Published in IET Communications
Received on 23rd January 2008
Revised on 7th July 2008
doi: 10.1049/iet-com:20080050

ISSN 1751-8628

Efficient secure channel coding based on quasi-cyclic low-density parity-check codes

A.A. Sobhi Afshar[1] *T. Eghlidos*[2] *M.R. Aref*[1]

[1]*Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*
[2]*Electronics Research Center, Sharif University of Technology, Tehran, Iran*
E-mail: aas.afshar@gmail.com

Abstract: A secure channel coding (joint encryption-channel coding) scheme provides both data security and reliability in one combined process to achieve faster processing and/or more efficient implementation. The issue of using quasi-cyclic low-density parity-check (QC-LDPC) codes in a symmetric-key secure channel coding scheme is addressed. A set of this class of LDPC codes has recently been recommended by the NASA Goddard Space Flight Center for near-earth and deep-space communications. The proposed scheme provides an efficient error performance, an acceptable level of security and a low-complexity practicable implementation. The results indicate that the proposed scheme can efficiently employ large QC-LDPC codes to achieve a relatively smaller secret-key size to be exchanged by the sender and the receiver, and higher information rates in comparison with the previous symmetric-key McEliece-like schemes. Simulation results indicate that there is no trade-off between the error performance and the security level of the proposed scheme unlike that of the previous ones. These characteristics make the proposed scheme suitable for high-speed communications, such as satellite communication systems.

1 Introduction

Combining security and channel coding processes is an attractive idea since it may reduce the overall processing cost or provide a faster and a more efficient implementation. In 1978, McEliece [1] proposed the first public-key cryptosystem based on algebraic coding theory. The security of the scheme is based on the fact that the decoding problem for general linear codes is NP-complete [2]. For an equivalent security level, it has the advantages of higher encryption/decryption speed and easier key generation number-theoretic cryptosystems [3, 4]. The McEliece scheme employs probabilistic encryption [5], which is better than other types of deterministic encryption in preventing the partial information leakage through public-key cryptography. Despite these advantages, the McEliece public-key cryptosystem is not widely used. This is because of its large public-key size and low information rate. However, some modifications [6, 7] have been proposed to improve the information rate of the McEliece scheme at the cost of higher encryption/decryption complexity, and to reduce the key to a relatively small size [4, 8, 9]. Another extension to the McEliece scheme was developed by Jorissen [10]. The idea was to add only $t' < t$ errors, such that $t - t'$ additional errors can be corrected. This implies that there will be a trade-off between the error-correcting capability and the security level in this modification as a joint scheme.

It is well known that public-key cryptosystems can be used as symmetric-key cryptosystems. In 1984, Rao [11] proposed a symmetric-key variant of the McEliece scheme, which could also be used as a joint encryption error correction scheme. Several modifications to this scheme have been developed, some of which employ the entire error-correcting capability of the code to achieve an acceptable level of security. These schemes are known as code-based cryptosystems. Rao and Nam [12] modified this symmetric-key scheme to enable the use of simpler codes to decrease the key size and increase the information rate. The Rao–Nam (RN) scheme is subject to some chosen-plaintext attacks [13–16]. Some modifications to the RN symmetric-key cryptosystem were proposed in [14, 15, 17]. These schemes are based on either allowing nonlinear

codes or modifying the set of allowed error patterns. The RN scheme using Preparata codes [14, 17] was proved to be insecure against a chosen-plaintext attack [18]. The RN scheme using burst-error correcting codes, such as Reed–Solomon (RS) codes [12], was also proved to be insecure [13]. The use of burst-error correcting codes for symmetric-key cryptosystems was generalised in [19, 20]. The idea of these two cryptosystems is based on the fact that the burst-error correcting capacity of a binary linear block burst-error correcting code is, in general, larger than its random error-correcting capacity. Sun and Shieh [21] and Al Jabri [22] showed that these two schemes are insecure against chosen-plaintext attacks.

In 1995, J. Campello de Souza and R.M. Campello de Souza [23] proposed a symmetric-key encryption scheme (CDS scheme) based on product codes. The idea of their scheme is to use a product code that is capable of correcting a special kind of structured errors and then disguise it as a general linear code. This makes it unable to correct the errors as well as their permuted versions. In 1998, Sun and Shieh showed [24] that the CDS scheme is still insecure against chosen-plaintext attacks. Then, they proposed a modified symmetric-key cryptosystem based on product codes, which is secure against the chosen-plaintext attacks proposed to break the CDS scheme and other similar schemes. This scheme uses matrix modular addition and multiplication operations in congruence classes, which increases the encryption/decryption complexity in comparison with previous schemes. Another scheme was suggested by Barbero and Ytrehus [25] to reduce the memory requirement and the design complexity in generating error vectors. This scheme employs nonlinear transformations to provide an acceptable level of security, which increases the design and implementation complexity to a great extent. Another scheme based on MDS codes was proposed by Xu [26], which can provide security by using the erasure-correcting capability of the code.

In the last decade, the introduction of new coding techniques, namely turbo and low-density parity check (LDPC) codes, has allowed to achieve near-capacity transmission over the additive white gaussian noise (AWGN) channel. Turbo codes have been employed in two different symmetric-key secure channel coding schemes in [27, 28]. Two other schemes have been proposed to use LDPC codes in a 'public-key' code-based cryptosystem [9, 29, 30].

