the secret image into n shares, which are then distributed amongst n recipients [3,4]. The shares contain seemingly random information, however, based on the transparent/frosted representation of the shares if any k (or more) recipients stack their shares printed as transparencies together on an overhead projector the secret image is visually revealed. On the other hand, any $(k-1)$ or fewer shares cannot be used to decrypt the transmitted information.

Unfortunately, visual sharing schemes cannot restore the transmitted information to its original quality when the original input is a natural image. This is due to the fact that the $\{k, n\}$-threshold scheme operates on binary or binarized inputs and uses optical frosted/transparent representation. A common procedure [3,4] is to convert continuous-tone images into halftone images [5] with a binary representation. Then the half-tone version of the input image is used instead of the original information. The requirement for inputs of the binary or dithered nature only and the fact that the output is not recovered in digital form limits the applicability of visual cryptography.

The secret sharing scheme proposed here offers a new approach to secret sharing encryption which differs significantly from traditional image sharing schemes in Refs. [2–4] or (color) input-specific {2, 2} (private-key) scheme in Ref. [1]. Unlike past image sharing schemes, the proposed $\{k, n\}$-technique operates directly on the bit planes of the digital

* Corresponding author. Tel.: +1 416 978 6845; fax: +1 416 978 4425.
E-mail address: lukacr@dsp.utoronto.ca (R. Lukac)
URL: http://www.dsp.utoronto.ca/~lukacr.
[1] Partially supported by a NATO/NSERC Science award.

0031-3203/$30.00
doi:10.1016/j.patcog.2004.11.010

input. If the input image with the B-bit code word representation of the samples is decomposed into B bit-levels (planes), each one can be viewed as a binary image. By stacking individually encrypted bit planes, the scheme produces the B-bit shares useful for secure distribution over the untrusted public networks. The decryption function recovers the original B-bit image content unchanged and without the need for expensive postprocessing operations. The decrypted output is readily available in digital form, and there is no requirement for external hardware (overhead projector) or manual intervention needed in Refs. [2–4] or vectorial fields' arrangements required in Ref. [1]. This feature in conjunction with the overall simplicity of the approach make the proposed input-agnostic solution attractive for real-time secret sharing-based encryption/decryption of natural images.

2. Conventional visual cryptography

A $\{k, n\}$-threshold visual cryptography scheme [2], often called $\{k, n\}$ visual secret sharing (VSS) or simply $\{k, n\}$-VSS, is used to encrypt an input image by splitting the original content into n, seemingly random, shares $S_1, S_2, \ldots, S_n$. The procedure is termed visual since the secret information is recovered through visual inspection of the stacked k (or more) allowed shares without the need for complicated cryptographic mechanisms and computations.

Due to the nature of conventional visual cryptography the input is a binary image [2,3]. To encrypt a $K_1 \times K_2$ binary image with spatial coordinates $i = 1, 2, \ldots, K_1$ and $j = 1, 2, \ldots, K_2$, each original binary pixel $r_{(i,j)}$ (i.e. $r_{(i,j)} = 1$ for white and $r_{(i,j)} = 0$ for black) is handled separately via an encryption function $f_e(\cdot)$ to produce a $m_1 \times m_2$ block of black and white pixels in each of the n shares. Thus, a $K_1 \times K_2$ input binary image is encrypted into n binary shares $S_1, S_2, \ldots, S_n$ each one with a spatial resolution of $m_1 K_1 \times m_2 K_2$ pixels. Since the spatial arrangement of the pixels varies from block to block, it is impossible to recover the useful information without accessing a predefined number of shares.

Let $f_e(\cdot)$ be the encryption function which maps a reference binary pixel $r_{(i,j)}$ located at position (i, j) in the original image into $m_1 \times m_2$-sized blocks in the various shares. Assuming for simplicity a basic $\{2, 2\}$ scheme with 2×2 blocks $\mathbf{s}_1 = [s'_{(2i-1,2j-1)}, s'_{(2i-1,2j)}, s'_{(2i,2j-1)}, s'_{(2i,2j)}]$ in the share S_1 and $\mathbf{s}_2 = [s''_{(2i-1,2j-1)}, s''_{(2i-1,2j)}, s''_{(2i,2j-1)}, s''_{(2i,2j)}]$ in the share S_2, the encryption process is given by

$$f_e(r_{(i,j)}) = \begin{cases} [\mathbf{s}_1, \mathbf{s}_2]^{\mathrm{T}} \in C_0 & \text{for } r_{(i,j)} = 0, \\ [\mathbf{s}_1, \mathbf{s}_2]^{\mathrm{T}} \in C_1 & \text{for } r_{(i,j)} = 1. \end{cases} \tag{1}$$

