IEEE ISIE 2005, June 20-23, 2005, Dubrovnik, Croatia

Efficient Encryption of Compressed Color Images

Karl Martin*, Rastislav Lukac*, Konstantinos N. Plataniotis*
*University of Toronto, Department of Electrical and Computer Engineering, Toronto, Canada

***Abstract*— An efficient, secure color image coder based on Color Set Partitioning in Hierarchical Trees (C-SPIHT) compression and partial encryption is presented. Confidentiality of the image data is achieved by encrypting only the significance bits of individual wavelet coefficients for K iterations of the C-SPIHT algorithm. The use of a stream cipher and the encryption of a small number of bits keeps computational demands at a minimum and makes the technique suitable for hardware implementation. By varying K, the level of confidentiality vs. processing overhead can be controlled. It was found that using $K = 2$ achieved an adequate level of security for test images coded at 0.8 bpp, resulting in an average of only 0.33% of the coded image being encrypted.**

I. Introduction

Security in communication systems has become increasingly important in recent times. The Internet has become a hostile environment with both wired and wireless channels offering no inherent assurance of confidentiality. Strong encryption schemes, such as the Advanced Encryption Standard (AES), have been designed to provide confidentiality for arbitrary binary data [1]. However, communications have become increasingly multimedia in nature and such strong encryption schemes do not take into account the special characteristics of multimedia data and the way in which they are accessed.

Images and video are typically large in size compared to text and audio, and often already consume significant computational resources at both the source and receiver for coding and decoding, respectively. Also, applications such as remote surveillance may involve the streaming of sensitive visual image data over untrusted networks. Confidentiality may be required, but blindly applying a strong encryption scheme such as AES would demand a prohibitive amount of computational resources for the large volume of real-time data. Other applications, such as online collaboration, may involve the use of power-limited mobile devices, such as mobile phones and personal digital assistants (PDAs) with embedded imaging capabilities, forming ad-hoc wireless networks. Most of the computational resources of the devices are dedicated to the coding and decoding of the visual data, making the application of schemes such as AES exceedingly difficult or impossible [2].

In this paper, we present a new efficient method for encrypting still color images based on the principle of *partial* or *selective encryption* [2], [3], [4]. This methodology combines compression with encryption so that correlations between the pixels in the input image are removed before encryption. In this way, the coding algorithm can be used to direct the encryption on only the most significant information, hence minimizing the computational overhead required for encrypting and making efficient use of resources.

The proposed scheme relies on the color set partitioning in hierarchical trees (C-SPIHT) [5] compression algorithm and a stream cipher to produce a secure, coded image. The C-SPIHT coder directs a stream cipher to encrypt only certain significance bits, thus requiring minimal computational overhead. The rate-distortion performance of the compression algorithm is unaffected since specific bits of the image are encrypted *after* encoding. This novel scheme is more efficient than previously published schemes, such as [4], since encryption is only performed on the significance bits of individual wavelet coefficients and not trees. Also, the level of confidentiality versus computational overhead may be controlled via a parameter which determines how many coding iterations for which the encryption should be performed. The confidentiality and protection of the image results not only from the encryption of the significance bits, but also from the unknown arrangement of encrypted and unencrypted bits. This scheme can be used in applications such as remote surveillance and online collaboration, where large volumes of streaming visual data may be transmitted over untrusted networks and the computational resources available to devices are limited.

The paper is organized as follows: Section II provides relevant background on the SPIHT and C-SPIHT coding schemes as well as the partial encryption paradigm. Section III describes the proposed efficient secure coder and Section IV provides some concluding remarks.

II. Background

A. Set Partitioning in Hierarchical Trees

The Set Partitioning in Hierarchical Trees (SPIHT) [6] wavelet-based compression algorithm is well regarded as a highly efficient technique for lossy image compression. It is based on the concept of zerotrees, whereby an image is transformed using an L-level discrete wavelet transform (DWT) and the resulting coefficients are grouped into spatial orientation trees (SOT) and coded using successive approximation quantization. The SOT structure is shown in Fig. 1. The SPIHT coder produces an *embedded* code which allows the decoding of the resulting bit-stream to be terminated at any point to produce an arbitrary bit-rate. Thus, the output bit-stream can be decoded progressively to provide rate-distortion scalability.

