Hindawi Publishing Corporation
EURASIP Journal on Information Security
Volume 2007, Article ID 98374, 9 pages
doi:10.1155/2007/98374

Research Article

Joint Encryption and Compression of Correlated Sources with Side Information

M. A. Haleem, K. P. Subbalakshmi, and R. Chandramouli

Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA

Correspondence should be addressed to M. A. Haleem, mhaleem@stevens.edu

Received 6 March 2007; Revised 8 July 2007; Accepted 7 November 2007

Recommended by E. Magli

We propose a joint encryption and compression (JEC) scheme with emphasis on application to video data. The proposed JEC scheme uses the philosophy of distributed source coding with side information to reduce the complexity of the compression process and at the same time uses cryptographic principles to ensure that security is built into the scheme. The joint distributed compression and encryption is achieved using a special class of codes called high-diffusion (HD) codes that were proposed recently in the context of joint error correction and encryption. By using the duality between channel codes and Slepian-Wolf coding, we construct a joint compression and encryption scheme that uses these codes in the diffusion layer. We adapt this cipher to MJPEG2000 with the inclusion of minimal amount of joint processing of video frames at the encoder.

1. INTRODUCTION

With several multimedia applications being launched over the Internet, compression and encryption of this type of data have gained a lot of attention. The issue of complexity in compression is taken into consideration in the video coding standards such as MJPEG2000 [1] where only the *intraframe* coding is performed to keep the computational complexity low. Nevertheless, video sequences are rich in *interframe* correlation and an efficient compression scheme should make use of this property. Traditionally, the approach has been to compress the data first and then encrypt in a concatenated manner. It is potentially possible to reduce the complexity of the compression and encryption if a joint paradigm for both functions could be designed. In this paper, we present a joint approach to encryption and compression of digitized data and formulate a secure MJPEG2000 framework that we call SMJPEG2000. Attempts to combine the computational steps in compression and encryption include multiple Huffman tables (MHT) based approach [2], Arithmetic Coding with Key-based interval Splitting (KSAC) [3], and randomized arithmetic coding (RAC) [4]. In MHT, different tables are used for compression. The tables and the order in which they are used to encode the symbols are kept secret. KSAC is designed to achieve both compression and confidentiality by using keys to specify how the intervals will be partitioned in each iteration of the arithmetic encoding. RAC differs from KSAC only in that the keys are used to specify the order of the intervals instead of the positions where they will be split. MHT and KSAC have been shown to be vulnerable to low complexity known and/or to chosen plaintext attacks [5]. Our work differs from the above in that we develop a framework for joint encryption and compression of correlated sources like a video sequence. The compression component of our algorithm works on the concept of matrix-based coding that has emerged in the distributed source coding community.

Distributed source coding has emerged as an alternative to achieve low-complexity compression for correlated sources. Based on the theoretical results by Slepian and Wolf on lossless coding, and the extension of it to lossy coding with quantization by Wyner and Ziv in the 1970s, the development of practical coding schemes has commenced recently. Pradhan and Ramchandran [6] presented a constructive practical framework based on algebraic trellis codes dubbed distributed source coding using syndromes (DISCUS), that is applicable in a variety of settings. Girod et al. presented a scheme based on Wyner-Ziv coding where intraframe encoder is combined with interframe decoding to achieve excellent compression ratios with low-encoding

complexity [7]. This framework also has been used to analyze concatenated compression and encryption schemes. Johnson et al. proved that reversing the order of compression and encryption to compress the encrypted data can still achieve significant compression [8]. In some cases, the proof is based on the framework of distributed source coding with side information, and the encryption key plays the role of side information.

Our work presented in this paper is about achieving both security and compression with the same set of computational operations. In our proposed joint encryption and compression (JEC) scheme, we use a class of codes called the *high-diffusion codes* (HD codes) [9–13] that were proposed in the context of joint encryption and error correction. In the current work, the JEC scheme has a structure similar to the advanced encryption standard (AES) [14, 15] in that it is a key alternating block cipher. The diffusion box of our proposed cipher performs the dual function of compression as well as diffusion. Diffusion is a necessary element in block ciphers like the AES, to spread the statistical characteristics of the cipher state as quickly as possible and is measured in terms of the branch number. We establish the necessary and sufficient condition for achieving a compression function satisfying the branch number property and show that distributed compression using the HD codes can satisfy this condition.

In Section 2, we discuss the concepts behind the proposed approach and present the framework showing the feasibility of joint-distributed encryption and compression. The proposed scheme is elaborated in Section 3. The application of this approach to achieve security and compression in SMJPEG2000 is described in Section 4. The implementation and simulation results are presented in Section 5. Conclusions follow in Section 6.

2. FEASIBILITY OF JOINT-DISTRIBUTED ENCRYPTION AND COMPRESSION

In the distributed source coding framework of SMJPEG2000, there are two underlying sources X and Y generating correlated information in the form of sequences of symbols in a Galois field of order 2^8 (GF(256)). The correlation is such that any block of n consecutive symbols generated by X differs at most by $t(<n)$ symbols from n consecutive symbols simultaneously generated by Y. As per the Slepian-Wolf theorem [16], X can be compressed to achieve a bit rate approaching the conditional entropy $H(X \mid Y)$ and with the knowledge of Y, the decoder is able to recover X perfectly. The source X does not need to know Y to achieve this.