LDPC codes, invented in [31], were later rediscovered and analysed in [32]. Regular LDPC codes, introduced in [31], have been extended to irregular LDPC codes, which show improved error performance [33]. The last several years of research have shown that LDPC codes can provide an error-correcting performance comparable with, or better than, that of turbo codes [34], with additional benefits. Compared with turbo decoders, LDPC decoders use a structure more suitable for high-speed hardware implementation, and also require fewer computations per decoded bit. Moreover, LDPC codes have the following advantages [35]: (1) they do not need a long interleaver to achieve good error performance; (2) their error floor occurs at a much lower BER; and (3) their decoding is not trellis-based.

For the above reasons, there has been interest in selecting a family of LDPC codes as the new NASA Consultative Committee for Space Data Systems (CCSDS) telemetry channel coding standard for deep-space applications [36–38]. Quasi-cyclic(QC) LDPC codes have an encoding advantage over other types of LDPC codes. Moreover, well-designed QC-LDPC codes have been shown to perform as well as computer-generated random LDPC codes, regular or irregular [39–41], in terms of bit-error performance, block-error performance and error floor, collectively. A set of QC-LDPC codes has recently been recommended by the NASA Goddard Space Flight Center for NASA's near-earth and deep-space high-speed satellite communications [42]. These codes were selected based on certain metrics among which are the error-correcting capability of the code, the computational complexity of the decoder, the architectural complexities of the encoder and decoder, and the ease of generating a family of structurally related codes at different code rates and block lengths.

In this paper, we discuss the efficiency and security of using QC-LDPC codes in a proposed symmetric-key secure channel coding scheme. Based on the methods recently presented for encoding of QC-LDPC codes [43], encoding/encryption of our scheme can efficiently be implemented in addition to practicable implementation of decoding/decryption. Moreover, the methods addressed to design LDPC codes in this paper can generate codes with high rates because of the design parameters [39]. This implies that the designer will have degrees of freedom both to determine the desired information rate and to choose other system parameters for the required error performance and security level because of the flexible construction of QC-LDPC codes.

The rest of this paper is organised as follows. In Section 2, we recall basic facts about QC codes and the construction of QC-LDPC codes based on finite geometries. We introduce a new code-based cryptosystem in Section 3 and then, we modify the design to obtain the new secure channel coding scheme. In Section 4, we discuss the comparative efficiency (key size, implementation complexity, error performance) and security of the proposed secure channel coding scheme. Finally, Section 5 concludes the paper with some remarks.

2 Introduction to QC-LDPC codes

Although iterative decoding of long LDPC codes can be practically implemented, encoding of these codes can be complex, since most of these codes, especially

doi: 10.1049/iet-com:20080050

computer-generated random codes, do not have sufficient structure to allow simple encoding. One class of structured LDPC codes, which permits low-complexity encoding, is the class of QC-LDPC codes [35, 44].

2.1 QC codes

Cyclic codes are a class of codes, which possess full cyclic symmetry; that is, cyclically shifting a codeword any number of cyclic shifts, either to the right or to the left, results in another codeword. This cyclic symmetry makes it possible to implement the encoding and decoding of cyclic codes using shift registers and logic circuits in a simple manner. There are other linear block codes that do not possess full cyclic symmetry but do have partial cyclic structure, namely, QC codes.

Definition 2.1: A code of length n is called quasi-cyclic of order n_0, for n a multiple of n_0, if every cyclic shift of a codeword by n_0 coordinates is again a codeword [35, 45].

It is clear that for $n_0 = 1$, a QC code is a cyclic code. The integer n_0 is called the 'shifting constraint'. If C is a QC code with a shifting constraint n_0, then the dual of C is also QC with the same shifting constraint. This is deduced from the fact that a code and its dual have the same automorphism group.

A general form for the generator matrix of a (mn_0, mk_0) QC code with a shifting constraint n_0 is

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}_0 & \boldsymbol{G}_1 & \cdots & \boldsymbol{G}_{m-1} \\ \boldsymbol{G}_{m-1} & \boldsymbol{G}_0 & \cdots & \boldsymbol{G}_{m-2} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{G}_1 & \boldsymbol{G}_2 & \cdots & \boldsymbol{G}_0 \end{pmatrix} \tag{1}$$

where each $\boldsymbol{G}_i$ is a $k_0 \times n_0$ submatrix. As it is observed, relation (1) displays the cyclic structure among the rows and columns in terms of the submatrices $\boldsymbol{G}_i$'s. We can also put $\boldsymbol{G}$ in the following form

$$\boldsymbol{G} = [\boldsymbol{M}_0, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_{m-1}]$$

where $\boldsymbol{M}_j$ with $0 \leq j < m$ are $mk_0 \times n_0$ submatrices which denote the jth columns of $\boldsymbol{G}$. For $0 \leq l < n_0$, let $\boldsymbol{Q}_l$ be the $mk_0 \times m$ submatrix that is formed by taking the lth columns from $\boldsymbol{M}_0, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_{m-1}$. Then, we can put $\boldsymbol{G}$ in the following form

$$\boldsymbol{G}_c = [\boldsymbol{Q}_0, \boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_{n_0-1}]$$

where the subscript c stands for a 'circulant' form. Each column of $\boldsymbol{Q}_l$ consists of mk_0 bits that are regarded as m k_0-bit bytes. In terms of bytes, $\boldsymbol{Q}_l$ is regarded as an $m \times m$ 'circulant' matrix.

Definition 2.2: A circulant is a square matrix in which each row is the cyclic shift (one place to the right) of the row above it, and the first row is the cyclic shift of the last row [35, 45].