0-7803-8738-4/05/$20.00

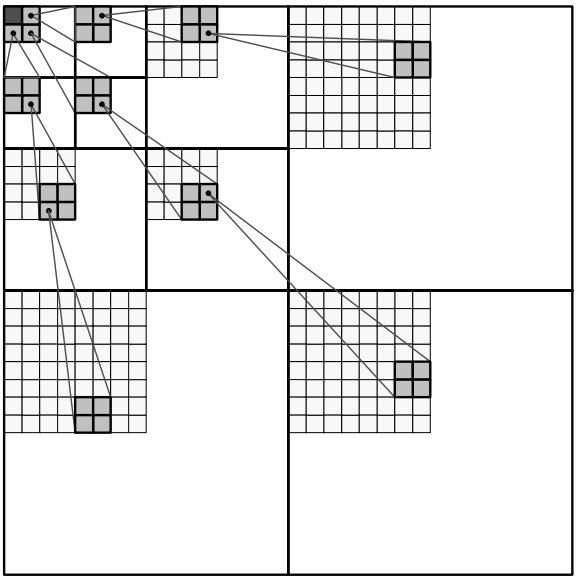


Fig. 1. Parent-child structure of wavelet coefficients in the SPIHT algorithm.

The algorithm employs three ordered lists: the *list of insignificant pixels* (LIP); the *list of insignificant sets* (LIS); and the *list of significant pixels* (LSP). The LIP is initialized with the coefficients from the LL subband of the wavelet decomposition; the LIS is initialized with all the SOTs; and the LSP is initialized to be empty. The coding is performed by iterating a successive approximation quantization procedure whereby a significance threshold T_k is reduced by a factor of 2 at each iteration k, for $k = 0, 1, 2, \ldots$. A coefficient $c_{i,j}$ at location (i, j) is deemed to be *significant* when $|c_{i,j}| \geq T_k$.

At each iteration k, the procedure has two main passes: a *sorting pass* and a *refinement pass*. In the first phase of the sorting pass, the elements in the LIP are assessed for significance at the current quantization threshold T_k. Significant coefficients are moved to the LSP and insignificant coefficients remain in the LIP. The significance information and sign of the significant coefficients is output directly into the bit-stream. In the second phase of the sorting pass, the SOTs in the LIS are tested to check if they contain any significant coefficients. The SOTs that contain no significant coefficients are called *zerotrees* and placed back into the LIS; the SOTs that contain significant coefficients are partitioned according to a well defined recursive procedure and the new smaller SOTs are tested for significance. The partitioning procedure continues until the significant coefficients from all the SOTs are isolated and moved to the LSP. The significance information for the trees and individual isolated coefficients is output directly to the bit-stream, as well as the sign of the coefficients found significant.

Once the two phases of the sorting pass are completed, the refinement pass involves traversing the LSP and transmitting the $(k-p)^{th}$ most significant bit (MSB) of all the coefficients in the list, where p signifies the iteration at which the coefficient was originally found significant. Upon completion of the refinement pass, the threshold is updated such that $T_{k+1} = T_k/2$, and the procedure repeats. The result is an efficient, ordered bit-plane coding of the wavelet transformed image. The algorithm can be terminated at any point during any iteration to achieve a particular desired bit-rate or image distortion.

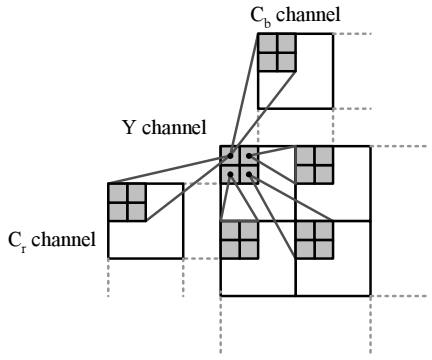


Fig. 2. Root parent-child structure in the C-SPIHT algorithm.

To decode an image coded using SPIHT, the same algorithm is followed except that each bit from the bit-stream is read and the sorting and refinement of coefficients is replicated. The coefficients are successively reconstructed with increased precision as the significance, sign, and refinement information is decoded. This makes the codec symmetric with the coding and decoding execution path being identical. The decoder must simply be initialized with the original image size and the number of DWT levels L.