In order to guarantee confidentiality, we would also like to encrypt X to produce a cipher text, E_X, such that an adversary that knows nothing about the key cannot infer anything about X by observing E_X alone. In other words, we require the conditional probability distribution $P(X \mid E_X)$ to be equal to the probability distribution $P(X)$ [17]. Except with keys based on a one-time pad [18], perfect secrecy is known to be infeasible. Nevertheless, ciphers are considered to be computationally secure if (a) the time required to break the cipher is more than the useful time of the data being encrypted and (b) the cost of computation to break the cipher is more than the value of the information [19]. In AES, this is achieved via the round functions where each round consists of a sequence of cryptographic primitives, namely, key addition, substitution, row shifting, and column mixing.

In this work, we provide a framework where the diffusion layer of the cipher has dual functionality: (a) compressing the correlated source and (b) providing the requisite diffusion for the cipher. Since the success of the compression depends on exploiting the correlation between the sources, it is imperative to make sure that the diffusion operation in our joint compression/encryption scheme does not destroy the correlation. To do this, we show that *the key addition does not change the bitwise Hamming distance between X and Y and substitution does not change the bytewise Hamming distance and preserves the correlation*.

2.1. Hamming distance under key XOR operation

The following lemma establishes that bitwise Hamming distance remains unchanged under key-addition operation.

Lemma 1. *Let x and y be two n-tuples in $\mathbb{F}_2^n$ (binary) and let K be a third such n-tuple representing the secret key. Then*

$$d_H(x \oplus K, y \oplus K) = d_H(x, y), \tag{1}$$

where $d_H(\cdot, \cdot)$ is the bitwise Hamming distance.

Proof. The Hamming distance between x and y can be found by the XOR operation followed by computation of the weight, that is, $d_H(x, y) = w(x \oplus y)$. For example, if $x = 01001$ and $y = 11010$, then $x \oplus y = 10011$ and $w(x \oplus y) = 3$ which is the Hamming distance between x and y. Therefore we can also write

$$d_H(x \oplus K, y \oplus K) = w\big((x \oplus K) \oplus (y \oplus K)\big). \tag{2}$$

The XOR operation $\oplus$ is associative. Therefore we can rewrite (2) as

$$\begin{aligned} d_H(x \oplus K, y \oplus K) &= w\big((x \oplus y) \oplus (K \oplus K)\big) \\ &= w(x \oplus y) \oplus \underline{0} \\ &= w(x \oplus y) \\ &= d_H(x, y), \end{aligned} \tag{3}$$

thus we prove (1). In the above, $\underline{0}$ represents an all-zero n-tuple.

It can be easily verified that this lemma is also valid when x, y, and k are n-tuples with elements from Galois field, GF(2^m) for any positive integer m. □

2.2. Correlation under substitution operation

An S-box in AES performs substitution of a symbol with another such that each byte of the plain text is uniquely mapped to another byte in a one-on-one manner. Thus, if ith bytes of two different blocks of plain text are equal prior to substitution, then they are equal following the substitution process as well. On the other hand, if ith bytes of the two blocks of

plain text are different, then they will remain different following the substitution. Therefore, we can conclude that the *bytewise* Hamming distance between two multibyte blocks of data does not change under the substitution operation. However, at bit level, the Hamming distance may change due to the substitution depending on the S-box. Therefore, the substitution operation can be considered to be nonlinear operation at the bit level, and linear at the byte level. We show in the sequel that the conditional entropy $H(X \mid Y)$ is preserved under linear or nonlinear mapping as long as the mapping is one on one.

Lemma 2. *Let the random variables X and Y assume values in the discrete sets $\{x_i \mid i = 1, \ldots, n\}$ and $\{y_i \mid i = 1, \ldots, n\}$, respectively. If the joint probability of the random variables X and Y is symmetric such that $p(X = x_i, Y = y_j) = p(X = x_j, Y = y_i)$ or simply $p(x_i, y_j) = p(x_j, y_i)$ for all $i, j = 1, \ldots, n$, then $H(X \mid Y) = H(Y \mid X)$.*

Proof. $p(x_i, y_j) = p(x_j, y_i)$ implies the equality of marginal probabilities, that is, $p(x_i) = p(y_i)$ leading to $p(y_j \mid x_i) = p(x_j \mid y_i)$. By definition,

$$\begin{aligned} H(X \mid Y) &= \sum_{i=1}^{n} p(Y = y_i) H(X \mid Y = y_i) \\ &= -\sum_{i=1}^{n} p(y_i) \sum_{j=1}^{n} p(x_j \mid y_i) \log_2 p(x_j \mid y_i) \\ &= -\sum_{i=1}^{n} \sum_{j=1}^{n} p(x_j, y_i) \log_2 p(x_j \mid y_i) \\ &= -\sum_{i=1}^{n} \sum_{j=1}^{n} p(y_j, x_i) \log_2 p(y_j \mid x_i) \\ &= H(Y \mid X). \end{aligned} \tag{4}$$