The sets

$$C_0 = \left\{ \begin{bmatrix} 0,1,0,1 \\ 1,0,1,0 \end{bmatrix}, \begin{bmatrix} 1,0,1,0 \\ 0,1,0,1 \end{bmatrix}, \begin{bmatrix} 0,0,1,1 \\ 1,1,0,0 \end{bmatrix} \begin{bmatrix} 1,1,0,0 \\ 0,0,1,1 \end{bmatrix}, \begin{bmatrix} 1,0,0,1 \\ 0,1,1,0 \end{bmatrix}, \begin{bmatrix} 0,1,1,0 \\ 1,0,0,1 \end{bmatrix} \right.,$$

and

$$C_1 = \left\{ \begin{bmatrix} 0,1,0,1 \\ 0,1,0,1 \end{bmatrix}, \begin{bmatrix} 1,0,1,0 \\ 1,0,1,0 \end{bmatrix}, \begin{bmatrix} 0,0,1,1 \\ 0,0,1,1 \end{bmatrix} \begin{bmatrix} 1,1,0,0 \\ 1,1,0,0 \end{bmatrix}, \begin{bmatrix} 1,0,0,1 \\ 1,0,0,1 \end{bmatrix}, \begin{bmatrix} 0,1,1,0 \\ 0,1,1,0 \end{bmatrix} \right.,$$

include all matrices (Fig. 1) obtained by permuting the columns of the $n \times m_1 m_2$ basis matrices

$$A_0 = \begin{bmatrix} 0,1,0,1 \\ 1,0,1,0 \end{bmatrix}, \quad \text{and} \quad A_1 = \begin{bmatrix} 0,1,0,1 \\ 0,1,0,1 \end{bmatrix},$$

respectively [2,3]. The size of the basis matrices depends on the expansion factor $m_1 m_2$ and the number of participants, which is given by n. Since $m_1 m_2$ represents the factor by which each share is larger than the original image, it is desirable to make $m_1 m_2$ as small as possible [2].

If a secret pixel is white, i.e. $r_{(i,j)} = 1$, then $[\mathbf{s}_1, \mathbf{s}_2]^{\mathrm{T}}$ can be any member of the set C_1. If a secret pixel is black, i.e. $r_{(i,j)} = 0$, then $[\mathbf{s}_1, \mathbf{s}_2]^{\mathrm{T}}$ should be selected from the set C_0. The choice of $[\mathbf{s}_1, \mathbf{s}_2]^{\mathrm{T}}$ is guided by a random number generator, which determines the random character of the

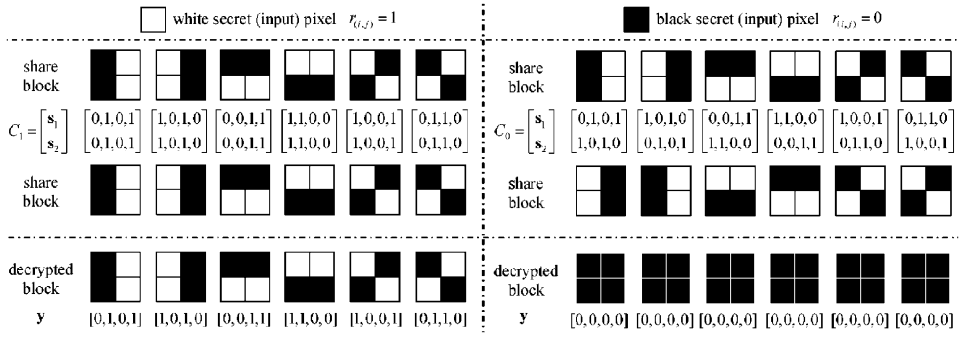

Fig. 1. Conventional visual cryptography strategy.