B. Color-SPIHT

The SPIHT coding algorithm was not designed for multi-channel images such as RGB color images [5]. The Color-SPIHT (C-SPIHT) algorithm of [5] uses SPIHT with a modified SOT structure to allow efficient coding of natural RGB images. The image is first transformed into a decorrelated color space, such as $\mathrm{YC_bC_r}$, then each channel is transformed using a DWT. The modified SOT structure, shown in Fig. 2, uses one in every four luminance (Y) channel LL subband coefficients as the root of a tree containing $\mathrm{C_b}$ and $\mathrm{C_r}$ chrominance channel coefficients. The rest of the tree structures are the same as in SPIHT. The net result is that the chrominance channel coefficients, comprising 2/3 of the total number of coefficients but typically having less energy than the luminance channel coefficients, are contained in only 1/4 of the SOTs. In this way, fewer bits can be used to code their insignificance in the initial iterations of the coding algorithm.

As in SPIHT, the C-SPIHT algorithm maintains the LIP, LIS, and LSP ordered lists.

C. Partial Encryption

Compression aids encryption by removing redundancies in the data and separating the data according to their significance. Hence, *partial* or *selective* encryption is the technique of securing the confidentiality of compressed data by encrypting only a fraction of the total data in order to reduce computational overhead [2], [3], [4]. 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. In [4], partial encryption of quadtree based compressed images and SPIHT compressed images was demonstrated. Using SPIHT with grayscale images, it was proposed that only the significance bits (both for individual coefficients and trees) of the top two tree levels, along with the initial quantization threshold, be encrypted. The number of bits encrypted is variable depending on the image content. Grayscale test images of size 512×512 for screen-display applications were encoded at 0.8 bpp and it was found that less than 2% of the coded source was typically encrypted. It was claimed in [4] that confidentiality is achieved not just through securing the most significant information, but by making the correct state of the decoder very difficult to determine.

A different approach has been proposed in [7], where 3D-SPIHT coded video is encrypted using two stages of coefficient confusion where each set of wavelet coefficients is scrambled both within a processing window cube and between cubes *before* compression. Also, the signs of the low frequency coefficients are encrypted using a chaotic stream cipher. The computational cost of encryption was found to be less than 10% of compression if the processing windows were no bigger than $32\times32\times32$ and the DWT was performed using 4 levels or less. In [8], random permutations of wavelet coefficients *before* coding using SPIHT or JPEG2000, as a means of fast encryption, was studied. However, the authors found that this significantly reduced compression performance (up to 27% for test images).

III. Novel Secure Coder for Efficient Color Image Encryption

Let us denote the C-SPIHT bit-stream as the ordered set of bits B. The bit-stream can be divided into the ordered subsets $B = \{B_0, B_1, B_2, \ldots\}$, where B_k is the set of bits obtained during iteration k of the C-SPIHT algorithm. Each B_k can be further subdivided into $B_k = \{B_{k,\mathrm{LIP}}, B_{k,\mathrm{LIS}}, B_{k,\mathrm{LSP}}\}$, where $B_{k,\mathrm{LIP}}$ denotes the ordered set of bits obtained during the first phase of the sorting pass where coefficients in the LIP are tested for significance; $B_{k,\mathrm{LIS}}$ denotes the ordered set of bits obtained during the second phase of the sorting pass where entire trees are tested for significance; and $B_{k,\mathrm{LSP}}$ denotes the ordered set of bits obtained during the refinement pass. This decomposition of the bit-stream is shown pictorially in Fig. 3.

As shown in Fig. 4, the set of bits $B_{k,\mathrm{LIP}}$ can be further decomposed into the interspersed sets of significance bits $B_{k,\mathrm{LIP-sig}}$ and sign bits $B_{k,\mathrm{LIP-sgn}}$. Similarly, each set of bits $B_{k,\mathrm{LIS}}$ is composed of significance bits $B_{k,\mathrm{LIS-sig}}$ and sign bits $B_{k,\mathrm{LIS-sgn}}$ for individual coefficients, as well as significance bits $B_{k,\mathrm{LIS-Tsig}}$ for trees.

