NovelApproach to Data Hiding in Encrypted Image with Distributed Source Encoding

Nischala N
Mtech IV Semester
Kalpataru institute of Technology,
Tiptur.

Raviprakash M L
Assistant Professor,
Dept. Of CSE, Kalpataru institute of Technology,
Tiptur.

***Abstract*—this paper proposes a novel scheme of reversible data hiding (RDH) in encrypted images using distributed source coding (DSC). After the original image is encrypted by the content owner using a stream cipher, the data-hider compresses a series of selected bits taken from the encrypted image to make room for the secret data. The selected bit series is Slepian-Wolf encoded using low density parity check (LDPC) codes. On the receiver side, the secret bits can be extracted if the image receiver has the embedding key only. In case the receiver has the encryption key only, he/she can recover the original image approximately with high quality using an image estimation algorithm. If the receiver has both the embedding and encryption keys, he/she can extract the secret data and perfectly recover the original image using the distributed source decoding. The proposed method outperforms previously published ones.**

***Index Terms*—Reversible data hiding, image encryption, image recovery**

I. INTRODUCTION

Information processing in the encrypted domain has attracted considerable research interests in recent years [1]. In many applications such as cloud computing and delegated calculation, the content owner needs to transmit data to a remote server for further processing. In some cases, the content owner may not trust the service supplier, and needs to encrypt the data before uploading. Thus, the service provider must be able to do the processing in the encrypted domain. Some works have been done for data processing in an encrypted domain, for example, compressing encrypted images [2]-[4], adding a watermark into the encrypted image [5][6], and reversibly hiding data into the encrypted image [7-13]. Unlike robust watermarking, reversible data hiding emphasizes perfect image reconstruction and data extraction, but not the robustness against malicious attacks [14]. Many RDH methods for plaintext images have been proposed [15-19], for example, a common framework of redundancy compression [14], difference expansion (DE) [15] and histogram shifting (HS) [16] approaches. However, these are not applicable to encrypted images since the redundancy in the original image cannot be used directly after image encryption. As a new trend, reversible data hiding in encrypted images allows the service provider to embed additional messages, e.g., image metadata, labels, notations or authentication information, into the encrypted images without accessing the original contents.

The original image is required to be perfectly recovered and the hidden message completely extracted on the receiving side. Reversible data hiding in encrypted images is desirable. For example, in medical applications, a patient does not allow his/her medical images to be revealed to any outsiders, while the database administrator may need to embed medical records or the patient's information into the encrypted images. On the other hand, the original medical image for diagnosis must be recovered without error after decryption and retrieval of the hidden message. The emerging methods [7]-[13] on reversible data hiding in encrypted images are reviewed in Section II.

This paper aims to enhance embedding payload in encrypted images. We propose a separable reversible data hiding method for encrypted images using Slepian-Wolf source encoding [21]. The idea is inspired by the DSC [22-24], in which we encode the selected bits taken from the stream-ciphered image using LDPC codes [25] into syndrome bits to make spare room to accommodate the secret data. With two different keys, the proposed method is separable. The hidden data can be completely extracted using the embedding key, and the original image can be approximately reconstructed with high quality using the encryption key. With both keys available, the hidden data can be completely extracted, and the original image perfectly recovered with the aid of some estimated side information. The proposed method achieves a high embedding payload and good image reconstruction quality, and avoids the operations of room-reserving by the sender.

The rest of the paper is organized as follows. Previous works of RDH in encrypted images are surveyed in Section II. The proposed system is described in Section III. Section IV presents the procedures of image encryption and data embedding. Data extraction and image recovery are elaborated in Section V. DATA EXTRACTION AND IMAGE RECOVERY VII. Discusses the proposed method. Section VII concludes the paper.

II.PREVIOUS WORKS.