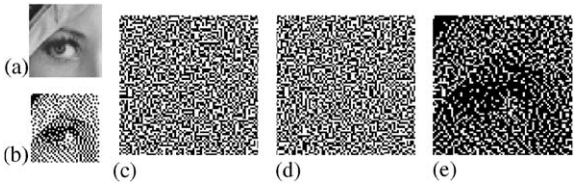

Fig. 2. Conventional $\{2, 2\}$-threshold visual cryptography framework applied to the gray-scale input: (a) a $K_1 \times K_2$ original gray-scale image, (b) a $K_1 \times K_2$ halftone image obtained using the Floyd–Steinberg filter [5], (c) a $2K_1 \times 2K_2$ share S_1, (d) a $2K_1 \times 2K_2$ share S_2, (e) a $2K_1 \times 2K_2$ decrypted binary (output) image.

shares. The graphical interpretation of the matrices included in C_0 and C_1 is given in Fig. 1. For a $\{2, 2\}$ scheme considered here, each pixel in $\mathbf{s}_1$ is equivalent to each pixel in $\mathbf{s}_2$ if $r_{(i,j)} = 1$, and each pixel in $\mathbf{s}_1$ should complement each pixel in $\mathbf{s}_2$ if $r_{(i,j)} = 0$. The figure also depicts the decrypted blocks obtained by stacking the share blocks. The decrypted block shown in Fig. 1 is produced through a decryption function $f_d(s'_{(u,v)}, s''_{(u,v)})$ which is defined as follows:

$$y_{(u,v)} = f_d(\cdot) = \begin{cases} 1 & \text{for } s'_{(u,v)} = 1 \vee s''_{(u,v)} = 1, \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where (u, v), for $u = 1, 2, \ldots, 2K_1$ and $v = 1, 2, \ldots, 2K_2$, denotes the spatial location in a $2K_1 \times 2K_2$ share. The term $y_{(u,v)}$ indicates a pixel in the $2K_1 \times 2K_2$ decrypted image.

The application of a conventional $\{k, n\}$-VSS scheme to a $K_1 \times K_2$ natural image with B-bit/pixel representation, such as the one depicted in Fig. 2a requires half toning. The image is first transformed into a $K_1 \times K_2$ halftone image (Fig. 2b) by using the density of the net dots to simulate the gray levels [5]. Since the halftone image is a binary image, it is perfectly suited for conventional visual cryptography. Note that there are many ways to obtain halftones and the $\{k, n\}$-threshold framework can work with all of them. Applying the $\{2, 2\}$-threshold scheme of (1) to the image depicted in Fig. 2b the two $2K_1 \times 2K_2$ binary shares shown in Fig. 2c and d are produced. Fig. 2e depicts the $2K_1 \times 2K_2$ decrypted image (result) obtained by stacking the two shares together using (2). Fig. 3a shows the block-diagram representation of the conventional visual cryptographic solution when it is applied to a B-bit natural image. As it can be seen, the procedure involves four steps: halftoning, encryption, decryption and inverse halftoning. Note that inverse halftoning does not recover the original continuous-tone image as the process introduces significant impairments and is usually computationally demanding [5].

Visual inspection of both the binary input and the recovered binary image indicates that: (i) the decrypted image is darker, and (ii) the input image is of quarter size compared to the decrypted output. Therefore, even in the case of binary (or dithered) input, the conventional $\{k, n\}$-threshold visual cryptography (i) cannot provide perfect reconstruction, either in terms of pixel intensity or spatial resolution, and (ii) is not appropriate for real-time applications. Thus, an alternative solution is needed.

3. *B*-bit image secret sharing

Let us consider a digital $K_1 \times K_2$ input image with a B-bit per pixel representation. For presentation purposes a gray-scale natural image with 8 bits/pixel will be used in the sequence. The 8-bit representation can describe 256 gray-scale levels (integers ranging from 0 to 255). In such a representation, each integer pixel value can be expressed equivalently in a binary form using

$$o_{(i,j)} = o^1_{(i,j)} 2^{B-1} + o^2_{(i,j)} 2^{B-2} + \cdots + o^{B-1}_{(i,j)} 2 + o^B_{(i,j)}, \quad (3)$$

