JOINT DISTRIBUTED COMPRESSION AND ENCRYPTION OF CORRELATED DATA IN SENSOR NETWORKS

M. A. Haleem, Chetan. N. Mathur, and K. P. Subbalakshmi
Department of Electrical and Computer Engineering,
Stevens Institute of Technology, Hoboken, NJ 07030.
Email:mhaleem@stevens.edu

Abstract— **In this paper, we propose, formulate, and study a joint distributed data compression and encryption scheme suitable for wireless sensor networks where we adopt the structured encryption system of Advanced Encryption Standard (AES) [9]. The distributed compression is achieved as per the Slepian-Wolf coding theorem, using channel codes. Core to achieving optimal compression in the joint compression and encryption is the preservation of correlation among different blocks of data despite applying cryptographic primitives. We establish that the correlation between sources remains unchanged when cryptographic primitives, namely key addition and substitution are applied. However, as a requirement of security in the encryption, any correlation between two inputs to a encryption system is removed with diffusion techniques. Compliance to the requirements of diffusion layer of AES cipher is achieved by designing the compression function so as to maintain *branch number property*. We establish the necessary and sufficient condition for achieving a compression function with branch number property and show that distributed compression using non-systematic Reed Solomon (RS) code can satisfy this condition.**

I. INTRODUCTION

Wireless sensor networks have been of increased interest in monitoring physical and ecological phenomena as well as in military and civil surveillance applications in recent time. Consequently many different issues related to sensor networks have received significant attention of researchers worldwide. Efficient and secure techniques for data gathering are among such issues of interest. Optimal method to process and forward the data collected by a sensor field to the destination (sink) is essential for the efficient use of wireless channel capacity, processing capabilities of the sensors, and limited power available at sensor nodes. In this regard, compression of data from a sensor with the knowledge on the correlation to data collected by a neighboring sensor destined to the same sink can be achieved by the technique of distributed source coding with side information as studied in [3][6]. Along with the necessity for optimization of performance metrics, security of the information against adversary attacks such as eavesdropping is of utmost importance. The state of the art in achieving the goals of best performance and security is to concatenate the tasks of performance related processing such as *data compression* and security related procedures such as *encryption*. There have been studies on the feasibility and advantages/disadvantages of encryption prior to compression of data as it is beneficial in scenarios where a node does not have the capability to compress but to encrypt [5]. In this paper we establish that a joint approach to data compression and encryption can achieve significant saving in computational complexity. This approach seeks to identify the common features of data compression algorithms and encryption primitives, and to invent techniques to achieve the two goals with the same computational procedures.

The information collected by the sensors in a sensor field is known to be rich in correlation structures. When the distance between two sensors shrinks, the difference between information generated by the two sensors also shrinks. The optimal process of gathering such correlated data should make use of such correlation to compress the data to the best possible extent as being routed to the destination. Slepian-Wolf coding theorem [2] establishes the possibility for such a compression technique without a sensor knowing the correlated information generated by the neighboring sensors. It suffices to know the correlation structure between the information produced by the two sensors. In principle, all what is required to know by a sensor (source) X to compress up to a rate exceeding the conditional entropy $H(X|Y)$ bits/sample so that to reproduce X with the neighboring sensor output Y as side information, is the correlation structure in terms of joint probability mass function $P_{XY}(x, y)$. The joint probability mass function should also be known at the decoder. In [3], the authors presented a constructional approach to achieve this compression rate using channel codes. This has been followed by significant studies on achieving any point on the Slepian-Wolf rate region as well as the extension of the technique to more than two correlated sources [6]. Linear block codes, convolutional codes, turbo codes, and low density parity check codes have been among the candidates being studied for the distributed compression of correlated data. In this paper, we show that wireless sensor networks can greatly benefit from a joint approach to distributed coding and encryption of correlated data with the use of channel codes for distributed compression.