For such a circulant, each column is the downward cyclic shift of the column on its left, and the first column is the cyclic shift of the last column. The row and column weights of a circulant are the same, and equal to γ. For simplicity, we say that the circulant has weight γ. Therefore $\boldsymbol{G}_c$ consists of n_0 circulants. Detailed information on QC codes, their construction and encoding can be found in [35, 45].

2.2 Construction of QC-LDPC codes

LDPC codes can be constructed algebraically based on the points and lines of finite geometries, such as Euclidean and projective geometries over finite fields [44]. In this paper, we consider the m-dimensional Euclidean geometry EG(m, q) over GF(q), where q is a power of a prime [35, 45]. This geometry consists of q^m points, and each point is represented by an m-tuple over GF(q). The point represented by the all zero m-tuple is called the origin. There are $J = q^{(m-1)}(q^m - 1)/(q - 1)$ lines in this geometry, and each line consists of q points. Each point in EG(m, q) is intersected by $(q^m - 1)/(q - 1)$ lines. Two lines in the EG(m, q) are either disjoint or intersect at one and only one point. A line that contains the origin is said to pass through the origin.

Now, we consider EG*(m, q) as the subgeometry of EG(m,q), obtained by removing the origin and all the lines passing through the origin. Therefore EG*(m, q) consists of $n = q^m - 1$ non-origin points and $J_0 = (q^{(m} - 1) - 1)(q^m - 1)/(q - 1)$ lines not passing through the origin of EG(m, q). Let GF(q^m) be the extension field of GF(q). Each element of GF(q^m) can be represented as an m-tuple over GF(q). It is known that GF(q^m) can be considered as a realisation of EG(m, q). Let α be a primitive element of GF(q^m). Then, $\alpha^0, \alpha, \ldots, \alpha^{n-1}$ represent $n = q^m - 1$ non-origin points of EG*(m, q). The incidence vector of a line $\boldsymbol{L}$ in EG*(m, q) is defined as an m-tuple over GF(2), $\boldsymbol{v}_L = (v_0, v_1, \ldots, v_{n-1})$, whose components correspond to the non-origin points of EG*(m, q), where $v_i = 1$ if and only if α^i is a point on $\boldsymbol{L}$, and $v_i = 0$, otherwise. The weight of $\boldsymbol{v}_L$ is q. For $0 \leq i \leq n$, let $\boldsymbol{v}_L^{(i)}$ be the vector obtained by cyclically shifting $\boldsymbol{v}_L$, i places to the right. For $i = 0$ and n, $\boldsymbol{v}_L^{(0)} = \boldsymbol{v}_L^{(n)} = \boldsymbol{v}_L$. It can be shown that $\boldsymbol{v}_L^{(i)}$ is the incidence vector of another line in EG*(m, q) [44]. Furthermore, for $0 \leq i, j < n$ and $i \neq j$, $\boldsymbol{v}_L^{(i)} \neq \boldsymbol{v}_L^{(j)}$ [44]. This indicates that $\boldsymbol{v}_L$ and its $n - 1$ cyclic shifts give n incidence vectors of n different lines in EG*(m, q). A vector with this property is said to be primitive. As a result of this primitive property, the incidence vectors of J_0 lines in EG*(m, q) can be partitioned into $K_0 = (q^{(m-1)} - 1)/(q - 1)$ cyclic classes, each consisting of n incidence vectors. The n incidence vectors in a cyclic class can be obtained by cyclically shifting any vector in the class n times.

For each cyclic class, we choose a representative and the rest of the incidence vectors are generated by cyclically shifting this representative. Now, we form an $n \times n$ circulant $\boldsymbol{H}_i$ with the representative as its first column and its downward cyclic shifts as next columns. This results in a class of K_0 $n \times n$ circulants, that is $\boldsymbol{S} = \{\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_{K_0}\}$. The weight of each circulant is q, which is small compared with the size of circulant $n = q^m - 1$ for $m \geq 2$. Hence, each circulant in $\boldsymbol{S}$ is a sparse matrix. Since the set of columns of a circulant is the same as the set of rows, the rows of each circulant $\boldsymbol{H}_i$ are also incidence vectors of n lines in EG$^*(m, q)$. A QC-LDPC code C_{qc} is given by the null space of an array $\boldsymbol{H}_{qc}$ of sparse circulants of the same size [35]. If no two rows (or two columns) of $\boldsymbol{H}_{qc}$ have more than one 1-component in common, then the Tanner graph [46] of C_{qc} is free of cycles of length four [44], and hence, has a girth of at least 6 [44]. Such a constraint on rows and columns is called the row–column (RC) constraint. If all the columns of $\boldsymbol{H}_{qc}$ have constant weight γ and all the rows have constant weight ρ, C_{qc} is said to be regular; otherwise, it is said to be irregular. It follows from the structural properties of lines in EG(m, q) that no two rows (or two columns) either from the same circulant or from two different circulants in $\boldsymbol{S}$ can have more than one 1-component in common. This implies that: (1) each circulant in $\boldsymbol{S}$ satisfies the RC constraint; (2) any two circulants in $\boldsymbol{S}$ either arranged in a row or a column also satisfy the RC constraint, called the pairwise RC constraint.