□

Lemma 3. *If the mapping $X \to U = g(X)$ is one on one, then*

$$H(Y \mid g(X)) = H(Y \mid X). \tag{5}$$

Proof. With one-on-one mapping we have $p(X = x) = p(u = g(X = x))$ and similar result holds for joint probabilities. The result is self-explanatory from the definition of conditional entropy. □

Theorem 1. *If (a), the joint probability matrix of X and Y, is symmetric (b) the mapping $X \to U = g(X)$ is one on one, then*

$$H(g(X) \mid Y) = H(X \mid Y). \tag{6}$$

Proof. From Lemma 2, we have

$$H(g(X) \mid Y) = H(Y \mid g(X)). \tag{7}$$

From Lemma 3, we have

$$H(g(X) \mid Y) = H(Y \mid X). \tag{8}$$

Again from Lemma 2, we have

$$H(g(X) \mid Y) = H(X \mid Y). \tag{9}$$

□

3. JOINT-DISTRIBUTED ENCRYPTION AND COMPRESSION FRAMEWORK

One of the practical methods of constructing Slepian-Wolf codes is to use binning based on *good* linear channel codes. Let x be an n-tuple generated by the source X; and let y be the n-tuple simultaneously generated by the correlated source Y. Both x and y can be considered as noise-corrupted versions of valid codewords generated by an (n, k) linear block code, $\mathcal{C}$. Further, x can be modeled as a noise-corrupted version of y if the correlation between X and Y can be modeled as additive noise. If $d_{\min}$ is the minimum distance of $\mathcal{C}$, then for any n-tuple x, there exists a valid codeword c_x within a Hamming distance $t = \lfloor d_{\min}/2 \rfloor$, the maximum number of correctable errors of the linear-block code. Similar result holds for y. Further, if the Hamming distance between x and y is $\leq t$, we have

$$\begin{aligned} x &= c_x + e_x, \\ y &= c_y + e_y, \\ y &= x + e_c = c_x + e_x + e_c, \end{aligned} \tag{10}$$

where c_x, c_y are the valid codewords within a Hamming distance $\leq t$; e_x and e_y are the error patterns corresponding to x and y, respectively, and e_c is the error pattern representing the correlation between x and y.

Now let H be the $(n - k) \times n$ parity check matrix. Then the projections of n-tuples x and y onto the dual space result in the syndromes $S_x = xH^T$ and $S_y = yH^T$, that is,

$$\begin{aligned} xH^T &= c_x H^T + e_x H^T = 0 + S_x, \\ yH^T &= c_y H^T + e_y H^T = 0 + S_y, \end{aligned} \tag{11}$$

where H^T is the transpose of H. Further we may write

$$S_y = yH^T = xH^T + e_c H^T = S_x + S_c, \tag{12}$$

that is,

$$S_c = S_x + S_y. \tag{13}$$

Note that the syndromes are $(n - k)$ tuples. This result leads to the method of compression and lossless decoding of X with the knowledge of side-information Y and the correlation between X and Y. The transmitter can compute S_x and send to the receiver where Y is available. Then the syndrome S_c can be computed using the received syndrome S_x and y. The error pattern e_c corresponding to S_c can be computed using a syndrome decoding technique. Since the HD code used in the proposed cipher is a general case of RS codes [13], the Berlekamp-Massey algorithm [20] that is generally used to decode RS codes, can be adapted in the decode/decrypt operation of this joint cipher. The n-tuple x can be found from

$$x = y + e_c. \tag{14}$$

Since the n-tuple x is transformed into the $n - k$ tuple S_x, we achieve a compression ratio of $n/(n - k)$. In the design of JEC, the transform used for compression, namely, the parity check matrix of the underlying linear block code, should

achieve the required spreading, or the *diffusion* achieved by the column mixing operations in the AES cipher. Diffusion is required to achieve robustness against both differential cryptanalysis and linear cryptanalysis. It has been shown [15] that the diffusion caused by a transform can be effectively measured using the branch number. Definitions 1 and 2 and Lemma 4 provide a concise description of branch number.

Definition 1. The differential branch number of a transform, ϕ, mapping an n-tuple to an l-tuple is defined as

$$\mathcal{B}_d^{\text{diff}} = \min_{d_H(x_1,x_2)\neq 0} \{d_H(x_1,x_2) + d_H(\phi(x_1),\phi(x_2))\}, \quad (15)$$

where x_1 and x_2 are two input n-tuples ($x_1 \neq x_2$) and d_H is the Hamming distance in a number of symbols [15].

Definition 2. The linear branch number of a transform, ϕ, mapping an n-tuple x to an l-tuple is defined as

$$\mathcal{B}_d^{\text{lin}} = \min_{x\neq 0} \{w(x) + w(\phi(x))\}, \quad (16)$$

where $w(\cdot)$ is the Hamming weight.

Lemma 4. *The upper bound of branch number is $l + 1$.*

Proof. With a diffusion-optimized transform, ϕ, a change in a single symbol x_1 should result in changes in all the output symbols leading to $d_H(x_1,x_2) + d_H(\phi(x_1),\phi(x_2)) = l + 1$, which is the minimum (maximum of this sum being $n + l$) and therefore is the branch number by Definitions 1 and 2. □

The design of the *diffusion layer* in Rijndael cipher adopted in AES ensures this upper bound for all possible values of linear/differential weights of the input [21]. We show in Theorem 2 that the necessary and sufficient condition to achieve such linear and differential branch number properties is that the transform ϕ is a *totally positive matrix*. The formal definition of a totally positive matrix is as follows.