As Advanced Encryption Standard (AES) [8] is of wide acceptance as the best known cipher system, we incorporate the distributed compression into AES by augmenting atomic encryption primitives therein so as to achieve both compression and security features. Ten round AES cipher has been part of mandated and optional components to achieve security in wireless local area networks (LAN) namely IEEE 802.11i standard. The security features, Wi-Fi Protected Access 2 (WPA 2) Of IEEE 802.11i use AES cipher in its mandated CCMP mode and the optional offset code book (OCB) mode. In particular,

the OCB mode which is claimed to have superior performance uses AES cipher for encrypting data. OCB mode proposal to NIST for a block-cipher mode of operation simultaneously provides privacy and authenticity [16]. In this paper we present our findings on the feasibility of joint distributed compression and encryption of correlated data and discuss an approach using Reed-Solomon (RS) code for the distributed compression. In Section II to follow we discuss the concepts behind the proposed approach and establish the results showing the feasibility of joint distributed compression and encryption. The proposed High Diffusion Distributed Compression (HDDC) is elaborated in Section III. In Section IV, we present the implementation of joint compression and encryption using HDDC technique and discuss the performance of the approach and the savings in computational cost. Section V concludes the paper.

II. Feasibility of Joint Distributed Coding and Encryption

The problem addressed in this work is as follows. There are two sensors X and Y generating correlated information in the form of sequences of symbols in Galois field of 2^8. The correlation is such that any block of n consecutive symbols generated by X differ at most by $t(< n)$ symbols from n consecutive symbols simultaneously generated by Y. As per the Slepian-Wolf theorem [2], X can be compressed to achieve a minimum bit requirement approaching the conditional entropy $H(X|Y)$ and sent to Y so as to perfectly recover X with the knowledge of Y. The sensor X does not require to know Y to achieve this. In the joint distributed compression and encryption problem we consider, we also need to encrypt X and produce a cipher-text E_X such that an adversary without the knowledge of cipher key(s) cannot infer any statistics on X by observing E_X. In other words we require the conditional probability distribution $P(X|E)$ to be equal to the probability distribution $P(X)$ [10]. Except with one-time-pad based key addition [7], the perfect secrecy is known to be practically 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 [11]. In AES cipher, 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. We show here that the key addition and substitution operations do not alter the Hamming distance between the correlated sources X and Y. In particular key addition maintains the bit-wise Hamming distance and substitution maintains byte-wise Hamming distance which suffices to achieve our goal of preserving the symbol-wise Hamming distance. The purpose of row-shifting and column mixing operation being to de-correlate two plain-texts that are correlated, by the diffusion process, and since the compression of X using linear block code also achieves this goal, we can achieve both compression and diffusion by the same set of computational steps at once leading to significant savings in computation.

A. Hamming distance under key XOR operation

The following Lemma establishes that bit-wise 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 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 $d_H(.,.)$ is the bit-wise Hamming distance.

Proof: The Hamming distance between x and y can by found by the XOR operation followed by computation of the weight *i.e.*,$d_H(x, y) = w(x \oplus y)$. 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

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

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

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

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 where the elements are from Galois field, $GF(2^m)$ for any positive integer m.

■

B. Correlation under substitution operation

An S-box in AES system performs substitution of a symbol with another so that, when combined with XOR operations with secret keys (key addition) provides resiliency against differential cryptanalysis [4][11]. In the substitution layer of AES cipher, each byte of the plain-text is uniquely mapped to another byte in a one-to-one manner. Obviously, the byte-wise Hamming distance between two multi-byte blocks of data does not change under the substitution operation. Further, substitution operation can be considered a non-linear operation at bit level, and a linear operation if considered byte-wise. We show in the sequel that the conditional entropy $H(X|Y)$ is preserved under linear or non-linear mapping as long as the mapping is one-to-one.

Lemma 2: Let the random variables X and Y assume values in the discrete sets respectively $\{x_i | i = 1, \cdots, n\}$ and $\{y_i | i = 1, \cdots, n\}$. 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 $i, j = 1, \cdots, n$ then the conditional entropies hold the result $H(X|Y) = H(Y|X)$.