In this section, the state-of-the-art RDH techniques of embedding secret message in encrypted images are reviewed. RDH for encrypted image are usually designed for the applications in which the data-hider and the image owner are not the same party. The data-hider cannot access the image content, and the secret message is

held by the data-hider. Thus, encryption is done by the sender, hiding by the data-hider, and data extraction and/or image reconstruction by the receiver. For the ease of discussion, explanations of some frequently used terms are listed in Table I. Table I. Terms used in the paper Terms Explanations original image natural image in plaintext form encrypted image image obtained by encrypting the original image additional bits secret message to be embedded into the encrypted image marked encrypted image encrypted image containing additional data bits approximate image the reconstructed image close to the original image recovered image perfectly restored image that is identical to the original image sender owner of the original image who encrypts the original image and uploads the encrypted image to the server data-hider service provider who embeds additional message into the encrypted image receiver one who receives the marked encrypted image, and performs data extraction and/or image reconstruction Existing RDH methods for encrypted images can be classified into two categories: "vacating room after encryption (VRAE)" and "vacating room before encryption (VRBE)" [11]. In VRAE, the original image is encrypted directly by the sender, and the data-hider embeds the additional bits by modifying some bits of the encrypted data. The idea was first proposed by Puech et al. [7], in which the owner encrypts the original image by Advanced Encryption Standard (AES), and the data-hider embeds one bit in each block containing n pixels, meaning that the embedding rate is 1/n bit-per-pixel (bpp). On the receiver side, data extraction and image recovery are realized by analyzing the local standard deviation during decryption of the marked encrypted image.

This method requires that image decryption and data extraction operations must be done jointly. In other words, extraction and decryption are inseparable. With a different idea, Zhang proposed a practical RDH method for encrypted images in [8], in which the data-hider divides the encrypted image into blocks and embeds one bit into each block by flipping three least significant bits (LSB) of half the pixels in the block. On the receiver side, the marked encrypted image is decrypted to an approximate image. The receiver flips the three LSBs of pixels to form a new block. Due to spatial correlation in natural images, original block is presumed to be much smoother than interfered block. Thus the embedded bits can be extracted and the original image recovered jointly. Embedding rate of this method depends on the block size. If an inappropriate block size is chosen, errors may occur during data extraction and image recovery. This method was improved in [9] by exploiting spatial correlation between neighboring blocks and using a side-match algorithm to achieve a better embedding payload with much lower error rates in image recovery. Both methods [8] and [9] are feasible based on the spatial correlation in natural images. Data extraction, however, are inseparable. To overcome the drawback of inseparability in [7]-[9], a separable RDH scheme was proposed for encrypted images in [10]. The data-hider pseudo-randomly permutes and divides the encrypted image into groups with size of L. The P LSB-planes of each group are compressed with a matrix G sized (P·L-S)×P·L to generate corresponding vectors. Thus, S bits are available for data embedding. On the receiver side, a total of (8-P) most significant bits (MSB) of pixels are obtained by decryption. The receiver then estimates the P LSBs by the MSBs of neighboring pixels. By comparing the estimated bits with the vectors in the coset Ω corresponding to the extracted vectors, the receiver can recover the original bits of the P LSBs. Because the additional bits are embedded in LSBs of the encrypted images, which can be extracted directly before image recovery, data extraction and image recovery are therefore separable. Besides, this method achieves a better embedding rate than [8] and [9]. Another separable method was proposed in [12], in which the data-hider embeds additional bits by a histogram shifting and n-nary data hiding scheme, greatly improving the embedding payload as compared to [8]-[10]. However, as the original image is encrypted with pixel permutation and affine transformation, leakage of image histogram is inevitable under exhaustive attack.

In the VRBE, the original images are processed by the owner before encryption to create spare space for data embedding, and the secret data are embedded into specified positions by the data-hider. For example, the method in [11] creates embedding room in the plaintext image by embedding LSBs of certain pixels into other pixels using a traditional RDH method. The pre-processed image is then encrypted by the owner to generate an encrypted image. Thus, positions of these vacated LSBs in the encrypted image can be used by the data-hider, and a large payload up to 0.5 bpp, can be achieved. With a similar idea, another method based on an estimation technique was proposed in [13], in which a large portion of pixels are used to estimate the rest before encryption, and the final version of encrypted image is formulated by concatenating the encrypted estimating errors and a large group of encrypted pixels. Additional bits can be embedded into the encrypted image by modifying the estimating errors. With this method, PSNR of the approximate image reconstructed by the receiver is higher than previous methods. Both [11] and [13] are separable RDH methods with good embedding rates and reconstruction capability, but require an additional RDH operation by the sender before image encryption. That means the issue of RDH in encrypted images is actually transformed into a traditional RDH in plaintext images. In summary, methods in both VRAE and VRBE categories are effective for RDH in encrypted images. However, there are some limitations. In VRAE RDH methods for encrypted images, estimation technique is necessary for the receiver, because no prior information of the original content is available except that he/she knows that the cover is a natural image. In traditional methods, LSB planes of the encrypted image are modified to accommodate the additional message, and the image recovery is based on estimating the original LSB planes with an assessment criterion, such as the "fluctuation function" in [8]-[10]. Because these estimations are not accurate enough, the recovery is only suitable for the case when a small amount of additional bits were embedded.