Based on the circulants in $\boldsymbol{S}$, various methods have been proposed to construct QC-LDPC codes [39–41], [47–49]. An interesting aspect of these methods, strictly related to security analysis, concerns the possibility of generating a large number of different codes with the same dimension and length, as well as identical row and column weights in the parity-check matrix. We will discuss this case in more detail in Section 4. We mention the following instance to introduce how to construct a QC-LDPC code based on the circulants in $\boldsymbol{S}$. We can use any $N \leq K_0$ of these circulants for this purpose. Here, we make use of all the circulants in $\boldsymbol{S}$ to construct our code. We are interested in considering QC-LDPC codes having parity-check matrices $\boldsymbol{H}$ in circulant form as discussed in Section 2. We form the matrix $\boldsymbol{H}_{\mathrm{EG},qc}$ which consists of $K_0(q^m - 1) \times (q^m - 1)$ circulants concatenated as follows

$$\boldsymbol{H}_{\mathrm{EG},qc} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_{K_0}] \tag{2}$$

where $\boldsymbol{H}_i$, $1 \leq i \leq K_0$, is formed by downward cyclically shifting each representative in the ith cyclic class of incidence vectors. The null space of $\boldsymbol{H}_{\mathrm{EG},qc}$, in relation (2), gives an EG QC-LDPC code. Once we have constructed a QC-LDPC code denoted by its parity-check matrix, we can use one of the two methods presented in [43] to find the generator matrix of the QC-LDPC code from its parity-check matrix given in circulant form. Various types of highly efficient encoding implementations using simple shift registers have recently been presented [43].

In this paper, we will consider, as an instance, (2044, 1024) binary QC-LDPC code constructed based on the Euclidean geometry EG$(3, 2^3)$ over GF(2^3) $(m = 3, q = 2^3)$ in [39]. Its subgeometry EG$^*(3, 2^3)$, as discussed above, consists of 511 non-origin points and 4599 lines not passing through the origin of EG$(3, 2^3)$; each line consists of eight points. The incidence vectors of the lines in EG$^*(3, 2^3)$ can be partitioned into nine cyclic classes. From these nine cyclic classes, we can form nine 511×511 circulants, each with weight 2. A method of construction in [39] assumes the selection of a subset of these circulants to construct a family of shortened QC-LDPC codes that are themselves QC. For instance, by taking various numbers of these nine circulants, the method constructs shortened QC-LDPC codes with sizes (2044, 1024), (3066, 2046), (4088, 3068), (5110, 4090) and (8176, 7156) and the corresponding rates of 1/2, 2/3, 3/4, 4/5 and 7/8. We will consider the (2044, 1024) QC-LDPC code as a typical code of this class of QC-LDPC codes for later comparisons.

3 New scheme based on QC-LDPC codes

In this section, we first introduce a new code-based cryptosystem and then, we modify the design to obtain the proposed secure channel coding scheme. As the fundamental component of our code-based cryptosystem, we have to construct a binary QC-LDPC code $C(n, k)$ characterised by its parity-check matrix as described in Section 2. For this purpose, we choose $N \leq K_0$ vectors from K_0 representatives of the circulants in $\boldsymbol{S}$. By the use of a public predetermined method, we construct the parity-check matrix $\boldsymbol{H}$ of the QC-LDPC code in the circulant form by the use of N representative vectors and calculates the corresponding generator matrix $\boldsymbol{G}$, as addressed in Section 2. As another component of the scheme, we choose a random permutation on l coordinates (where l must be a divisor of the code length n), denoted by the submatrix $\pi_{l \times l}$, and construct $\boldsymbol{P}$ as follows

$$\boldsymbol{P} = \begin{pmatrix} \pi & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \pi & \ldots & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \pi \end{pmatrix} \tag{3}$$

The permutation matrix, in relation (3), has a block diagonal form in which the diagonal elements are identical permutation matrices of size $l \times l$, that is $\pi_{l \times l}$, and the off-diagonal elements are zero submatrices. We have chosen this special form to further reduce the key size, and more explanation will be given in Section 4. The secret key K_s consists of the N representatives compressed with a

doi: 10.1049/iet-com:20080050

known lossless compression algorithm, as discussed in the next section, and the permutation matrix $\pi_{l\times l}$.

3.1 Encryption

Now, the sender computes the ciphertext $\boldsymbol{c} = (\boldsymbol{mG} + \boldsymbol{e}(\boldsymbol{s}))\boldsymbol{P}$, where $\boldsymbol{e}(\boldsymbol{s})$ is an error vector (perturbation vector from a cryptographic point of view) obtained by first choosing a random syndrome $\boldsymbol{s} \in F_2^{n-k}$ [the set of all $(n-k)$-tuples over GF(2)] for each plaintext $\boldsymbol{m}$ and calculating

$$\boldsymbol{e}(\boldsymbol{s}) = \boldsymbol{s} \cdot (\boldsymbol{H}^{-1})^{\mathrm{T}} \tag{4}$$

where $\boldsymbol{H}^{-1}$ is a right inverse matrix of the parity-check matrix $\boldsymbol{H}$ of the QC-LDPC code. The error vectors calculated in this way are not correctable in the ordinary sense as they can have arbitrary hamming weights up to the code length. Despite this fact, these error vectors can be computed by the authorised receiver and, thus, be subtracted from the received vector. This method for choosing the error vectors was first proposed by Barbero and Ytrehus [25]. The sender and the receiver have to calculate the right inverse of $\boldsymbol{H}$, that is $\boldsymbol{H}^{-1}$, which is a matrix such that $\boldsymbol{H}_{(n-k)\times n}\boldsymbol{H}^{-1}_{n\times(n-k)}$ is the $(n-k)\times(n-k)$ identity matrix, from the definition of the right inverse in linear algebra [50, 51]. This right inverse is not unique, but the sender and the receiver can agree on a common deterministic public procedure to obtain $\boldsymbol{H}^{-1}$ from $\boldsymbol{H}$ in a unique way.