Proof: $p(x_i, y_j) = p(x_j, y_i)$ implies the equality of marginal probabilities i.e., $p(x_i) = p(y_i)$ leading to $p(y_j|x_i) = p(x_j|y_i)$. By definition,

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

■

Lemma 3: If the mapping $X \rightarrow U = g(X)$ is *one-to-one*, then

$$H(Y|g(X)) = H(Y|X) \tag{3}$$

Proof: With one-to-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 $X \rightarrow U = g(X)$ is one-to-one and then

$$H(g(X)|Y) = H(X|Y) \tag{4}$$

Proof: From Lemma 2 we have

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

From Lemma 3 we have

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

Again from Lemma 2 we have,

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

■

III. Proposed High Diffusion Distributed Compression (HDDC) for lossless decoding with side information

In this section, the details on the compression with transforms for diffusion and lossless decoding with side information are given. The use of linear block codes for lossless distributed compression is based on the results as follows. Let x be an n-tuple generated at the source X and y be the n-tuple simultaneously generated at the correlated source Y. Then x and y can be considered as noise corrupted versions of valid codewords generated with an (n, k) linear block code. If d_{min} is the minimum Hamming distance between any pair of valid codewords, then for any n-tuple x, there exists a valid codeword within a Hamming distance $t = \lfloor \frac{d_{min}}{2} \rfloor$, the maximum number of errors correctable by the linear block code. Further, if the Hamming distance between x and y is $\leq t$ we have,

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

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 respectively x and y, 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 (which is the generator matrix of the dual space of the code space). Then the projection of n-tuple x and y onto the dual space results in the syndromes $S_x = xH^T$ and $S_y = yH^T$ i.e.,

$$\begin{aligned} xH^T &= c_x H^T + e_x H^T = 0 + S_x \\ yH^T &= c_y H^T + e_y H^T = 0 + S_y \end{aligned} \tag{8}$$

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

$$S_y = yH^T = xH^T + e_c H^T = S_x + S_c$$

i.e.,

$$S_c = S_x + S_y \tag{9}$$

Equation (9) results from the fact that addition and subtraction are equivalent in $GF(2^m)$ for any integer m. Note that the syndromes are $(n-k)$ tuples. This result leads to the method of compression of X with the knowledge on correlation with Y and the lossless decoding with the knowledge of 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. With RS code, Berlekamp-Massey algorithm provides an iterative decoding procedure that eliminates the need for storing syndromes and error patterns. The n-tuple x can be found from,

$$x = y + e_c \tag{10}$$

Since the n-tuple x is transformed into the $n-k$ tuple S_x prior to transmission, we achieve a compression ratio of $\frac{n}{n-k}$. In the design of joint compression and encryption, the transform used for compression, namely the parity check matrix of the underlying linear block code, should achieve the required spreading, or the *diffusion* otherwise achieved by the row shifting and column mixing operations in the AES cipher. Diffusion is a requirement in cipher to achieve robustness against (a) differential cryptanalysis and (b)linear cryptanalysis. It has been shown in [13] that the diffusion

can be effectively measured using the branch number of a function. The Definitions 1-2 and Lemma 4 provides a concise description of relevant branch number functions and their properties.

Definition 1: The differential branch number of a transformation ϕ mapping a n-tuple onto a l-tuple is defined as

$$\mathcal{B}_d^{diff} = \min_{d_H(x_1,x_2)\neq 0} \{d_H(x_1,x_2) + d_H(\phi(x_1),\phi(x_2))\} \quad (11)$$

where x_1 and x_2 are two input n-tuples $(x_1 \neq x_2)$ and d_H is the Hamming distance in number of symbols [13].