Although VRBE can achieve a higher payload, it requires that the sender must perform an extra RDH before image encryption. This may be impractical, in case the sender has no idea of the forthcoming data hiding by the data-hider, or he/she has no computational capability of the traditional RDH. On the other hand, in case the sender can reserve room for embedding by reversibly hiding redundant bits into the original plain image, all embedding tasks can also be done on the sender side and then the data-hider becomes redundant. In view of these problems, we propose a method to achieve high embedding payload by combining the MSB estimation with DSC. As estimating MSB is much more accurate than estimating LSB planes, the original data of the MSB plane can be recovered by DSC decoding with an acceptable decoding error probability. In other words, large embedding capacity can be achieved by this kind of combination.

III. SYSTEM DESCRIPTION

The proposed system is sketched in Fig. 1, which consists three phases: image encryption, data embedding, and data extraction/image recovery. In phase I, the sender encrypts the original image into an encrypted image using a stream cipher and an encryption key. In phase II, the data-hider selects and compresses some MSB of the secret image using LDPC codes to generate a spare space, and embeds additional bits into the encrypted image using an embedding key. In phase III, the receiver extracts the secret bits using the embedding key. If he/she has the encryption key, the original image can be approximately reconstructed via image decryption and estimation. When both the encryption and embedding keys are available, the receiver can extract the compressed bits, and implement the distributed source decoding using the estimated image as side information to perfectly recover the original image.

IV. IMAGE ENCRYPTION AND DATA EMBEDDING

A.Image Encryption Without loss of generality, we assume the original image O is a grayscale image with all pixel values falling into [0, 255], and the image size is M×N where both M and N are power of 2. First, the image owner turns the original image into plain bits by decomposing each pixel into 8 bits using b , , O⌊ = , 0,1, 2,...,7=mod 2, u ⌋2 u i j u i j (1) where Oi,j are pixels of the original image, 1≤i≤M, 1≤j≤N

The owner then chooses an encryption key KENC to generate pseudo-random bits using a stream cipher function (e.g., RC4 or SEAL), and encrypts the bitstream of the original image by ei, j,u ki, j,u⊕ bi, j,u = 0,1, 2,...,7 (2)=, u where ki,j,u are the key stream bits, ei,j,u the generated cipher text, denotes exclusive OR.⊕and Accordingly, the encrypted image E can be constructed by =∑ · = 7 0 , , , 2 u u i j i j u E e (3) Ei,j are pixel values of the encrypted image, 1≤i≤M, 1≤j≤N. Note that the stream cipher in (2) only scrambles pixel values but does not shuffle pixel locations. B. Data Embedding After image encryption, the content owner sends the encrypted image to the data-hider. To embed additional data into the image, the data-hider first decomposes the encrypted image E into four sub-images E (1) , E (2) , E (3) and E (4), each sized M/2×N/2. 1,2,..., / 2 1,2,..., / 2 , (,) (2 ,2) (,) (2 ,2 1) (,) (2 1,2) (,) (2 1,2 1) (4) (3) (2) (1) j N i M E i j E i j E i j E i j E i j E i j E i j E i j = = ⎪ ⎪ ⎩ ⎪ ⎪ ⎨ ⎧ = − − = − − = (4) E (k) (i,j) (k=1,…,4) are pixels of the sub-images. Fig. 2 shows an example, in which a 4×4 image is down-sampled to four 2×2 sub-images. Fig. 2 An example of image decomposing Bits of the three MSB planes of the sub-images E (2) , E (3) and E (4) are collected, resulting in a total of 3MN/4 bits. Using a selection key KSL, the data-hider pseudo-randomly selects L bits (1≤L≤3MN/4) from them and shuffles the selected bits. The shuffling is controlled by a shuffle key KSF. L/(3MN/4) the selection ratio, and divide the=Call α groups, each containing n bits. ⌋L/n⌊ =shuffled bits into K Denote the bits in each group as C(k,l) where k=1, 2, … , K and l = 1, 2, … , n. Next, the data-hider uses the Slepian-Wolf codes to compress the selected bits C.