Definition 3. A rectangular matrix $\mathcal{A} = (a_{ij})$, $i = 1,\ldots,n$; $j = 1,\ldots,l$ is called *totally positive* if all its minors (determinants of submatrices) of any order are positive [22].

Although the original definition in [22] is for matrices of real values, it can be easily extended to the case with elements in Galois field GF(2^m).

Theorem 2. *Over a field $\mathcal{F}$, the linear transformation of n-tuples in an n-dimensional space, V^n, into l-tuples in an $l(\leq n)$-dimensional space, V^l by an operation $y = x\mathcal{A}$, achieves the branch number properties if (sufficient) and only if (necessary) $\mathcal{A}$ is a totally positive matrix.*

Proof. First we prove that total positivity is a necessary condition to achieve the branch number properties. From Definitions 1, 2, and Lemma 4, for transformation $\mathcal{A}$ to be optimal in terms of diffusion, we require that

$$\begin{aligned} &d(x_1,x_2) + d(x_1\mathcal{A}, x_2\mathcal{A}) \geq l+1 \\ &\quad\Longrightarrow w(x_1 \oplus x_2) + w(x_1\mathcal{A} \oplus x_2\mathcal{A}) \geq l+1. \end{aligned} \quad (17)$$

Table 1: Minimum change in the output to maintain branch number.

$w(e)$	$\min\{w(e\mathcal{A})\}$
0	0
1	l
2	$l-1$
⋮	⋮
r	$l-(r-1)$
⋮	⋮
l	1
$\geq l+1$	0

Since $\mathcal{A}$ is a linear transformation, (17) implies

$$w(x_1 \oplus x_2) + w((x_1 \oplus x_2)\mathcal{A}) \geq l+1. \quad (18)$$

Let $x_1 \oplus x_2 = e$. Then (18) reduces to

$$w(e) + w(e\mathcal{A}) \geq l+1. \quad (19)$$

The minimum values of $w(e\mathcal{A})$ corresponding to the values of $w(e)$ required to satisfy (19) are as given in Table 1.

It can be seen that for $w(e) = r$, $\min\{w(e\mathcal{A})\} = l - (r-1)$. Let the columns of $\mathcal{A}$ be denoted by h_j, $j = 1,\ldots,l$. Then with a given $r \in \{1,2,\ldots,l\}$, we require $\mathcal{A}$ to have at most $r-1$ columns such that $e\cdot h_j = 0$. This implies that in the $r \times l$ submatrix formed by selecting the rows of $\mathcal{A}$ corresponding to the nonzero elements of e, every $r \times r$ submatrix (contiguous as well as noncontiguous) should be of full rank. Since the r nonzero elements in e can occur at any r out of n-positions, the above implies that every $r \times r$ submatrix of $\mathcal{A}$ should be of full rank, that is, positive for $r = 1,\ldots,l$. Thus by Definition 3, $\mathcal{A}$ should be a totally positive matrix.

Next we prove that the total positivity of the transformation matrix is sufficient to achieve the maximum branch number. If $\mathcal{A}$ is a totally positive matrix, every $r \times r$ submatrix is positive, that is, has full rank for $r = 1,\ldots,l$. Let the rows of $\mathcal{A}$ be a_i, $i = 1,\ldots,n$. Then the linear combination of any r rows, $\sum_{i=1}^{r}\alpha_i a_i$ with $\alpha_i > 0$ results in an l-tuple with at most $r-1$ zero elements leading to $w(e) + w(e\mathcal{A}) = l+1$ and hence achieves the branch number. While this proof explicitly addresses the case of differential branch number, the case of linear branch number is implicit. □

From Theorem 2, we achieve a test for branch-number property for any given transform. Further, it serves as a guideline for designing transforms to achieve the desired branch-number properties. While the testing of all possible square submatrices of a matrix for positivity has an exponential-order complexity, [23, Theorem 9] provides a method of polynomial-order complexity. This theorem states that a square matrix is totally positive if and only if all its initial minors are positive. The initial minors are minors that are contiguous and include the first row or the first column. This approach reduces the number of minors required to be tested for an $n \times n$ matrix from $\binom{2n}{n} - 1$ to n^2.

One known example of totally positive matrix is the *generalized Vandermonde* matrix [22] given by

$$\begin{pmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{(p-1)} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{(p-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_q & a_n^2 & \cdots & a_n^{(p-1)} \end{pmatrix}, \tag{20}$$

where $0 < a_1 < a_2 < \cdots < a_n$.

Recently a class of codes called high-diffusion codes (HD-codes) were developed [9, 12] which incorporated the branch-number criterion as well as the being maximum distance separable. Two constructions for error-correcting ciphers were then proposed using these codes [10, 11, 13]. In this paper, we will use the duality between error-correcting codes and Slepian-Wolf coding to construct a joint-compression encryption system using these HD codes.