Definition 2: The linear branch number of a transformation ϕ on mapping a n-tuple x onto a l-tuple is defined as

$$\mathcal{B}_d^{lin} = \min_{x\neq 0}\{w(x) + w(\phi(x))\} \quad (12)$$

where $w(.)$ is the Hamming weight in number of non-zero symbols.

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

Proof: With a diffusion optimized transform ϕ, change in a single symbol of 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 *diffusion layer* in Rijndael cipher adopted in AES, ensures this upper bound for all possible values of linear/differential weights of the input [14]. 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 be a *totally positive matrix*. Formal definition of totally positive matrix is as follows.

Definition 3: A rectangular matrix $\mathcal{A} = (a_{ij}), i = 1,\cdots,n; j = 1,\cdots,l$ is called *totally positive* if all its minors (determinants of sub-matrices) of any order are positive [12].

Although the original definition in [12] 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 n dimensional space V^n into l-tuples in $l(\leqslant n)$ 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 the necessary condition to satisfy the branch number properties is the total positivity. From Definitions 1, 2, and Lemma 4, for transformation $\mathcal{A}$ to be diffusive, we require that

$$d(x_1,x_2) + d(x_1\mathcal{A}, x_2\mathcal{A}) \geqslant l+1$$

$$\Rightarrow w(x_1 \oplus x_2) + w(x_1\mathcal{A} \oplus x_2\mathcal{A}) \geqslant l+1 \quad (13)$$

Since $\mathcal{A}$ is a linear transformation, (13) implies

$$w(x_1 \oplus x_2) + w((x_1 \oplus x_2)\mathcal{A}) \geqslant l+1 \quad (14)$$

Let $x_1 \oplus x_1 = e$. Then (14) reduces to

$$w(e) + w(e\mathcal{A}) \geqslant l+1 \quad (15)$$

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

TABLE I

MINIMUM CHANGE IN THE OUTPUT TO MAINTAIN BRANCH NUMBER.

The minimum values of $w(e\mathcal{A})$ corresponding to the values of $w(e)$ to satisfy (15) are as given in Table I.

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 $h_j, j = 1,\cdots,l$. Then with a given r for $r = 1,\cdots,l$ we require $\mathcal{A}$ to have at most $r-1$ columns such that $e \cdot h_j = 0$. This implies that in the $r \times l$ sub-matrix formed by selecting the rows of $\mathcal{A}$ corresponding to the non-zero elements of e, every $r \times r$ sub-matrix (contiguous as well as non-contiguous) should be of full rank. Since the r non-zero elements in e can occur at any r out of n positions, the above implies that every $r \times r$ sub-matrix of $\mathcal{A}$ should be of full rank *i.e.*, positive for $r = 1,\cdots,l$. 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 $r \times r$ sub-matrix is positive *i.e.*, has full rank for $r = 1,\cdots,l$. Let the rows of $\mathcal{A}$ be $a_i, i = 1,\cdots,n$. 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 atmost $r-1$ 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 property, the case of linear branch number property is implicit. ■

From the Theorem 2 above, 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 sub matrices of a matrix for positivity has an exponential order complexity, Theorem 9 of [17] provides a method of polynomial order complexity. This theorem states that a square matrix of size 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 [12] given by

$$\begin{pmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{(p-1)} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{(p-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_q & a_n^2 & \cdots & a_n^{(p-1)} \end{pmatrix} \tag{16}$$

where $0 < a_1 < a_2 < \cdots < a_n$.
It is seen that the parity check matrix of non-systematic RS code as in (17) is a good example of Vandermonde matrix. Therefore RS code in non-systematic form provides a readily applicable transform for diffusive compression by syndrome forming.

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{(p-1)} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(p-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^q & \alpha^2 & \cdots & \alpha^{q(p-1)} \end{pmatrix} \tag{17}$$

In (17), α is a root of the *primitive polynomial* used in the RS code.