The proposed efficient encryption scheme uses an encryption function $f_E(\cdot)$ to encrypt only the bits $B_{k,\mathrm{LIP-sig}}$ and $B_{k,\mathrm{LIS-sig}}$, for $k = 0, 1, \ldots, K-1$, where the parameter K is controlled by the user during the

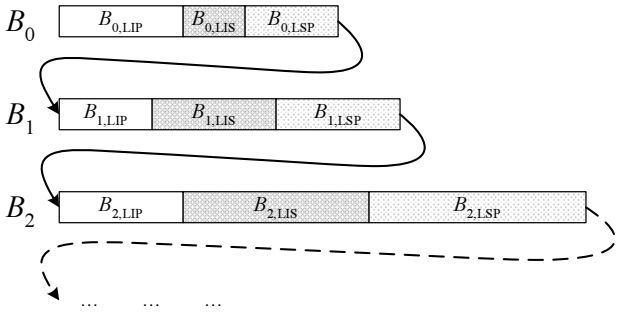


Fig. 3. C-SPIHT bitstream.

encryption/encoding procedure to determine the number of coding iterations to be encrypted. That is, only the coefficient significance bits obtained during the first K iterations of the C-SPIHT algorithm are encrypted. The sign bits $B_{k,\mathrm{LIP-sgn}}$ and $B_{k,\mathrm{LIS-sgn}}$ remain unencrypted, as do the tree significance bits $B_{k,\mathrm{LIS-Tsig}}$. While the bits $B_{k,\mathrm{LIS-Tsig}}$ represent significance information, they are not encrypted because their values are determined by the significance of one or more unknown coefficients within a large group of coefficients. This can be considered non-specific information, not directly affecting reconstruction values and hence not requiring encryption.

The combined coding and encryption system is shown in Fig. 5(a). The encryption function $f_E(\cdot)$ is implemented using a symmetric key stream cipher. A block cipher cannot be used because the bits need to be decrypted individually to determine the execution path of the decoder, which in turn specifies the location of the subsequent bits to be decrypted before decoding. The use of a stream cipher also provides two significant advantages: i) it generally require less computational and memory resources than block ciphers, making it more suitable for hardware implementation, and ii) it does not contribute to error propagation.

The proposed scheme can be viewed as partial bit-plane encryption in the wavelet domain. Only the higher order bit-planes of the DWT coefficients found significant during the initial K iterations of the C-SPIHT algorithm are encrypted. However, confidentiality is achieved not only by encrypting these high-order bit-planes of the most significant coefficients, but also by making the correct interpretation of the unencrypted portions of the bit-stream difficult. This results from the nature of the C-SPIHT decoding algorithm which involves a sequence of binary

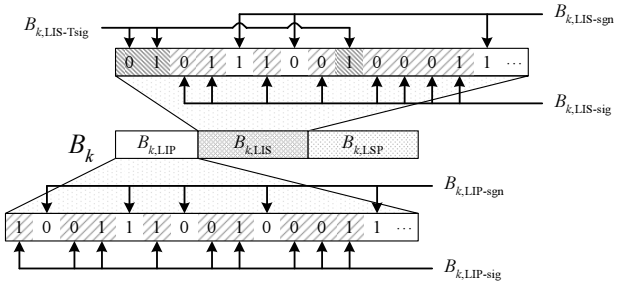


Fig. 4. Composition of subset B_k of C-SPIHT bit-stream.