4. SECURE MJPEG2000 (SMJPEG2000)

The distributed source coding framework for correlated sources can be used in secure compression of video sequences. Figure 1 shows the image coding framework as per JPEG2000. In the motion JPEG2000 (MJPEG2000), each frame is simply encoded independent of the rest of the frames. In JPEG2000, the 2D wavelet transform provides the different subbands as in Figure 2. The subbands of a frame from "foreman" sequence are shown as an example. The wavelet coefficients are then quantized and converted to integers. Treating these integer values as symbols, entropy coding is achieved by the use of run-length coding followed by Huffman coding [24]. The one-dimensional sequence, $\{x_n\}$, of symbols from the alphabet $\mathcal{A}_X$ is run-length coded by replacing $\{x_n\}$ with a sequence of symbol pairs, $\{(a_k, r_k)\}$, representing symbol values, $a_k \in \mathcal{A}_X$, and run-lengths, $r_k \in \mathbb{Z}_+$, where $\mathbb{Z}_+$ represents the set of nonnegative integers. The mapping between $\{(a_k, r_k)\}$ and $\{x_n\}$ is such that $x_n = a_k$ for all n such that

$$\sum_{j=1}^{k-1} r_j < n \leq \sum_{j=1}^{k} r_j, \tag{21}$$

where $k = \{1, 2, \dots\}$ and $n = \{1, 2, \dots\}$. The value r_k is normally the longest run of symbols, $x_n, n > \sum_{j=1}^{k-1} r_j$, such that x_n has a constant value, a_n. The sequence of run-length symbol pairs $\{(a_k, r_k)\}$ is coded with Huffman code in our experiments, although arithmetic coding may also be used. Separate codes are constructed for the symbol values a_k and the run-lengths r_k. Through experiments, we find that the benefit of run-length coding in terms of the compression is significant only for the zero values of the quantized wavelet coefficients. Thus the run-length coding in our work is confined to coding of zero runs. Further, since the representation of each run length requires two symbols, coding of only the runs of three or more zeros results in compression.

Figure 3 shows our proposed framework where some of the interframe dependence is captured via the proposed joint-distributed compression and encryption scheme. Following the quantization as in JPEG2000, the block of symbols (integers) are run length coded. Next, each wavelet coefficient is represented using the minimum required bits. Instead of the Huffman coding stage, the JEC is used. At the decoder, joint decryption and decompression is performed using information from the previously decoded frame as the side information.

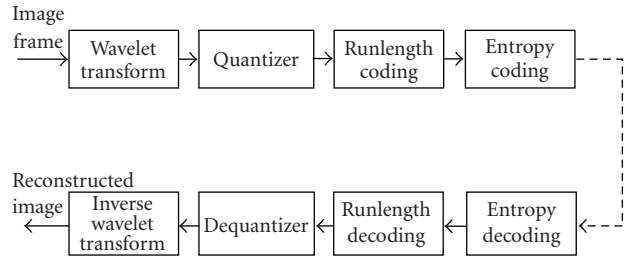

Figure 1: Functional diagram of JPEG2000.

Cardinality of the set of symbols (integers), needed to represent the quantized wavelet transforms, varies over each subband. LL has the largest set whereas HH has the smallest. Therefore, separate allocation of bits for each subband is required. Once the symbols are represented by bits, they are parsed to form a single block of bits for the entire frame. Note that the application of run-length coding to each frame independently would result in the loss of synchronization between the blocks of data corresponding to adjacent frames. This will make it difficult to apply the JEC scheme. In order to overcome this issue, we propose to process a set of frames jointly during run length coding. Thus only the symbol runs that are common to all the frames in the set are run-length coded. The first frame of each such set serves as the key frame and is compressed independently of the remaining frames just as in the current JPEG2000. However, for the run-length computations as mentioned above, we include the key frame as well. The key frame is independently compressed and then encrypted using AES in a concatenated manner. The key frame provides the run-length coding parameters to the decoder. The JEC scheme is applied to the successive frames. Key-frame refresh rate is selected so as to control the degradation in quality due to error propagation in the sequence of frames during decoding.

For a frame other than the key frame, run-length coding is followed by the representation of blocks of wavelet coefficients in each subband by the minimum number of bits required, $\lceil \log_2 |S_i| \rceil$, where S_i is the set of different values in subband i. Thus the total bit requirement is $\sum_{i=1}^{N} \lceil \log_2 |S_i| \rceil$. The resulting bit stream is segmented into bytes in order to directly apply GF(2^8) arithmetic during the joint encryption and compression process. Since this approach maintains synchronization among the data corresponding to all the frames that are jointly processed during run-length coding, it allows us to successfully apply JEC as described in Section 3. JEC allows compression by a factor given by $n/(n-k)$ with an $(n, k, 256)$ HD code since a block of n-bytes is transformed into a block of $n-k$ bytes at the joint compression/diffusion

(a) Schematic (b) Foreman

Figure 2: Passband structure for a 2D subband transform with $D = 3$.

stage of the JEC. As long as the difference between two adjacent frames is such that for each block of n-bytes, the difference is only $t \leq (n-k)/2$ bytes, the frames can be perfectly decoded. However, the differences in the wavelet coefficients of adjacent frames are distributed rather non-uniformly in general, and therefore limited difference per block of n-bytes as mentioned above is not guaranteed. We achieve the best result by systematically swapping the bytes prior to JEC to achieve $t \leq (n-k)/2$ bytes of difference per block of n-bytes wherever possible. In the process, a swap table is built and included in the header. This process significantly enhances the overall decoding capability with a given t. Nevertheless, if the difference between the adjacent frames is excessive, not all blocks can be decoded successfully, that is, there is a limit to the overall correctable errors. However, this is true of any Slepian-Wolf coding scheme based on error-correcting codes.