3.2 Right inverse computation algorithm

Here, we propose a method based on linear algebra on how to calculate such a right inverse in a unique way. We can formulate this problem into solving a linear system of equations. In fact, from the definition, we have the equation $\boldsymbol{H}_{(n-k)\times n}\boldsymbol{H}^{-1}_{n\times(n-k)} = \boldsymbol{I}_{(n-k)\times(n-k)}$. Now, the problem is to solve for $n\times(n-k)$ unknown elements of $\boldsymbol{H}^{-1}$ in the $(n-k)\times(n-k)$ equations obtained. If the sender and the receiver follow exactly the same public procedure for solving the equations, they will come to the same expression for the solutions and having agreed also on a way to give values to $k\times(n-k)$ extra variables in these equations, they will obtain the same right inverse. We know that the parity-check matrix $\boldsymbol{H}$ is an $(n-k)\times n$ matrix of maximum rank. Now, each side, that is the sender and receiver, should do as follows:

1. Let $h_{i,j}$ denote the element of $\boldsymbol{H}$ in the ith row and jth column.

2. Pivot on matrix elements in positions $h_{1,1}$, $h_{2,2}$, $h_{3,3}$ and so forth as far as it is possible in this order, with the objective of creating the largest possible identity matrix $\boldsymbol{I}$ in the left portion of $\boldsymbol{H}$, similar to the Gauss–Jordan method to obtain the row-reduced form of a matrix. The columns that have a pivot in positions $h_{i,i}$ are called pivotal columns.

   i. If one of these pivoting elements is zero, then, first interchange its row with the nearest lower row that has a non-zero element in the corresponding column.

   ii. If you reach a row that has a zero element in $h_{i,i}$ position and none of its lower rows has a non-zero element in the corresponding column, pivot $h_{i,i+1}$ position and continue with this strategy till all the rows are swept and, as a result, all the pivotal columns are determined.

3. Assign zeros to all the elements in the rows of $\boldsymbol{H}^{-1}$ whose row indices correspond to the column indices of the non-pivotal columns in $\boldsymbol{H}$.

4. Construct a square matrix from the pivotal columns of $\boldsymbol{H}$ by putting them as columns of this square matrix in their respective orders.

5. Calculate the inverse of this non-singular square matrix.

6. Set the rows of this inverse as the rows of $\boldsymbol{H}^{-1}$, whose row index corresponds with the column index of the pivotal column in $\boldsymbol{H}$.

In this way, the sender and the receiver will obtain a unique right inverse matrix.

3.3 Decryption

The legal receiver has to extract the parity-check matrix of the code only once from K_s and calculate its right inverse as mentioned in the above public algorithm. Then, he can decrypt as follows:

1. Apply $\boldsymbol{P}^{-1}$ to $\boldsymbol{c}$ and obtain $\boldsymbol{c}' = \boldsymbol{c}\boldsymbol{P}^{-1}$.

2. Compute the syndrome for $\boldsymbol{c}'$ and obtain $\boldsymbol{s} = \boldsymbol{c}'\boldsymbol{H}^{\mathrm{T}}$.

3. Subtract the error vector $\boldsymbol{e}(\boldsymbol{s}) = \boldsymbol{s}\,.\,(\boldsymbol{H}^{-1})^{\mathrm{T}}$ from $\boldsymbol{c}'$.

4. Decode $\boldsymbol{mG}$ and recover $\boldsymbol{m}$.

As we know, for a permutation matrix $\boldsymbol{P}$, $\boldsymbol{P}^{-1} = \boldsymbol{P}^{\mathrm{T}}$. We also note that the matrix $\boldsymbol{P}$ need not necessarily be a permutation matrix in our proposed code-based cryptosystem, which serves as a foundation for our new secure channel coding scheme. It only has to be a non-singular matrix and this considerably increases the security of the code-based cryptosystem. This is also the case for the RN scheme, but this was not mentioned in their papers [12, 14].

Thus far, we have developed a code-based cryptosystem and we will modify this scheme to obtain secure channel coding. To make use of the error-correcting capability of the QC-LDPC code, the syndrome employed in encryption

by the sender must be known to the receiver for each plaintext block synchronously. The sender and the receiver can utilise feedback shift registers (FSRs) of proper length and an initial state, which should also be regarded as part of the secret key to generate the pseudorandom syndrome synchronously. This condition is not necessarily considered as being restrictive for the design, as in most applications, such FSR-based synchronising circuits are obligatory for frame synchronisation [52].

In the proposed method, to calculate the right inverse of the parity-check matrix $\boldsymbol{H}$ of the QC-LDPC code, the dependent columns of $\boldsymbol{H}$ will yield zero columns resulting in fixed zero coordinates in $\boldsymbol{e}(\boldsymbol{s})$ as in relation (4). This was not an important case in the proposed code-based cryptosystem as $\boldsymbol{P}$ can be chosen as a non-singular matrix and prevent the information leakage from these fixed zero coordinates. To be able to exploit the error-correcting capability of the QC-LDPC code, the matrix $\boldsymbol{P}$ needs to be chosen as a permutation matrix for the reason discussed later in this section. Hence, as a simple method to further increase the security of the secure channel coding scheme by randomising the fixed zero coordinates in the vector $\boldsymbol{e}(s)$, we can add a vector $\boldsymbol{b}(\boldsymbol{s})$ to the vector $\boldsymbol{e}(\boldsymbol{s})$ in relation (4) to obtain the new perturbation vector $\boldsymbol{e}_{\mathrm{p}}$ as