It is also possible to obtain a transform for diffusive compression starting with a systematic RS code. In essence, we are required to suitably augment the parity check matrix of the RS code under consideration. In the usual implementation of linear block codes, the generator matrix is in the systematic form and one can obtain the parity check matrix H with straightforward manipulation of generator matrix. The H matrix in systematic form in this scenario is given by,

$$H = \begin{pmatrix} P & I \end{pmatrix} \tag{18}$$

where P is an $(n-k) \times k$ full rank matrix and I is a $(n-k) \times (n-k)$ identity matrix. Obviously, this H matrix may not comply with the total positivity criteria and therefor is not suitable for diffusive compression. To remove this effect, the syndrome S_x can be post multiplied by a mixing transform or equivalently, the matrix H can be augmented to achieve H_a by the following transformation.

$$S_x A = x(H^T \cdot A) = x H_a^T \tag{19}$$

where

$$H_a^T = \begin{pmatrix} P^T \\ I \end{pmatrix} A \tag{20}$$

or

$$H_a^T = \begin{pmatrix} P^T \cdot A \\ A \end{pmatrix} \tag{21}$$

In this, A is a $(n-k)\times(n-k)$ full rank matrix. As an example, the following matrix corresponds to an RS code systematic form with $n = 7$, $k = 3$, and $m = 8$.

$$H = \begin{pmatrix} 218 & 145 & 30 & 1 & 0 & 0 & 0 \\ 112 & 130 & 216 & 0 & 1 & 0 & 0 \\ 126 & 161 & 231 & 0 & 0 & 1 & 0 \\ 255 & 177 & 116 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The following matrix A can be used obtain augmented matrix.

$$A = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{pmatrix}$$

Thus we have

$$H_a = \begin{pmatrix} 187 & 210 & 159 & 2 & 1 & 1 & 3 \\ 18 & 167 & 28 & 3 & 2 & 1 & 1 \\ 73 & 228 & 204 & 1 & 3 & 2 & 1 \\ 203 & 146 & 26 & 1 & 1 & 3 & 2 \end{pmatrix}$$

where the entries of the matrices are decimal representation of the elements in $GF(2^8)$. Initial minor test confirms that this transform satisfies total positivity and hence achieves branch number property.

A. Special cases

It is seen in our approach to compression, all the n-tuples in a coset get mapped onto the same syndrome. Nevertheless, the minimum Hamming distance between a pair of n-tuples in the same coset being $d_{min} = (n-k)+1$, the branch number property namely the condition $d_H(x_1, x_2) + d_H(S_{x_1}, S_{x_2}) \geq |S_x| + 1$ is satisfied since $|S_x| = n - k$ for a pair of n-tuples from the same syndrome. It should also be noted that an zero input will result in an all zero output and therefore the linear branch number property is not achieved in this case. Nevertheless, this special case exist in any cipher system.

IV. Implementation and simulation Results

In the implementation of joint compression and encryption scheme, the compression is included in the first layer of ten round AES cipher as the diffusion layer as shown in Fig. 1. The row shifting and column mixing operations in the first round is replaced by the High Diffusion Distributed Compression (HDDC). Similarly, during the decryption, the inverse-column mix and inverse-row shift operations of the last round is replaced by the High Diffusion Distributed Decompression (HDDD). In the software implementation of our joint distributed compression and encryption scheme, we used $(7, 3)$ RS code i.e., $n = 7, k = 3$ with the following parity check matrix of elements in $GF(2^8)$ obtained with the primitive polynomial in $GF(2)$ given by $\alpha^8+\alpha^4+\alpha^3+\alpha^2+1$ (285 decimal).