A nonkey frame is jointly decrypted and decompressed with the use of previously decoded frame. The intermediate results following the joint decryption and decompression of such a frame are stored to be used as side information for the decoding of the next nonkey frame. Following the joint decryption and decompression phase, the bits are regrouped to represent the encoded wavelet coefficients. Run-length decoding and inverse wavelet transform follow.

5. IMPLEMENTATION AND SIMULATION RESULTS

In the proposed JEC scheme, the compression is included in the first layer of tenth round of the joint compression-encryption scheme as shown in Figure 4. The row shifting and column mixing operations in the first round is replaced by the syndrome encoding of HD codes. Similarly, during the decryption, the inverse-column mix and inverse-row shift operations of the last round are replaced by joint decryption and decompression process. In the implementation of our JEC scheme, we used $(7, 3, 256)$-HD code, that is, $n = 7$, $k = 3$ with the following parity check matrix of elements in GF(2^8):

$$H = \begin{pmatrix} 1 & 2 & 4 & 8 & 16 & 32 & 64 \\ 1 & 4 & 16 & 64 & 29 & 116 & 205 \\ 1 & 8 & 64 & 58 & 205 & 38 & 45 \\ 1 & 16 & 29 & 205 & 76 & 180 & 143 \end{pmatrix}. \quad (22)$$

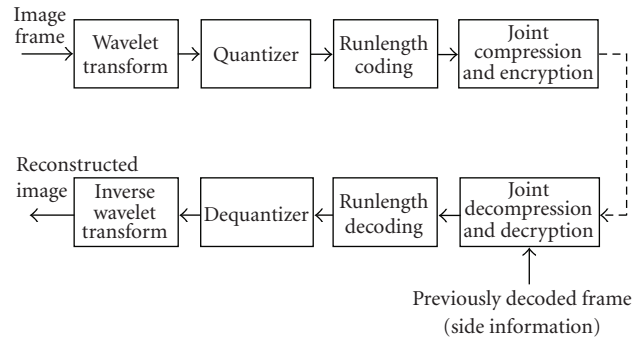

Figure 3: Functional diagram of proposed MJPEG2000.

5.1. Compression and savings in computation

This implementation achieves a lossless compression ratio of $n/(n-k) = 7/4$. Although other implementations with varying degrees of compression are possible using other HD codes, we leave the design of a family of joint compression-encryption ciphers for future work.

In the AES cipher, 128 bit blocks of data are arranged in a 4×4 matrix [15]. This matrix of data undergoes initial key addition and substitution. Each of the round functions that follow consists of a diffusion layer implemented by the row shifting and column mixing operation followed by the addition of a round key and substitution. In the proposed JEC scheme, we start with a matrix of 7×4 bytes of data. Each column of 7 bytes is compressed using syndrome forming transform obtained from the $(7, 3, 256)$ HD-code. This leads to a 4×4 data matrix. The key addition and substitution function of the first round and the functionalities of remaining rounds follow the AES cipher.

The savings in computational steps of the JEC compared to a concatenated system in a layer (compression followed by encryption) are as follows. For the basic operations on a byte, namely, addition, substitution, and multiplication, we assume one unit of complexity. The actual complexity of these different operations may vary, and are highly dependent on the particular architecture. Nevertheless with reasonably optimized architecture, energy consumptions for these operations will be comparable and may not be drastically different. In the JEC, we start with a matrix of 7×4 bytes of row data.

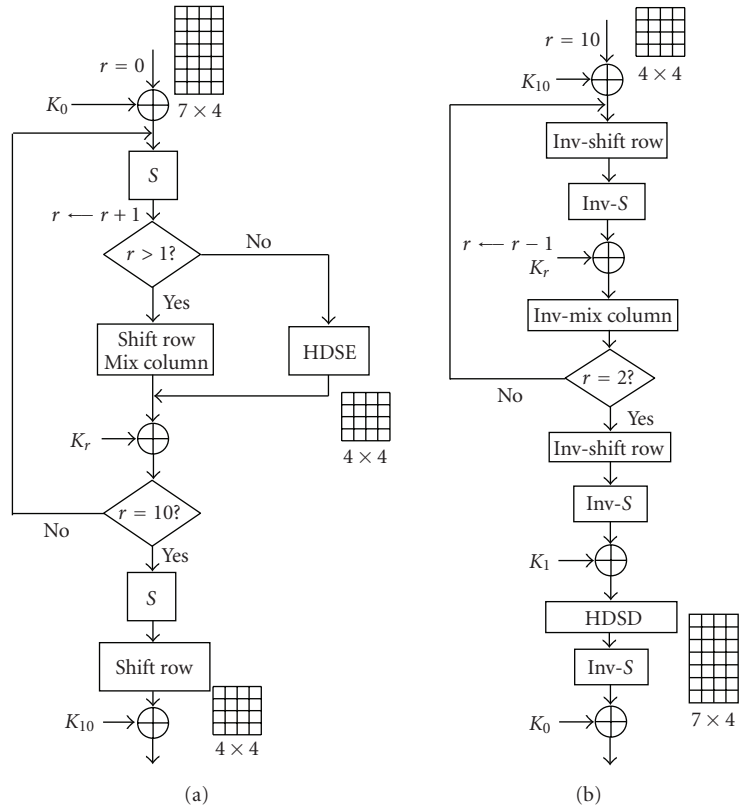

Figure 4: Flow chart of the proposed secure joint-distributed encryption and compression: (a) compression/encryption (b) decompression/decryption. HDSE stands for high-diffusion syndrome decoding, and represents multiplication with the HD parity check matrix; and HDSD (high-diffusion syndrome decoding) represents the syndrome decoding process.