$$\boldsymbol{e}_{\mathrm{p}} = \boldsymbol{e}(\boldsymbol{s}) + \boldsymbol{b}(\boldsymbol{s}) \tag{5}$$

where $\boldsymbol{b}(\boldsymbol{s})$ is a vector that has non-zero coordinates only in the places corresponding to the fixed zero coordinates of the vector $\boldsymbol{e}(\boldsymbol{s})$, that are known to the authorised sender and receiver from the computation of $\boldsymbol{H}^{-1}$. The non-zero coordinates in $\boldsymbol{b}(\boldsymbol{s})$ can be filled with $\bar{\boldsymbol{s}}$ bits, which represent the flipped (complemented) bits of the syndrome vector $\boldsymbol{s}$, or a succession of these bits in the case $n-k<k$. The reason for flipping the bits of the syndrome vector $\boldsymbol{s}$ is to prevent the condition in which a low-weight syndrome vector and a possible low-density matrix $\boldsymbol{H}^{-1}$ (considered as a weak secret key) could probably result in a low-weight perturbation vector $\boldsymbol{e}_{\mathrm{p}}$. This neutralises the expected information leakage as the probability that the perturbation vector $\boldsymbol{e}_{\mathrm{p}}$ is low-weight is drastically reduced. Based on the above discussion, we can now formulate our secure channel coding scheme.

On the condition that the transmitted block is corrupted by the channel noise, the received block can be expressed as $\boldsymbol{r} = (\boldsymbol{mG} + \boldsymbol{e}_{\mathrm{p}})\boldsymbol{P} + \boldsymbol{e}'$, where $\boldsymbol{e}'$ represents the channel noise and $\boldsymbol{e}_{\mathrm{p}}$ is the perturbation vector calculated by the sender from the relation (5). In this case, the authorised receiver will do as follows:

1. Apply $\boldsymbol{P}^{-1}$ to $\boldsymbol{r}$ and obtain $\boldsymbol{r}' = \boldsymbol{rP}^{-1} = \boldsymbol{mG} + \boldsymbol{e}_{\mathrm{p}} + \boldsymbol{e}'\boldsymbol{P}^{-1}$.

2. Compute the perturbation vector $\boldsymbol{e}_{\mathrm{p}} = \boldsymbol{e}(\boldsymbol{s}) + \boldsymbol{b}(\boldsymbol{s}) = \boldsymbol{s} \cdot (\boldsymbol{H}^{-1})^{\mathrm{T}} + \boldsymbol{b}(\boldsymbol{s})$ for the corresponding syndrome $\boldsymbol{s}$, subtract it from $\boldsymbol{r}'$ and obtain $\boldsymbol{r}'' = \boldsymbol{r}' - \boldsymbol{e}_{\mathrm{p}}$.

3. Decode $\boldsymbol{r}'' = \boldsymbol{mG} + \boldsymbol{e}'\boldsymbol{P}^{-1}$ that is a codeword corrupted with the channel error by a decoding algorithm for LDPC codes and recover $\boldsymbol{m}$.

It is noteworthy that $\boldsymbol{e}'\boldsymbol{P}^{-1}$ is an error vector with the same Hamming weight as that of $\boldsymbol{e}'$. This is because $\boldsymbol{P}^{-1} = \boldsymbol{P}^{\mathrm{T}}$ is also a permutation matrix. We also note that the matrix $\boldsymbol{P}$ in this case must be a permutation matrix (as mentioned above) since $\boldsymbol{e}'\boldsymbol{P}^{-1}$ must have the same Hamming weight as the channel error vector $\boldsymbol{e}'$. Otherwise, the correctable channel error vectors may be transformed into uncorrectable error vectors in this multiplication.

4 Comparative efficiency and security

In this section, we will evaluate the efficiency of the proposed secure channel coding scheme and will then discuss its security with respect to previous known cryptanalytic scenarios. The (2044,1024) QC-LDPC code, discussed in Section 2, is considered as a basis for our comparisons.

4.1 Efficiency

The efficiency of the proposed scheme is discussed in three categories including implementation complexity, error performance and key size compared with previous schemes. Here, we consider McEliece-like schemes which possess an acceptable level of security or efficiency with their original recommended parameters for comparison.

4.1.1 Key size: Owing to the sparse structure of the parity-check matrix, we have been able to reduce the secret key to a very reasonable size. Making use of a combination of run-length coding (RLC) and Huffman source coding (HSC) algorithms [53], a compression gain of 2 to 6 can typically be achieved in the size of the representative vectors, which comprise the main part of the secret key. Table 1 includes the results of using such an algorithm to compress the representative vectors. The codes, the parameters of which we have given in Table 1, are discussed in [35]. The original size of these representative vectors for certain QC-LDPC codes and their corresponding compression gains are given. As we expected, higher compression gains are obtained for sparser vectors.

As shown in Table 1, for the (2044, 1024) QC-LDPC code, there are eight representative vectors of length 511 resulting in the total size of 4088 bits. When compressed with the RLC-HSC lossless compression algorithm, a compression gain of 4.17 is achieved because of the sparsity of the representative vectors. Therefore the key size because of these vectors is $4088/4.17 = 980$ bits.