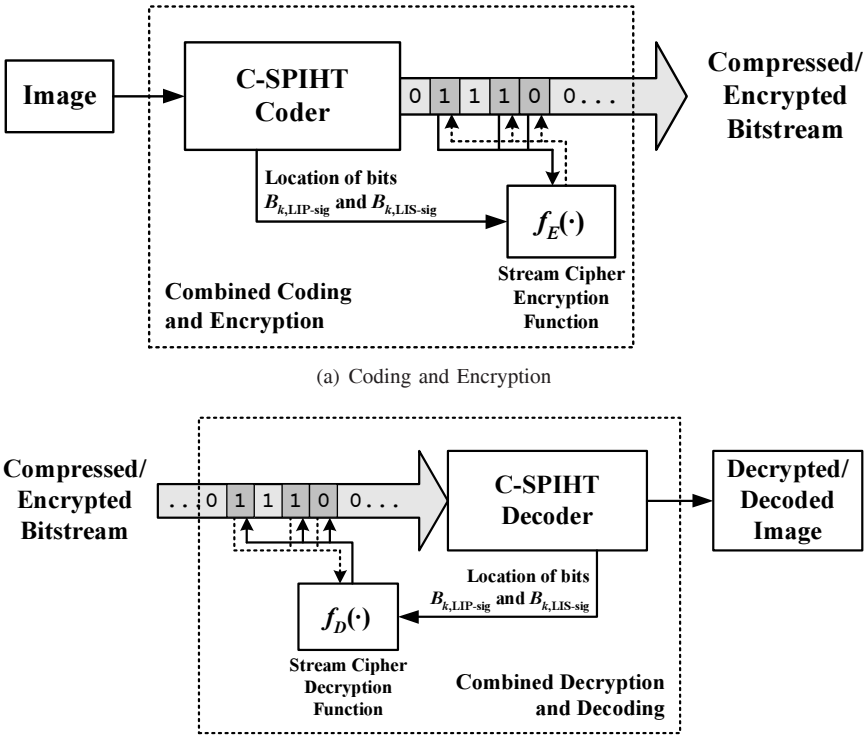


Fig. 5. System level diagram of coding and encryption (a), and decryption and decoding (b). Dark gray bits represent encrypted significance bits.

decisions based on the bit-stream data. The determination of whether a particular bit in the bit-stream is a significance or sign bit, and which DWT coefficient or tree it corresponds to, requires the correct interpretation of all the previous bits. By making the initial, most significant portion of the bit-stream confidential via encryption, the remaining unencrypted bits cannot be easily intepreted.

Decryption and decoding is achieved in a similar manner (Fig. 5(b)) with only the significance bits $B_{k,\text{LIP}-\text{sig}}$ and $B_{k,\text{LIS}-\text{sig}}$, for $0 \leq k < K$, requiring decryption at the appropriate points in the C-SPIHT algorithm. The decryption function $f_D(\cdot)$ of the stream cipher uses the secret key to perform the actual decryption. Each decrypted bit must be passed to the decoder before the decryption continues so that the decoder can instruct the stream cipher which bit to decrypt next. It is assumed that the transmission channel for the encrypted/coded image is error free.

To demonstrate the difficulty encountered by a cryptanalyst attempting to determine which bits are unencrypted, we use $b^j_{k,\text{LIP}}$ to denote the j^{th} bit in the set $B_{k,\text{LIP}}$, for $j = 0, 1, 2, \ldots, N_{k,\text{LIP}} - 1$, where $N_{k,\text{LIP}}$ is the total number of bits in $B_{k,\text{LIP}}$. It is known a priori that the first bit is a significance bit:

$$b^0_{k,\text{LIP}} \in B_{k,\text{LIP}-\text{sig}},\ \forall k < K. \tag{1}$$

However, the classification of the second bit depends on the value of $b^0_{k,\text{LIP}}$:

$$b^1_{k,\text{LIP}} \in \begin{cases} B_{k,\text{LIP}-\text{sig}} & \text{if } b^0_{k,\text{LIP}} = 0 \\ B_{k,\text{LIP}-\text{sgn}} & \text{he ı e} \end{cases} \tag{2}$$

This can be generalized for $1 \leq j < N_{k,\text{LIP}}$ as follows:

$$b^j_{k,\text{LIP}} \in B_{k,\text{LIP}-\text{sgn}} \tag{3}$$

if $b^{j-1}_{k,\text{LIP}} \in B_{k,\text{LIP}-\text{sig}}$ and $b^{j-1}_{k,\text{LIP}} = 1$. Otherwise,

$$b^j_{k,\text{LIP}} \in B_{k,\text{LIP}-\text{sig}} \tag{4}$$

From (3) and (4), it is evident that the cryptanalyst must correctly interpret all previous bits $b^l_{k,\text{LIP}}$, $0 \leq l < j$ in order to determine the classification of $b^j_{k,\text{LIP}}$ and, hence, whether it is unencrypted. Without the decryption key, not only do the encrypted bits remain confidential, but the locations of the unencrypted bits within B cannot be determined and are thus also confidential. The situation is even more difficult for $1 \leq k < K$ since the beginning location of $B_{k,\text{LIP}}$ within B will also be unknown.