V.DATA EXTRACTION AND IMAGE RECOVERY

On the receiver end, with the marked encrypted image, the hidden data can be extracted using the embedding key, and the original image can be approximately reconstructed using the encryption key, or losslessly recovered using both of the keys. Three cases are analyzed below in Subsections A, B and C respectively, in which the receiver has the embedding key only, the encryption key only, and both. We denote the received encrypted image containing secret data as V. A. Data Extraction In the first case, the receiver extracts the embedded secret data using the embedding key KEMB and the parameters PR=(L,n,r), where KEMB=(KSL,KSF,KSC). Divide V into four sub-images V (1) , V (2) , V (3) and V (4) using the same algorithm of Eq. (4). Collect all bits in the MSB planes of V (2) , V (3) and V (4) , and select L bits according to the selection key KSL. Shuffle the selected bits using the shuffle key KSF and divide the shuffled bits into K groups, each containing n bits. Denote the bits in each group as D(k,l) where k=1,2,…,K and l=1,2,…,n. For each group, extract the last (n–r) bits [D(k,r+1), D(k,r+2), …, D(k,n)]. Thus, the secret data can be reproduced by concatenating all the extracted bits from all K groups, and decrypted to the plaintext message using the key KSC. B.Image Decryption and Estimation In the second case, since the receiver has the encryption key but not the embedding key, he can reconstruct an approximate image based on image estimation.

Denote pixels in V as V'i,j (1≤i≤M, 1≤j≤N). Decompose V'i,j into 8 bits v'i,j,0, v'i,j,2, …., v'i,j,7 using Eq. (1), and decrypt the bits using stream decipher b' i, j,u v'= ki, j,u⊕i, j,u 0,1, 2,...,7 (11)=, u where ki,j,u are the key stream bits generated by the encryption key. Values of the decrypted image can be constructed from the deciphered bits using Eq. (3). We denote the decrypted image as A. Without the embedding key, the selected pixels that contain secret bits cannot be identified. To approximately recover the content, the receiver down-samples the decrypted image A into four sub-images A (1)

, A (2) , A (3) and A (4) according to Eq. (4). Pixel values in sub-image A (1) are the same as the pixel values in the original image, while part of the MSBs of the pixels in A (2) , A (3) and A (4) may differ from that in the original image. As a result, an estimation algorithm is defined to do the approximate reconstruction. The receiver generates a reference image B sized M×N from the sub-image A (1) using bilinear interpolation. Then, he/she down-samples B into four sub-images B (1) , B (2) , B (3) and B (4) . With A (k) and B (k) , the estimated sub-images A' (k) (k=2, 3, 4) can be generated using Eq. (12). Because the last seven LSBs in the sub-images A (k) (k=2, 3, 4) are unchanged during data hiding, the interpolated values in B (k) is used to optimize the MSB planes in A (k) . For each sub-image A (k) (k=2, 3, 4), on the (i,j)-th position ($1\le i\le M/2$, $1\le j\le N/2$), if the calculated value mod(A (k) (i,j),128)+128 is closer to the interpolated value B (k) (i,j) than mod(A (k) (i,j),128), the MSB of the (i,j)-th pixel is estimated to be 1; otherwise 0.