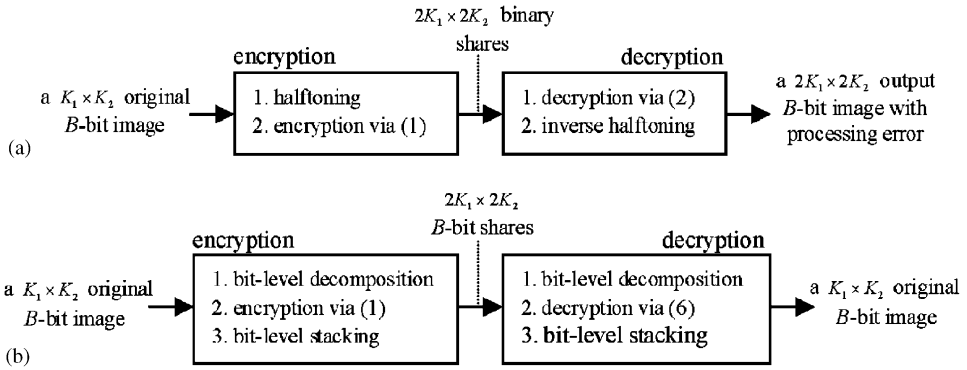

Fig. 3. Block scheme of: (a) the conventional VSS-based solution when applied to B-bit image, (b) the proposed B-bit secret sharing solution. In both examples expansion parameters $m_1 = m_2 = 2$ are considered.

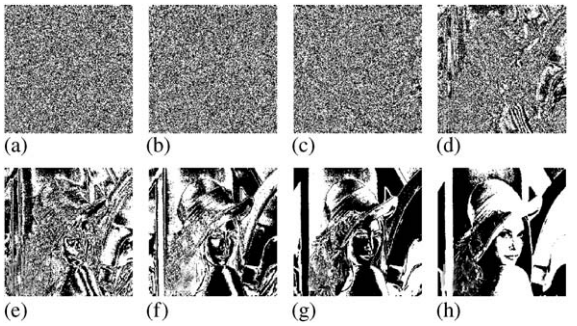

Fig. 4. Binary images corresponding to the bit-levels of the gray-scale ($B=8$) image Lena: (a) $b=8$, (b) $b=7$, (c) $b=6$, (d) $b=5$, (e) $b=4$, (f) $b=3$, (g) $b=2$, (h) $b=1$.

where (i, j) denotes the spatial location and $o^b_{(i,j)}$ indicates the bit value at the bit level $b = 1, 2, \ldots, B$, with $o^1_{(i,j)}$ corresponding to the most significant bit (MSB). The bit-level decomposition is a natural way to decompose the input image to a series of B binary images depicted in Fig. 4, and from this point of view constitutes the ideal preprocessing step for share-based encryption.

After achieving B binary planes (Fig. 4), the conventional encryption function (1) is utilized to generate the binary shares S^b_1 and S^b_2 using the reference pixel $r_{(i,j)} = o^b_{(i,j)}$. Assuming that $s'^b_{(u,v)} \in S^b_1$ and $s''^b_{(u,v)} \in S^b_2$, for $u = 1, 2, \ldots, 2K_1$ and $v = 1, 2, \ldots, 2K_2$, denote the pixels in the $2K_1 \times 2K_2$ binary shares S^b_1 and S^b_2, respectively, the B-bit share pixels $s'_{(u,v)} \in S_1$ and $s''_{(u,v)} \in S_2$ are constituted by bit-level stacking as follows:

$$s'_{(u,v)} = s'^1_{(u,v)}2^{B-1} + s'^2_{(u,v)}2^{B-2} + \cdots + s'^{B-1}_{(u,v)}2 + s'^B_{(u,v)}, \tag{4}$$

$$s''_{(u,v)} = s''^1_{(u,v)}2^{B-1} + s''^2_{(u,v)}2^{B-2} + \cdots + s''^{B-1}_{(u,v)}2 + s''^B_{(u,v)}. \tag{5}$$

Depending on the particular bit-levels on which $f_e(\cdot)$ is applied and the random choice of the block representing $o^b_{(i,j)}$, the original pixel $o_{(i,j)}$ and the integer-valued share pixels $s'_{(2i-1,2j-1)}$, $s'_{(2i-1,2j)}$, $s'_{(2i,2j-1)}$, $s'_{(2i,2j)}$ and $s''_{(2i-1,2j-1)}$ $s''_{(2i-1,2j)}$, $s''_{(2i,2j-1)}$, $s''_{(2i,2j)}$ can differ significantly. Assuming that N denotes the number of unique matrices obtained by column permutations of the basis matrices corresponding to the $\{k, n\}$-scheme, the B-bit pixel is encrypted using one of N^B unique $m_1 \times m_2$ share blocks of B-bit pixels. Thus, compared to the schemes operating on binary (dithered) images which allows for using only N unique share blocks of binary pixels, the method increases security and prevents unauthorized decryption through brute-force enumeration.