A similar argument can be made for the bits in $B_{k,\text{LIS}}$. We use $b^j_{k,\text{LIS}}$ to denote the j^{th} bit in the set $B_{k,\text{LIS}}$, for $j = 0, 1, 2, \ldots, N_{k,\text{LIS}} - 1$, where $N_{k,\text{LIS}}$ is the total number of bits in $B_{k,\text{LIS}}$. The C-SPIHT algorithm employs two different types of trees in the LIS: type "A" and type "B". When a type "A" tree is found to have a significant coefficient, it's highest 4 children coefficients are tested for significance before the tree is partitioned into 4 trees. On the other hand, a type "B" tree is partitioned directly without testing individual coefficients. Hence, the set of bits $B_{k,\text{LIS}-\text{Tsig}}$ can be subclassified into $B_{k,\text{LIS}-\text{TsigA}}$ or $B_{k,\text{LIS}-\text{TsigB}}$, depending on whether the significance bit pertains to a type "A" or "B" tree, respectively.

It is known a priori that the first bit is a tree significance bit, $b^0_{k,\text{LIS}} \in B_{k,\text{LIS}-\text{Tsig}},\ \forall k < K$. For $k = 0$ it is also

TABLE I

SUMMARY OF CLASSIFICATION RULES FOR $b^j_{k,\mathrm{LIS}}$ IN THE CASE OF A TYPE "A" TREE OR COEFFICIENT FROM TYPE "A" TREE. "DC" DENOTES THE "DON'T CARE" CONDITION.

$b^j_{k,\mathrm{LIS}}$ Class	$b^{j-1}_{k,\mathrm{LIS}}$ Class	Value	Other Conditions
$B_{k,\mathrm{LIS-Tsig}}$	$B_{k,\mathrm{LIS-Tsig}}$	0	DC
	$B_{k,\mathrm{LIS-sig}}$	0	4 coefficient significance bits have been encountered for the current tree
	$B_{k,\mathrm{LIS-sgn}}$	DC	4 coefficient significance bits have been encountered for the current tree
$B_{k,\mathrm{LIS-sig}}$	$B_{k,\mathrm{LIS-Tsig}}$	1	DC
	$B_{k,\mathrm{LIS-sig}}$	0	1 to 3 coefficient significance bits have been encountered for the current tree
	$B_{k,\mathrm{LIS-sgn}}$	DC	1 to 3 coefficient significance bits have been encountered for the current tree
$B_{k,\mathrm{LIS-sgn}}$	$B_{k,\mathrm{LIS-sig}}$	1	DC

known that $b^0_{k,\mathrm{LIS}} \in B_{k,\mathrm{LIS-TsigA}}$. In general, for any k with $b^0_{k,\mathrm{LIS}} \in B_{k,\mathrm{LIS-TsigA}}$, the classification of the second bit is as follows:

$$b^1_{k,\mathrm{LIS}} \in \begin{cases} B_{k,\mathrm{LIS-sig}} & \text{if } b^0_{k,\mathrm{LIS}} = 1 \\ B_{k,\mathrm{LIP-Tsig}} & \text{he ı e} \end{cases} \tag{5}$$

However, for $b^0_{k,\mathrm{LIS}} \in B_{k,\mathrm{LIS-TsigB}}$, the classification of the second bit is always $b^1_{k,\mathrm{LIS}} \in B_{k,\mathrm{LIS-Tsig}}$.

For the case of $b^0_{k,\mathrm{LIS}} \in B_{k,\mathrm{LIS-TsigA}}$ and $b^0_{k,\mathrm{LIS}} = 1$, the third bit is classified as follows:

$$b^2_{k,\mathrm{LIS}} \in \begin{cases} B_{k,\mathrm{LIS-sig}} & \text{if } b^1_{k,\mathrm{LIS}} = 0 \\ B_{k,\mathrm{LIS-sgn}} & \text{he ı e} \end{cases} \tag{6}$$

The rule described by (6) continues to hold for subsequent bits until 4 coefficient significance bits are encountered.