VI. DISCUSSION

The proposed method belongs to the VRAE category. Purpose of the present work is to improve the embedding payload in comparison with the other VRAE methods. To this end, the MSB plane is used as the cover to accommodate additional bits. As the error probability q of estimating MSB plane is much less than estimating the other planes, a much larger embedding payload can be achieved using DSC. During image reconstruction, although some noise would occur when directly decrypting the marked encrypted image, a provided estimation mechanism can be used to eliminate the noise. The proposed method is different from the method of [10]. In [10], some encrypted LSB planes are compressed using a linear transform to vacate embedding room, and the original image can be reconstructed by estimating the LSB planes. However, efficiency of compression and LSB plane estimation is insufficient, resulting in a low embedding payload. To increase the payload, we have used the MSB plane and the LDPC based DSC. As to the embedding payload, VRBE substantially outperforms the previous VRAE method because VRBE has the advantage of handling the plaintext image by the sender. When comparing with VRBE methods, embedding payload of the proposed method is not as large as that of [11], which in fact transforms the problem of RDH in encrypted image into the traditional RDH in the plaintext image. The flaws of [11] is that the sender is partly involved in the embedding computation, supposing he/she has the prior knowledge that a data embedding is to be done by the server. This might not be practical in case the sender has no idea about the possible data hiding, or has no computation capability other than image encryption. On the other hand, if the sender can do RDH himself, all tasks can be done on the sender side and the data-hider becomes redundant. Instead, embedding of the proposed method is entirely realized in the encrypted domain, and a high payload is achieved with the association of DSC decoding computation on the receiver side. Two aspects of data security need to be considered: security of the image content and security of the additional message. The content owner does not allow the service provider to access the image content, and the data-hider does not allow adversaries to crack the embedded message. For the content owner, the original image is encrypted with a stream cipher algorithm using an encryption key KENC. For the data-hider, the additional bits are also encrypted with the stream cipher using another key KSC. Many stream cipher algorithms can be used, such as RC4 or SEAL. It is infeasible for the adversary to execute an exhaustive search if appropriate keys are used, e.g., the seed keys no less than128 bits in RC4, or no less than 160 bits in SEAL. Thus, security of the image content and the additional message can be guaranteed. The extraction security can also be guaranteed. On the data-hider side, L MSB bits are selected from 3MN/4 pixels using a selection key KSL, and shuffled by a shuffle key KSF. After compressing the shuffled L bits using DSC and appending the encrypted additional bits to the end of the compressed bits, a further reshuffle is done with the key KSF to generate the marked L bits that are put into the original MSB positions. For an adversary without both the embedding and encryption keys, the success rate of correctly extracting the hidden bits is as low as 1 (!) C3 / 4 L L ·NM . Supposing the adversary has the encryption key, after directly decrypting the marked encrypted image, the amount of pixels containing noises is about L/2. With the only L/2 MSB bits, the success rate of correctly extracting the additional bits is as low as 1 (!) / 2 C3 / 4 / 2 L L NM L – · . Therefore, exhaustive searching is virtually impossible for the adversaries to extract the additional message provided M, N and L are not too small. Data extraction and image recovery is separable in the proposed method. Solutions to three different cases on the receiver side are provided: with the encryption key only, with the embedding key only, and with both the encryption and embedding keys. In the previous VRAE methods, when the receiver acquires the embedding key later, he/she still cannot extract the hidden message from the decrypted image directly, unless the marked encrypted version has been saved (this is likely the case) and is used, or extra re-encryption is carried out using the encryption key to regenerate a marked encrypted copy. In the proposed method, the receiver can save the MSB bits of the marked encrypted image before estimating the original image. After the embedding key later is acquired later, he/she can extract the additional bits from the save bits.

VII.CONCLUSION

This paper proposes a scheme of reversible data hiding in encrypted images using distributed source coding. After encrypting the original image with a stream cipher, some bits of MSB planes are selected and compressed to make room for the additional secret data. On the receiver side, all hidden data can be extracted with the embedding key only, and the original image approximately recovered with high quality using the encryption key only. When both the embedding and encryption keys are available to the receiver, the hidden data can be extracted completely and the original image recovered perfectly.

With the idea of DSC, the proposed method substantially increases the payload as compared with the existing VRAE methods. An LDPC parity-check matrix is used to generate corresponding syndromes. Associated with the estimated image generated from the proposed estimation algorithm, the receiver can decode these syndromes back to the original bits using iterative BPA decoding.

Because embedding operations are performed to the encrypted data, the data-hider cannot access the contents of the original image. That ensures security of the contents in data hiding. As the embedding and recovery are protected by the encryption and embedding keys, an adversary is unable to break into the system without these keys.