To faithfully decrypt the original B-bit image from its B-bit shares, the decryption function must satisfy the perfect reconstruction property meaning that the output should be identical to the original input. This can be obtained only if the encryption and decryption operations are reciprocal. Taking advantage of the arrangements of the binary pixels in the sets C_0 and C_1 for the specific case of a $\{2, 2\}$ scheme [1], the decryption function $f_d(\cdot)$ recovers $o^b_{(i,j)} = 1$ for $s'^b_{(2i-1,2j-1)} = s''^b_{(2i-1,2j-1)}$ and $o^b_{(i,j)} = 0$ for $s'^b_{(2i-1,2j-1)} \neq s''^b_{(2i-1,2j-1)}$, with (i, j) denoting location in a $K_1 \times K_2$ reference image. By decimating via a factor of 2 it is possible to associate the share bits located at $(2i-1, 2j-1)$ to the original bit located at (i, j) for each of the bit-levels $b = 1, 2, \ldots, B$.

Since $f_d(\cdot)$ defined through the above consistent/complement decryption concept of [1] can be used for a simple $\{2, 2\}$-scheme only, the decryption function $f_d(\cdot)$ generalized for any $\{k, n\}$ configuration is proposed here as follows:

$$o^b_{(i,j)} = f_d(\mathbf{s}^b_1, \mathbf{s}^b_2, \ldots, \mathbf{s}^b_k) = \begin{cases} 1 & \text{for } [\mathbf{s}^b_1, \mathbf{s}^b_2, \ldots, \mathbf{s}^b_k]^{\mathrm{T}} \in C_1, \\ 0 & \text{for } [\mathbf{s}^b_1, \mathbf{s}^b_2, \ldots, \mathbf{s}^b_k]^{\mathrm{T}} \in C_0, \end{cases} \tag{6}$$

where $\mathbf{s}^b_q \in S^b_q$, for $q = 1, 2, \ldots, k$, denotes a $m_1 \times m_2$ block at the bth bit level S^b_q of the share S_q. This concept can be generalized for any set of the share blocks $\{\mathbf{s}^b_1, \mathbf{s}^b_2, \ldots, \mathbf{s}^b_k\} \subseteq \{\mathbf{s}^b_1, \mathbf{s}^b_2, \ldots, \mathbf{s}^b_n\}$ required in the existing $\{k, n\}$-threshold decryption functions for the case of B-bit images. The determination of the relationship between $\{\mathbf{s}^b_1, \mathbf{s}^b_2, \ldots, \mathbf{s}^b_k\}$ and the sets C_0 and C_1 can easily be done using the contrast properties of the conventional $\{k, n\}$-schemes of [2].

It should be mentioned that the bit-level processing allows for a completely different interpretation of the application of the $\{k, n\}$ secret sharing framework. Since encryption (1) and decryption (6) are reciprocal, perfect reconstruction, a property unavailable in conventional $\{k, n\}$ schemes is obtained. The faithful recovery of the encryption input in digital form makes our scheme ideal for integration into any image processing and communication solution.

Fig. 5 offers a visual comparison of the results obtained via the conventional decryption of (2) and those obtained through the application of the decryption method of (6). It is not difficult to see that in the case of the $\{2, 2\}$-threshold structure our method produces a $K_1 \times K_2$ noise-free image (Fig. 5c). This should be contrasted to the $2K_1 \times 2K_2$ decrypted output of (2) which contains a number of random, noise like, pixels (Fig. 5b). Our solution recovers the spatial dimensionality of the input as (6) performs simultaneously subsampling and decryption, and the original B-bit pixels are generated by stacking the decrypted bit levels o^b according to (3). Fig. 6 provides an overview of the process for the selected part of the test image 'Lena' depicting the $K_1 \times K_2$ original B-bit input image (Fig. 6a), the two $2K_1 \times 2K_2$-sized B-bit shares shown in Fig. 6b and c,

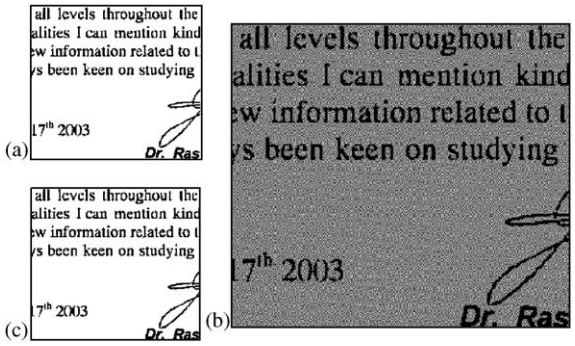

Fig. 5. The {2, 2}-threshold cryptography framework applied to the binary input: (a) a $K_1 \times K_2$ reference image, (b) a $2K_1 \times 2K_2$ output of the conventional VSS decryption procedure, (c) a $K_1 \times K_2$ output of the proposed decryption procedure.