The generalization of the classification rules for $b^j_{k,\mathrm{LIS}}$ for $1 \le j < N_{k,\mathrm{LIS}}$ is complex, involving two basic cases and several subcases:

1) Case I: Bit $b^j_{k,\mathrm{LIS}}$ pertains to a type "A" tree or coefficient from a type "A" tree. A summary of the classification rules for this case are shown in Table I. The first column shows the classification of the current bit $b^j_{k,\mathrm{LIS}}$ with the remaining columns showing the various combinations of requisite conditions. The second and third columns show the classification and value of the previous bit $b^{j-1}_{k,\mathrm{LIS}}$, respectively. The fourth column shows any other requisite conditions. "DC" is used to denote the "Don't Care" condition.

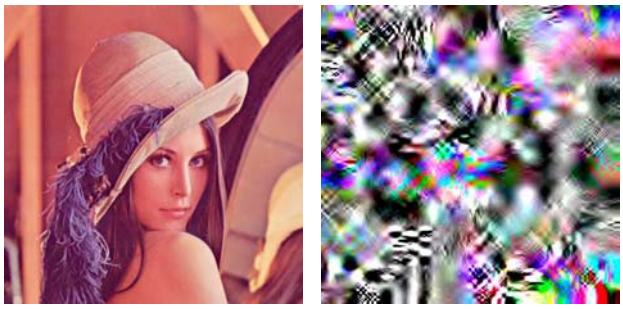
2) Case II: Bit $b^j_{k,\mathrm{LIS}}$ pertains to a type "B" tree, then we always have $b^j_{k,\mathrm{LIS}} \in B_{k,\mathrm{LIS-Tsig}}$.

It can be seen from the general rules described above that the classification of the bit $b^j_{k,\mathrm{LIS}}$ requires the correct interpretation of all previous bits $b^l_{k,\mathrm{LIS}}$, $0 \le l < j$ as well as the correct state of LIS in order to determine which of the two main cases is in effect. The correct state of the LIS cannot be determined without knowledge of all the previous bits from the *entire* bit-stream B.

In summary, the proposed encryption scheme achieves confidentiality in two ways: i) encryption of the most significant information of individual wavelet coefficients; and ii) making the correct state of the decoder very difficult to determine. This differs from the scheme in [4] in three important ways: i) the bits $B_{k,\mathrm{LIP-sig}}$ and $B_{k,\mathrm{LIS-sig}}$ are encrypted regardless of which subband the particular coefficients being considered reside in; ii) computational overhead is reduced by not encrypting $B_{k,\mathrm{LIS-Tsig}}$; and iii) the level of confidentiality achieved versus computational overhead can be controlled via the parameter K. Property i) of the proposed scheme means that no a priori decisions are made as to which particular coefficients should encrypted; the scheme in [4] limits encryption to the coefficients in the top two levels of the trees. In the scheme proposed here, the decision is made by the C-SPIHT coder via the successive approximation quantization, hence encrypting the coefficients which represent the most significant features of the image regardless of which subbands they reside in.

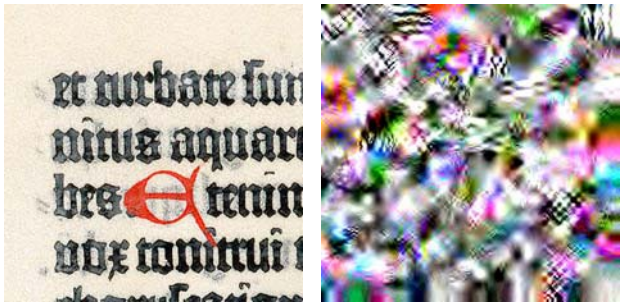
It should also be noted that protection of the image depends on two important factors: i) the number of encrypted bits being large enough to make an exhaustive search over all possible bit combinations computationally infeasible; and ii) the inability of the cryptanalyst to locate the beginning of the unencrypted block, $b^0_{k,\mathrm{LIP}}$ for $k = K$ and correctly decode from that point on. The first factor requires that K be larger than a certain image-dependent threshold value to achieve a minimum number of required encrypted bits. The second factor requires that the number possible locations of the bit $b^0_{k,\mathrm{LIP}}$ for $k = K$ as well as the number of possible states of the lists be too large for an exhaustive search. Further analysis of these two factors is beyond the scope of this paper. However, if K is large enough, the unencrypted block will not contain any significant image features, making the second factor irrelevant.