Thus the initial key addition requires $7 \times 4 = 28$ additions. Equal number of substitutions follows. In the compression phase, there are 28 multiplications and equal number of additions. In total, there are $28 \times 4 = 112$ operations.

Compared to that, in a concatenated approach (compression followed by encryption), the compression requires 28 multiplications and that many additions. The joint compression-diffusion operation of the first round has an output of $4 \times 4 = 16$ bytes. In the encryption stage, there are 16 key addition operations and 16 substitutions. The row shifting operation requires 16 multiplications and many additions. The mix-column operation also requires equal amount of computations. Thus there are $2 \times 28 + 4 \times 16 = 120$ units of operations in total. Similarly, at the decoder, the JEC requires 28 substitutions and 28 additions during key addition in addition to the decompression procedure leading to $2 \times 28 = 56$ units of computations. In contrast, the concatenated system requires $8 \times 16 = 128$ units of computation in the inverse column mixing, row shifting, substitution, and key addition operations prior to decompression. Thus we have a saving of $(120 + 128) - (112 + 56) = 80$ units. The total number of computations in the compression and first round of AES cipher in the concatenated system being $2 \times 28 + 8 \times 16 = 184$ units, we have a saving of 43.5% in this round.

Considering all 10 rounds of AES cipher, we have $2 \times 28 + 10 \times 8 \times 16 + 4 \times 16 = 1400$ units of computation thus resulting in a saving of 5.7%. Note that if a technique to progressively compress at more than one round is achievable, larger saving will result. The computational results from the implementation show that in all the cases with Hamming distances $\leq t$ between the correlated vectors x and y, x is perfectly decoded with the knowledge of y in compliance with the theoretical conclusions.

5.2. SMJPEG2000 video coding

We incorporated the implementation of JEC as parameterized above into MJPEG2000 video coding to produce the S-MJPEG 2000 joint compression encryption scheme. Three-layer coding was used ($D = 3$). With the "container" sequence as the test sample, we obtained savings in bit rate while maintaining the same quantization step sizes for both cases. With the quantization step sizes fixed, we achieve the same peak signal-to-noise ratio (PSNR) performance with standard MJPEG2000 and the proposed SMJPEG2000. Comparison of rate allocations with the standard JPEG2000 and the proposed scheme is shown in Table 2 with varying quantization step sizes. We observe savings up to 9.7% with this sequence. Figure 5 shows the comparison of PSNR for step

Table 2: Comparison of average bit rates achieved for the MJPEG 2000 and the proposed S-MJPEG 2000 for the subset of five frames of the "Container" sequence. The first column shows the step sizes used for the different wavelet bands.

Step Sizes ($HL_1, LH_1, HH_1, HL_2, LH_2, HH_2, HL_3, LH_3, HH_3, LL_3$)	Bits per pixel		Saving (%)
	MJPEG2000	JEC	
32.5, 32.50, 65.00, 16.25, 16.25, 32.50, 8.13, 8.13, 16.25, 4.06	1.7544	1.7058	2.77
16.25, 16.25, 32.50, 8.13, 8.13, 16.25, 4.06, 4.06, 8.13, 2.03	1.1018	1.0374	5.84
8.13, 8.13, 16.25, 4.06, 4.06, 8.13, 2.03, 2.03, 4.06, 1.02	0.6455	0.5830	9.68

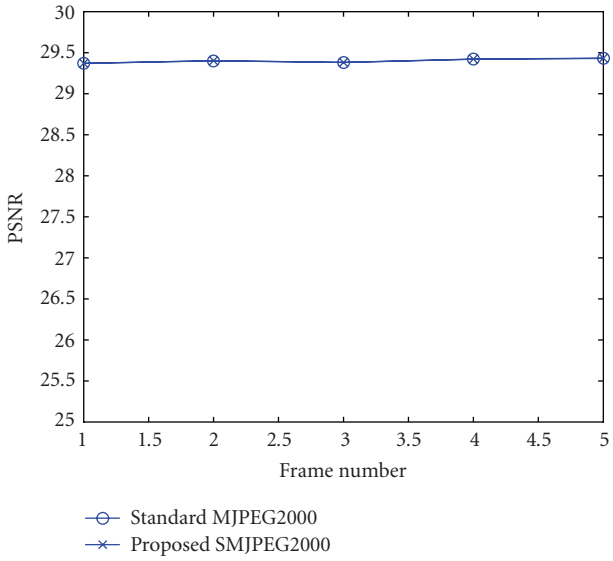

Figure 5: Comparison of peak signal-to-noise ratio for various frames of the "Container" sequence at bit rates of 0.6455 bits/pixel for the MJPEG 2000 and 0.5830 bits/pixel for the S-MJPEG 2000 algorithm.

sizes as in the third row of Table 2. The size of the swap table in this case has been 2.4% of the total amount of data from the encoded frame. For sequences with more motion, this amount is observed to increase. For example, for foreman sequence and bus sequence, we observe, respectively, 7.6% and 18% overheard. This framework also achieved security with savings in computational requirements as discussed in the previous sections.