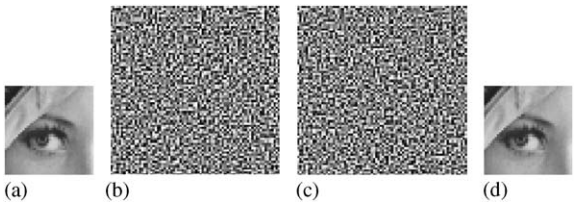

Fig. 6. The proposed B-bit {2, 2}-secret sharing framework applied to the gray-scale input: (a) a $K_1 \times K_2$ original gray-scale image, (b) a $2K_1 \times 2K_2$ gray-scale share S_1, (c) a $2K_1 \times 2K_2$ gray-scale share S_2, (d) restored output.

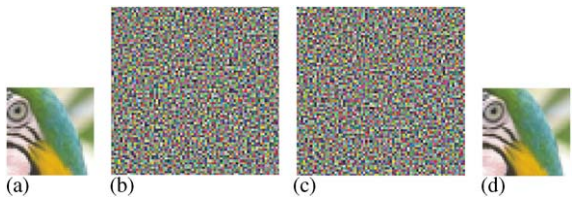

Fig. 7. The proposed B-bit {2, 2}-secret sharing framework applied to the color input: (a) a $K_1 \times K_2$ original color image, (b) a $2K_1 \times 2K_2$ color share S_1, (c) a $2K_1 \times 2K_2$ color share S_2, (d) restored output.

respectively; and the recovered original in Fig. 6d. The proposed scheme can be applied to any B-bit image, natural or computer generated, and therefore can be used to process color RGB images such as the part of the color test image 'Parrots' depicted in Fig. 7a. As before Figs. 7b and c depict the color shares S_1 and S_2 obtained via a {2, 2}-threshold framework while Fig. 7d lists the decrypted output. Simple visual inspection reveals that the image shown in Fig. 7d is identical to the original. Fig. 8 depicts images obtained using a B-bit {2, 6}-threshold structure. As it can be seen from the images listed there, the proposed scheme perfectly works also for higher-order configurations.

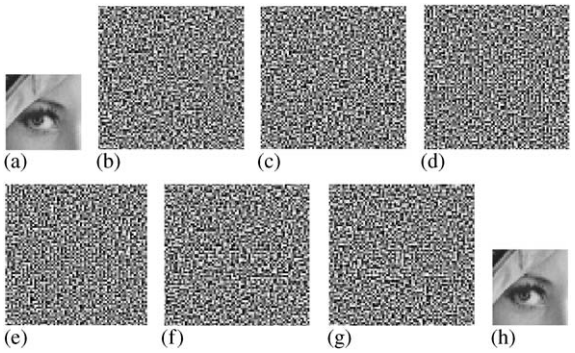

Fig. 8. The proposed B-bit {2, 6}-secret sharing framework applied to the gray-scale input: (a) a $K_1 \times K_2$ original gray-scale image, (b–g) $2K_1 \times 2K_2$ gray-scale shares $S_1, S_2, \ldots, S_6$, (h) restored output using shares S_2 and S_5.

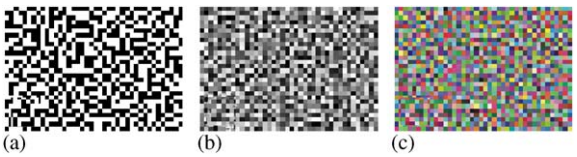

Fig. 9. Details of the share generated by the proposed B-bit secret sharing framework applied to: (a) binary image, (b) gray-scale image, (c) color image. Based on the image representation of the original image, the encrypted image (share) contains random information in the form of: (a) binary noise, (b) gray-scale noise, (c) color noise.