A set of test images and their encrypted versions for $K = 2$ are shown in Fig. 6. The images were coded using C-SPIHT with a 5-level DWT using the CDF 9/7 biorthogonal wavelet filters. The images on the right are decoded without knowledge of the encryption key. Clearly, no features of the images are revealed and confidentiality is ensured. Table II lists the number of bits encrypted for each test image for $K = 1, 2, \ldots, 6$. The secure coder performed better for the natural images "Lena" and "Tulip", generally requiring fewer encrypted bits than for the artificial images "Map" and "Document", for $K \ge 4$. This reflects the characteristics of C-SPIHT, which performs better with natural visual data. With $K = 2$, an adequate level of confidentiality could be achieved with only an average 0.33% of the bits encrypted for the test images coded at a typical representative example bit-rate of 0.8 bpp. For a given image and choice of K, the number of encrypted bits will remain constant regardless of the final coded image bit-rate. Hence, the percentage encrypted bits will increase or decrease accordingly if the final coded bit-rate is smaller or larger, respectively.



(a) Lena

(b) Map



(c) Document

(d) Tulip

Fig. 6. Original images (left) and encrypted images using $K = 2$ (right). All images are 512×512 in size. The encrypted images are decoded without decrypting.

TABLE II

THE NUMBER OF BITS ENCRYPTED FOR THE TEST IMAGES USING DIFFERENT VALUES OF K. THE PERCENTAGE OF BITS ENCRYPTED FOR THE TEST IMAGES CODED AT 0.8 BPP IS SHOWN IN BRACKETS.

	Test Image			
K	Lena	Map	Document	Tulip
1	256 (0.1%)	256 (0.1%)	260 (0.1%)	256 (0.1%)
2	990 (0.5%)	619 (0.3%)	780 (0.4%)	428 (0.2%)
3	1718 (0.8%)	1776 (0.9%)	2074 (1.0%)	844 (0.4%)
4	2898 (1.4%)	8615 (1.5%)	7053 (3.4%)	1591 (0.8%)
5	6061 (2.9%)	27594 (13.2%)	20721 (9.9%)	3048 (1.5%)
6	13527 (6.5%)	95111 (45.4%)	54444 (26.0%)	5683 (2.7%)

IV. CONCLUSION

An efficient, secure color image coder was proposed, employing the C-SPIHT compression algorithm and a stream cipher. The scheme uses the C-SPIHT algorithm to direct the stream cipher to encrypt only the significance bits for individual coefficients encountered during the first K sorting passes of the C-SPIHT algorithm. Confidentiality of the image data is ensured by encrypting the most significant information and by making the unencrypted information difficult to locate within the bit-stream B. The user may control the confidentiality versus processing overhead by choosing K at the time of encrypting/encoding. The proposed scheme may be used in applications such as remote surveillance and online collaboration where the C-SPIHT image coder is already implemented. The added logic and the use of a stream cipher make the scheme appropriate for implementation in hardware embedded in consumer and industrial electronic devices. The principles behind the proposed scheme may also be applied to other embedded image and video coders such as JPEG2000, Motion JPEG2000, and 3D-SPIHT.

ACKNOWLEDGMENT

The work of the first author is partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) under the Network for Effective Collaboration Technologies through Advanced Research (NECTAR) project.

REFERENCES

[1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[2] T. Lookabaugh and D. C. Sicker, "Selective encryption for consumer applications," *IEEE Commun. Mag.*, vol. 42, no. 5, pp. 124–129, May 2004.

[3] K. Martin, R. Lukac, and K. N. Plataniotis, "Efficient encryption of wavelet-based coded color images," *Pattern Recognition*, to appear, vol. 38, 2005.

[4] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Processing*, vol. 48, no. 8, pp. 2439–2451, Aug. 2000.

[5] A. A. Kassim and W. S. Lee, "Embedded color image coding using SPIHT with partially linked spatial orientation trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 2, pp. 203–206, Feb. 2003.

[6] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[7] S. Lian, J. Sun, and Z. Wang, "A secure 3D-SPIHT codec," in *Proceedings European Signal Processing Conference*, Sept. 2004, pp. 813–816.

[8] R. Norcen and A. Uhl, "Encryption of wavelet-coded imagery using random permutations," in *Proceedings IEEE International Conference on Image Processing*, Oct. 2004, pp. 3431–3434.