To further reduce the size of the exchanged key, we use a certain public procedure to store the permutation submatrix $\pi_{l\times l}$, which serves as a building block for the permutation

doi: 10.1049/iet-com:20080050

Table 1 Compression gains obtained by the RLC-HSC algorithm

QC-LDPC $C(n, k)$	Construction EG(m, 2^s)	No. of vectors	Original size, bits	Compression gain
(1533, 1161)	EG(3, 2^3)	3	1533	2.85
(2044, 1024)	EG(3, 2^3)	8	4088	4.17
(3066, 2694)	EG(3, 2^3)	6	3066	3.62
(4599, 4227)	EG(3, 2^3)	9‘	4599	5.48

matrix $\boldsymbol{P}$. Clearly, the number of keys in this case is $l!$, and so the key size (depending on the representation) is lower bounded by $\lceil \log_2(l!) \rceil$ bits. However, when l is large, it could be a complex task to map an arbitrary permutation into $\lceil \log_2(l!) \rceil$ bits and back. A straightforward encoding of π where the position of each coordinate in the permutation is given by a $\lceil \log_2(l) \rceil$-bit number, yields a key of size $\lceil \log_2(l) \rceil$ bits. It is, however, easy to obtain a more efficient representation of the key, for instance in the following way [25]. Let $l' = 2^{\lfloor \log_2(l) \rfloor}$ be the largest power of 2 smaller than or equal to l. The $l - l'$ first positions in the permutation are represented by $\log_2(l') + 1$ bits each. When these positions have been listed, the remaining l' positions can be renumbered and the next $l'/2$ positions can be represented by a $\log_2(l')$-bit number each, and so on. In this way, the required number of bits to represent the permutation is

$$\begin{aligned}&(l - l')(\log_2 l' + 1) + \sum_{i=1}^{\log_2 l'} i2^{i-1} \\ &= (l - l')(\log_2 l' + 1) + l'(\log_2 l' - 1) + 1 \\ &= l(\log_2 l' + 1) - 2l' + 1 \end{aligned} \quad (6)$$

Relation (5) yields a size of 500 bits, with $l = 73$ and $l' = 64$, to store a permutation submatrix $\pi_{73\times73}$ in this way. In fact, for $l = 73$, we will have 28 submatrices $\pi_{73\times73}$ in the permutation matrix $\boldsymbol{P}_{2044\times2044}$ aswe have $2044 = 28 \times 73$. The choice of l as the dimension of $\pi_{l\times l}$ is discussed in Section 4.2.

A comparison based on the key size of various McEliece-like schemes with their originally recommended code parameters is given in Table 2. These sizes denote the size of the secret key to be exchanged by the sender and receiver. The key size of the original Rao scheme [11] represents the total sizes of the permutation matrix, the non-singular matrix and the generator matrix of the (1024, 524) Goppa code as the three parts of the key similar to the public-key scheme proposed by McEliece [1]. There is a similar calculation of the key size for other schemes in Table 2 with their recommended parameters. The key size of the Barbero–Ytrehus scheme [25] represents their main scheme that has more reliable security. It is observed that for the (2044, 1024) large QC-LDPC code corresponding to a higher security level, we have been able to achieve a very reasonable key size of 2.5 Kbits to be exchanged. This amount is because of three factors:

1. The compressed representative vectors with respect to the result in Table 1, that is, $4088/4.17 = 980$ bits.

2. The storage of the permutation π as in the above procedure, that is, 500 bits.

3. The seed to generate the pseudorandom syndrome by the use of FSRs, which is at most $n - k = 2044 - 1024 = 1020$ bits for the (2044, 1024) code.

These yield the total secret key size of 2500 bits($\simeq$2.5 Kbits) for K_s to be exchanged.

4.1.2 Implementation complexity: Here, we will provide evidence for the practicability of the proposed scheme for the (2044, 1024) QC-LDPC code and even for codes with larger block lengths as addressed in [42] for satellite communications. We discuss the implementation complexity of our scheme for encoding/encryption and decoding/decryption algorithms. This is subject to the precomputation and computation phases.

The precomputation phase includes three parts: extraction of the parity-check matrix $\boldsymbol{H}$, computation of $(\boldsymbol{H}^{-1})^{\mathrm{T}}$ based on our proposed algorithm in Section 3 and calculation of the generator matrix from the parity-check matrix. These three parts of the precomputation phase are done only once and the result will be stored in memory as long as the secret key (K_s) is not changed. Extraction of $\boldsymbol{H}$ from the secret key K_s mainly involves the combined RLC-HSC

Table 2 Key size of the new scheme compared with other McEliece-like schemes

Scheme	Code	Key size
Rao [11]	C(1024, 524, 101)	2 Mbits
RN [12]	C(72, 64, 3)	18 Kbits
Struik–Tilburg [15]	C(72, 64, 3)	18 Kbits
Sun–Shieh [24]	C(49, 36)	42 Kbits
Barbero–Ytrehus [25]	C(30, 20) over GF(2^8)	4.9 Kbits
new scheme	C(2044, 1024, 12)	2.5 Kbits

decompression algorithm [53]. This has a complexity order of O(nL), where n is the code length and L is the maximum length of the Huffman-coded words, which is smaller than n. We also know from linear algebra that the computation of a right inverse as discussed in Section 3 is upper-bounded by a complexity order of $O(n^3)$ bit operations, where n is the code length [51]. The complexity of calculating the generator matrix from the parity-check matrix, as discussed in [43], is upper-bounded by $O(n^3)$.