Finally, Fig. 9 provides a visual overview of the differences between the shares generated by the proposed framework for the case of binary ($B = 1$), gray-scale ($B = 8$) and color images ($B = 3 \times 8$). Depending on the depth of the B-bit representation of the input image, the shares contain binary, gray-scale or color random information, respectively. The figure suggests that as we move towards richer visual inputs the degree of security afforded by our method increases, as it becomes increasingly difficult to "guess" by operating on the integer (B-bit) domain.

Apart from the actual performance of any algorithm, its computational complexity is a realistic measure of its practicality and usefulness. Since the proposed cryptographic solution is determined for PC-based applications, the efficiency of the encryption and decryption operations is measured, in terms of the execution time, in such a computing platform. The execution of the developed tool on a personal computer equipped with an Intel Pentium IV 2.40 GHz CPU, 512 MB RAM, Windows XP operating system and MS Visual C++ 5.0 programming environment, required, on average, 0.931 s per a 256×256 gray-scale image for encryption and 1.473 s for decryption. In the case of a 256×256 color image, the execution required 2.914 s for encryption and 4.462 s for decryption.

4. Summary

A *B*-bit secret sharing framework (Fig. 3b) that affords perfect reconstruction of the encrypted image input was introduced. The method proposed here (i) utilizes bit-level decomposition and stacking operations to both encrypt and decrypt *B*-bit image, (ii) preserves all the features of traditional $\{k, n\}$ sharing schemes, (iii) allows for perfect reconstruction of the input *B*-bit image, (iv) encrypts binary, gray-scale and color images, and (v) can be effectively implemented either in software or hardware.

References

[1] R. Lukac, K.N. Plataniotis, Colour image secret sharing, IEE Electron. Lett. 40 (9) (2004) 529–530.
[2] M. Naor, A. Shamir, Visual cryptography, Proc. Eurocrypt '94, LNCS 950 (1994) 1–12.
[3] C.C. Lin, W.H. Tsai, Visual cryptography for gray-level images by dithering techniques, Pattern Recognition Lett. 24 (1–3) (2003) 349–358.
[4] J.C. Hou, Visual cryptography for color images, Pattern Recognition 36 (7) (2003) 1619–1629.
[5] R.A. Ulichney, Dithering with blue noise, Proc. IEEE 76 (1) (1988) 56–79.

About the Author—RASTISLAV LUKAC received the M.S. (Ing.) and Ph.D. degrees in Telecommunications from the Technical University of Kosice, Slovak Republic in 1998 and 2001, respectively. From February 2001 to August 2002 he was an Assistant Professor at the Department of Electronics and Multimedia Communications at the Technical University of Kosice. Since August 2002 he is a Researcher in Slovak Image Processing Center in Dobsina, Slovak Republic. From January 2003 to March 2003 he was a Postdoctoral Fellow at the Artificial Intelligence & Information Analysis Lab at the Aristotle University of Thessaloniki, Greece. Since May 2003 he has been a Postdoctoral Fellow with the Edward S. Rogers Sr. Department of Electrical and Computer Engineering at the University of Toronto in Toronto, Canada.

His research interests include digital camera image processing, microarray image processing, multimedia security, and nonlinear filtering and analysis techniques for color image & video processing. Dr. Lukac is a Member of the IEEE Signal Processing Society. In 2003 he was awarded the NATO/NSERC Science Award.

About the Author—KONSTANTINOS N. PLATANIOTIS received the B. Engineering degree in Computer Engineering from the Department of Computer Engineering and Informatics, University of Patras, Patras, Greece in 1988 and the M.S. and Ph.D. degrees in Electrical Engineering from the Florida Institute of Technology (Florida Tech), Melbourne, Florida in 1992 and 1994 respectively. He was affiliated with the Computer Technology Institute (C.T.I.), Patras, Greece from 1989 to 1991. From August 1997 to June 1999 he was an Assistant Professor with the School of Computer Science at Ryerson University. He is currently an Assistant Professor at the Edward S. Rogers Sr. Department of Electrical & Computer Engineering where he researches and teaches adaptive systems and multimedia signal processing.

Dr. Plataniotis is a Senior Member of IEEE, a past member of the IEEE Technical Committee on Neural Networks for Signal Processing, and the Technical Co-Chair of the Canadian Conference on Electrical and Computer Engineering (CCECE) 2001, and CCECE 2004.