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

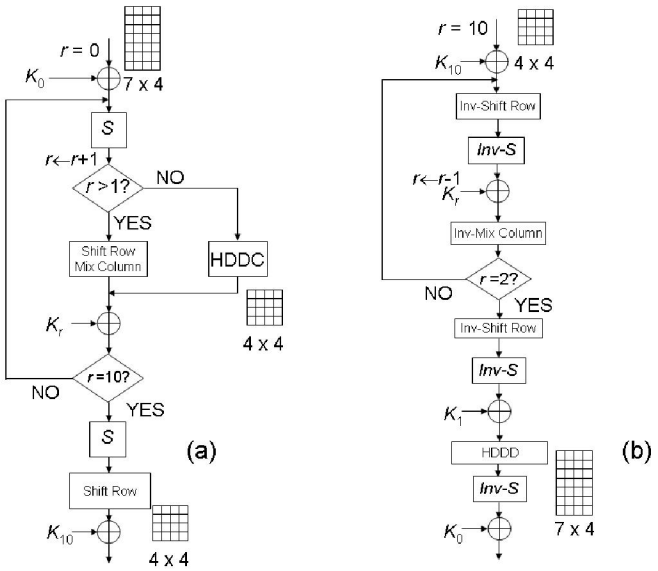

Fig. 1. Flow chart of the proposed joint distributed compression and encryption (a) compression/encryption (b) decompression/decryption.

A. Compression and savings in computation

This implementation achieves a lossless compression ratio of $\frac{n}{n-k} = \frac{7}{4}$. This compression ratio remains same for both systematic and non-systematic forms for the code. Further, there is no penalty in compression ratio due to the use of the compression process as the diffusive layer in the round based cipher. Nevertheless, savings in computation results compared to a concatenated compression-encryption system. In particular, the savings are in the row shifting and column mixing operations otherwise necessary in the diffusion layer of the AES encryption otherwise necessary in the concatenated approach. Since the diffusion layer in a round is known to eliminate the correlation between any two blocks of data, extension of the compression process to more than one layer appears to be hard at this time[1].

In the conventional AES cipher, 128 bit blocks of data are arranged in a 4×4 matrix [13]. This matrix of data undergo initial key addition and substitution phase. Each of the round functions to follow consists of a diffusion layer implemented by the row shifting and column mixing operation followed by the addition of round key and substitution. In the proposed joint compression and encryption scheme, we start with a matrix of 7×4 bytes of data. Each column of 7 bytes are compressed using syndrome forming transform obtained from the $(7, 3)$ RS-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 complexity of the joint compression and encryption approach compared to a concatenated system in a layer (compression followed by encryption) is as follows. As for the computational complexity of 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 highly dependent on the particular architecture. Nevertheless with reasonably optimized architecture, energy consumptions for these atomic operations will be comparable and may not be drastically different at the least. In the joint approach, we start with a matrix of 7×4 bytes of row data. 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 compression stage 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 that many additions. The column mixing 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 joint approach 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 amount computation 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

[1] We are currently investigating the possibilities to extend the compression to more than one layer with the use of successive projection onto dual space of RS coding.

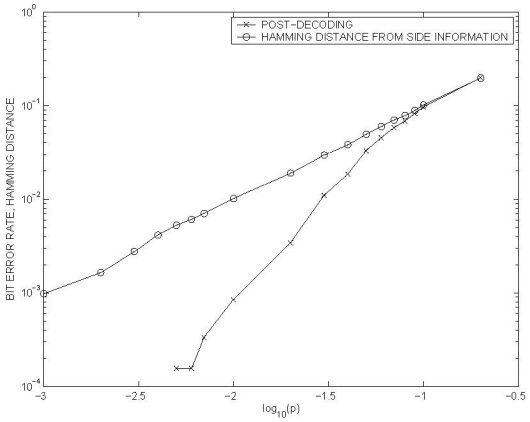

Fig. 2. Decoder performance of joint distributed compression and encryption of correlated data.

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 shows 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.