REFERENCES

[1] Z. Erkin, A. Piva, S. Katzenbeisser, et al., "Protection and retrieval of encrypted multimedia content: When cryptography meets signal processing," EURASIP Journal on Information Security 2007, 2008.
[2] M. Johnson, P. Ishwar, V. M. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," IEEE Trans. Signal Process., vol. 52, no. 10, pp. 2992–3006, Oct. 2004.
[3] W. Liu, W. Zeng, L. Dong, and Q. Yao, "Efficient compression of encrypted grayscale images," IEEE Trans. Image Process., vol. 19, no. 4, pp. 1097–1102, Apr. 2010.
[4] X. Zhang, G. Feng, Y. Ren and Z. Qian, "Scalable Coding of Encrypted Images," IEEE Trans. Inform. Forensics Security, vol. 21, no. 6, pp.3108-3114, June 2012.
[5] M. Deng, T. Bianchi, A. Piva, and B. Preneel, "An efficient buyer-seller watermarking protocol based on composite signal representation," in Proc. 11th ACM Workshop Multimedia and Security, 2009, pp. 9–18.
[6] S. Lian, Z. Liu, Z. Ren, and H. Wang, "Commutative encryption and watermarking in video compression," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 6, pp. 774–778, Jun. 2007.
[7] W. Puech, M. Chaumont and O. Strauss, "A reversible data hiding method for encrypted images," Proc. SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, 68191E, Feb. 26, 2008, doi:10.1117/12.766754.
[8] X. Zhang, "Reversible data hiding in encrypted images," IEEE Signal Process. Lett., vol. 18, no. 4, pp. 255–258, Apr. 2011.
[9] W. Hong, T. Chen, and H. Wu, "An improved reversible data hiding in encrypted images using side match," IEEE Signal Process. Lett., vol. 19, no. 4, pp. 199–202, Apr. 2012.
[10] X. Zhang, "Separable reversible data hiding in encrypted image," IEEE Trans. Inf. Forensics Security, vol. 7, no. 2, pp. 826–832, Apr. 2012.
[11] K. Ma, W. Zhang, et al. "Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption," IEEE Trans. Inf. Forensics Security, vol. 8, no. 3, 553-562, 2013.
[12] Z. Qian, X. Han and X. Zhang, "Separable Reversible Data hiding in Encrypted Images by n-nary Histogram Modification," 3rd International Conference on Multimedia Technology (ICMT 2013), pp. 869-876, Guangzhou, China, 2013.
[13] W. Zhang, K. Ma and N. Yu, "Reversibility improved data hiding in encrypted images," Signal Processing, vol. 94, pp. 118–127, 2014.
[14] T. Kalker and F. M. Willems, "Capacity bounds and code constructions for reversible data-hiding," in Proc. 14th Int. Conf. Digital Signal Processing (DSP2002), 2002, pp. 71–76.
[15] J. Fridrich and M. Goljan, "Lossless data embedding for all image formats," in Proc. SPIE Proc. Photonics West, Electronic Imaging, Security and Watermarking of Multimedia Contents, San Jose, CA, USA, Jan. 2002, vol. 4675, pp. 572–583.
[16] J. Tian, "Reversible data embedding using a difference expansion," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 8, pp. 890–896, Aug. 2003.
[17] Z. Ni, Y. Shi, N. Ansari, and S. Wei, "Reversible data hiding," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 3, pp. 354–362, Mar. 2006.
[18] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," IEEE Trans. Image Process., vol. 16, no. 3, pp. 721–730, Mar. 2007.
[19] L. Luo et al., "Reversible image watermarking using interpolation technique," IEEE Trans. Inf. Forensics Security, vol. 5, no. 1, pp. 187–193, Mar. 2010.
[20] X. L. Li, B. Yang, and T. Y. Zeng, "Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection," IEEE Trans. Image Process., vol. 20, no. 12, pp. 3524–3533, Dec. 2011.
[21] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," IEEE Trans. Inform. Theory, vol. IT-19, pp. 471–480, July 1973.
[22] T. M. Cover and J. A. Thomas, Elements of Information Theory. New York: Wiley, 1991.
[23] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," IEEE Trans. Inform. Theory, vol. 49, pp. 626–643, Mar. 2003.
[24] W. Liu, W. Zeng, L. Dong, et al. "Efficient compression of encrypted grayscale images," IEEE Trans. on Image Processing, vol. 19, no. 4, pp. 1097-1102, 2010.
[25] W. E. Ryan, "An introduction to LDPC codes," in CRC Handbook for Coding and Signal Processing for Recoding Systems (B. Vasic, ed.), CRC Press, 2004.
[26] A. D. Liveris, Z. Xiong and C. N. Georghiades. "Compression of binary sources with side information at the decoder using LDPC codes," IEEE Communications Letters, vol. 6, no. 10, pp. 440-442, 2002.
[27] D. Varodayan, A. Aaron and B. Girod. "Rate-adaptive codes for distributed source coding," Signal Processing, vol. 86, no.11, pp. 3123-3130, 2006.
[28] G. Schaefer and M. Stich, "UCID: An Uncompressed Colour Image Database," in Proc. SPIE: Storage and Retrieval Methods and Applications for Multimedia, 2004, vol. 5307, pp. 472–480.