6. CONCLUSION

We presented a joint encryption and compression paradigm for correlated sources. The theoretical framework establishing the feasibility of such a paradigm has been discussed. It is shown that under key addition and substitution primitives of encryption process, the correlation between blocks of data is preserved leading to the possibility of joint distributed compression and encryption. We also presented theorems establishing the necessary and sufficient conditions for a transform to achieve maximum branch number so required in the diffusion layer of state-of-the-art data encryption schemes. We discussed the construction of one such joint encryption compression scheme based on the recently proposed high-diffusion (HD) codes. We also presented a secure MJPEG2000 (SMJPEG2000) framework where the joint encryption and compression scheme is successfully applied to achieve improved compression by exploiting interframe correlation while at the same time ensuring that the content is encrypted. Since the proposed scheme is a joint encryption compression scheme, it has a computational advantage over the traditional concatenated schemes.

ACKNOWLEDGMENTS

The work presented in this paper was funded in part by the NSF-CT Grant no. 0627688 and the US Army Picatinny Arsenal/iNeTS.

REFERENCES

[1] ISO/IEC 15444-3:2002, "Information technology—JPEG2000 image coding system—part 3: motion jpeg2000," 2002.

[2] C.-P. Wu and C.-C. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 828–839, 2005.

[3] J. G. Wen, H. Kim, and J. D. Villasenor, "Binary arithmetic coding with key-based interval splitting," *IEEE Signal Processing Letters*, vol. 13, no. 2, pp. 69–72, 2006.

[4] M. Grangetto, E. Magli, and G. Olmo, "Multimedia selective encryption by means of randomized arithmetic coding," *IEEE Transactions on Multimedia*, vol. 8, no. 5, pp. 905–917, 2006.

[5] G. Jakimoski and K. P. Subbalakshmi, "Cryptanalysis of some multimedia encryption schemes," to appear in *IEEE Transactions on Multimedia*.

[6] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction, (DCC '99)," in *Proceedings of the Conference on Data Compression*, p. 158, Washington, DC, USA, 1999.

[7] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[8] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 2992–3006, 2004.

[9] C. N. Mathur, K. Narayan, and K. P. Subbalakshmi, "High diffusion codes: a class of maximum distance separable codes for error resilient block ciphers," in *Proceedings of the IEEE GLOBECOM Workshop: 2nd IEEE International Workshop on Adaptive Wireless Networks (AWiN '05)*, St. Louis, Mo, USA, November 2005.

[10] C. N. Mathur, K. Narayan, and K. P. Subbalakshmi, "On the design of error-correcting ciphers," *Eurasip Journal on Wireless Communications and Networking*, vol. 2006, Article ID 42871, 12 pages, 2006.

[11] C. N. Mathur, K. Narayan, and K. P. Subbalakshmi, "High diffusion cipher: encryption and error correction in a single cryptographic primitive," in *Proocedings of the 4th International Conference on Applied Cryptography and Network Security (American Conference on Neutron Scattering)*, vol. 3989, pp. 309–324, Singapore, June 2006.
[12] K. Narayan, "On the design of secure error resilient diffusion layers for block ciphers," M.S. thesis, Steven Institute of Technology, Hoboken, NJ, USA, May 2005.
[13] C. N. Mathur, *A mathematical framework for combining error correction and encryption*, Ph.D. thesis, Department of Electrical and Computer Engineering, Stevens Institute of Technology, Castle Point on Hudson, Hoboken, NJ, USA, 2007.
[14] "Specification for the advanced encryption standard (AES)," Federal Information Processing Standards (FIPS) Publication 197, 2001.
[15] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer, Secaucus, NJ, USA, 2002.
[16] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
[17] C. E. Shannon, "Communication Theory of Secrecy System," Now declassified confidential report, 1946.
[18] G. S. Vernam, "Secret signaling system," U.S. Patent 1310719, July 1919.
[19] D. R. Stinson, "Cryptography: Theory and Practices," in *Discrete Mathematics and Its Applications*, K. H. Rosen, Ed., CRC Press, 2000 Corporate Blvd., N.W., Boca Raton, Fla, USA, 1995.
[20] S. Lin and D. J. Costello, *Error Control Coding*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2004.
[21] J. Daemen and V. Rijmen, AES Proposal: Rijndael, http://csrc.nist.gov/archive/aes/index.html.
[22] F. R. Gantmacher, *The Theory of Matrices*, vol. 2, Chelsa, New York, NY, USA, 1964.
[23] S. Fomin and A. Zelevinsky, "Total positivity: tests and parameterizations," December 1999, http://arxiv.org/PS_cache/math/pdf/9912/9912128v1.pdf.
[24] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*, Kluwer Academic, Dordrecht, The Netherlands, 2002.