B. Decoding Error Performance

In this section, we illustrate the decoder performance in terms of the decoding error rates. Basically, since the Reed-Solomon code being used in the compression has the capability to decode the transmitted information perfectly when the Hamming distance between the compressed block of data and the side information is within the correctable limit. In our simulations, the source to be compressed was a stream of iid bits where a bit assumes $0/1$ with equal probability. The side information was a noise corrupted version of the source. The noise was modeled using binary symmetric channel (BSC) with a variable cross-over probability p. The source was jointly compressed and encrypted 28 bytes at time resulting in a 16 byte block thus achieving a compression ration of 1.75. The diffusive compression layer processes data blocks of 7 bytes at time to output a 4 byte compressed data to transmit. When the number of bytes in error are less than or equal to 2, the data is perfectly. A byte is in error whenever one or more bits in the byte are in error.

Fig. 2 shows the post-decoding bit error rate against the cross-over probability p. Below $p = 0.005$, the decoding error is observed to be zero (within a significantly large sample size). Also shown in the figure is the Hamming distance between the source and the side information (bits) as a fraction of the number of bits in the block. As p is increased the decoding error approaches the Hamming distance and the two converge around $p = 0.1$.

V. Conclusion

In this paper We presented a joint approach to distributed compression and encryption of correlated data such as in wireless sensor networks. It was shown that under key addition and substitution primitives of encryption process, the correlation between blocks of data is preserved leading to the possibility of compression as per Slepian-Wolf theorem along with such encryption primitives. We also presented theorems establishing the necessary and sufficient conditions for diffusive transform such that to achieve the branch number property required in the diffusion layer of state of the art data encryption schemes. These theorems provided systematic methods to design compression functions via linear block codes so that to achieve diffusion property required in joint distributed compression and encryption. We presented examples of diffusive transform suitable for compression based on RS-codes and discussed the implementations. We showed that it is possible to achieve significant savings in computational complexity by the joint distributed compression and encryption compared to a concatenated approach. Further savings in computational complexity will be possible with extension of the approach to more than one layer of the round based cipher.

VI. Acknowledgment

This work was partially supported by NSF-CT grant No.0627688.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, ser. Wiley Series in Telecommunicatons. New York: Wiley-Interscience, 1991.

[2] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, 1973.

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

[4] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*, 2nd ed. New York: Wiley, 1996.

[5] M. Johnson, D. Wagner, and K. Ramchandran, "On compressing encrypted data without the encryption key" in *TCC*, 2004, pp. 491–504.

[6] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code construction for the entire Slepian-Wolf rate region for arbitrary correlated sources," IEEE Data Compression Conference (DCC) 2004.

[7] G. S. Vernam, "Secret signaling system," U.S. Patent 1310719, July 1919.

[8] B. Gladman, "A Specification for The AES Algorithm," V3.11, September 2003, http://www.comms.scitech.susx.ac.uk/fft/crypto/aesspec.pdf

[9] FIPS: Specification for the advanced encryption standard (AES). Federal Information Processing Standards Publication 197 (2001)

[10] C. E. Shannon, "Communication Theory of Secrecy System," Now declassified confidential report, 1946.

[11] D. R. Stinson, *Cryptography: Theory and Practices*, ser. Discrete Mathematics and its Applications, K. H. Rosen, Ed. 2000 Corporate Blvd., N.W., Boca Raton, Florida 33431: CRC Press Inc., 1995.

[12] F. R. Gantmacher, *The Theory of Matrices-Vol. 2*, Chelsa Publishing Company, New York, N.Y.,1964.

[13] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer-Verlag, Secaucus, NJ, 2002.

[14] J. Daemen, V. Rijmen, AES Proposal: Rijndael, http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael.pdf

[15] R. Cristescu and B. Beferull-Lozano and M. Vetterli, "On Network Correlated Data Gathering," IEEE INFOCOM 2004.

[16] P. Rogaway, "OCB Mode Proposal to NIST for a block-cipher mode of operation which simultaneously provides privacy and authenticity," http://eprint.iacr.org/2001/026.pdf

[17] S. Fomin and A. Zelevinsky, "Total Positivity:Tests and Parameterizations," http://arxiv.org/PS_cache/math/pdf/9912/9912128.pdf, arXiv:math.RA/9912128, vol. 1, Dec 15,1999.