In the computation phase, the complexity of each of these algorithms consists of two parts. For the encoding/encryption algorithm, the complexity includes the total complexities of encoding and encryption, and similarly for the decoding/decryption algorithm. The encryption and decryption algorithms are of the same complexity since they both calculate an error vector as in relation (4) and add the result to the codeword or the received vector. Here, we consider the computational complexity of calculating the error vectors as in relation (4) because the difference in complexity with those calculated by relation (5) is negligible. For $k \simeq n/2$, (as for the (2044,1024) code) the complexities of encoding and encryption are of the same order since they both require the multiplication of a vector (message vector $\boldsymbol{m}$ or syndrome $\boldsymbol{s}$) of an approximate size of $n/2$ by a given matrix ($\boldsymbol{G}$ or $(\boldsymbol{H}^{-1})^{\mathrm{T}}$) that has an approximate number of $n/2$ rows and exactly n columns. Each of these matrices has a certain characteristic (QC property of $\boldsymbol{G}$ and zero columns of $(\boldsymbol{H}^{-1})^{\mathrm{T}}$) that can help optimise the computation in hardware in an equivalent way. For $k > n/2$, it is reasonable to infer that the complexity of encryption is upper-bounded by the complexity of encoding. For codes with $k < n/2$, the rate is small and there is not much interest in using such codes in high-speed communications.

Based on the above discussion, we will focus on the complexity of encoder and decoder as the key factors, which can help evaluate the implementation complexity of our scheme. We will consider one of the large QC-LDPC codes, namely the (8176, 7156) code of the construction family of (2044, 1024) code discussed in Section 2. In fact, the (8176, 7156) code is the foundation for the standard code recommended by NASA Goddard Space Flight Center for near-earth satellite communications where a bit-error rate below 10^{-10} is required. For such an error rate requirement, the error floor of the code must be below 10^{-10}.

The (8176, 7156) base code needs to be modified to ease implementations for current space and ground systems as they manipulate and process data at 32-bit computer word size. It is therefore beneficial to shorten the code to the dimensions of (8160, 7136) bits or (255, 223) 32-bit words [42].

Encoder complexity: The complexity of LDPC codes has been an area of research and discussion. For a field programmable gate array (FPGA) or application specific integrated circuit (ASIC) implementation, the complexity of the encoder is dominated by two factors: the total number of required logic gates and the routing complexity. For the (8176, 7156) QC-LDPC code, the QC property allows for the use of shift registers whose required number of logic gates is linearly proportional to $(n-k)$ [43] or $8176-7156=1020$ (unshortened). With regard to the routing complexity, there is currently no way to predict this figure and would depend on a number of factors such as the choice of FPGA or ASIC, routing algorithm and layout of the device.

An FPGA implementation was used in [42] for the development of the (8176, 7156) base code. For this purpose, a Xilinx 8000 Virtex-2 FPGA was used for the test. The device contained both the encoder and decoder. It is noteworthy that an architectural evaluation has been performed prior to implementation to produce a quasi-optimal implementation based on routing, logic requirements and BER performance. The encoder algorithm was a shift register-based encoder described in [43]. The experimental results indicate that the encoder used 2535 logic slices out of 46 592 available or 5.4% and four memory blocks out of 168 available or 2.4%. The number of logic slices is an aggregate measure of the number of logic gates required and the routing complexity, whereas the memory blocks figure is the number of dedicated FPGA memory blocks used.

Decoder complexity: Now, we consider the belief propagation (BP) decoding algorithm [32, 35] for the (8176, 7156) QC-LDPC code for which the encoder complexity was discussed. The decoder complexity is larger than that of the encoder and even more difficult to predict. The primary complexity factors (the total number of required logic gates and the routing complexity) are a function of the choice of various BP decoding algorithms [4, 35, 54] as well as architectural decisions (i.e. parallel or serial processing, number of bits of finite precision, fixed number of iterations or the stopping rule, use of look-up tables and so on). These choices also determine the decoder bit error rate (BER) performance.

In the above FPGA test, the decoding algorithm was the scaled min-sum parallel BP described in [54]. The results indicate that the decoder used 21 803 logic slices out of 46 592 or 46.8% and 137 memory blocks out of 168 or 81.5%. Itis clear from these statistics that the encoder is of much lower complexity than the decoder using only 5.4% of resources of the logic slices, whereas the decoder requires 46.8%. For testing, the system clock was set to 100 MHz. The encoder data rate was limited to 200 Mbps and the decoder operated at 140 Mbps for ten iterations.

Based on the results and statistics in the above section, we will determine the complexity of our new scheme with the (2044, 1024) QC-LDPC code. As mentioned in Section 2.2, the

doi: 10.1049/iet-com:20080050

encoding complexity of an (n, k) QC-LDPC code has been shown to be linearly proportional to $n - k$ [43]. For the (2044, 1024) code, $n - k = 2044 - 1024 = 1020$, and for the (8176, 7156) code, $n - k = 8176 - 7156 = 1020$. Therefore these two codes will have the same encoding complexity. Moreover, as discussed in [39], the (2044, 1024) and (8176, 7156) QC-LDPC codes and other codes in their construction family, discussed in Section 2, can be decoded with the same decoding circuit. Hence, it is reasonable to infer that, for the (2044, 1024) QC-LDPC code, encoding/encrpytion complexity of the proposed scheme is at most twice the complexity of encoding of this code. In other words, the encoding/encyption complexity for the (2044, 1024) code will at most be twice the encoding complexity mentioned in the previous section for the (8176, 7156) code. As mentioned above, for the decryption/decoding algorithm, the complexity is at most equal to the sum of the complexities of encoding and decoding. Therefore the decryption/decoding complexity of the proposed scheme for the (2044, 1024) code is at most equal to the sum of the encoding and decoding complexities of the (8176, 7156) QC-LDPC code discussed above.

To sum up these results, the encoding/encryption of the new scheme with the (2044, 1024) QC-LDPC code can be implemented using at most 10.8% of the logic slices and 4.8% of the memory blocks on a Xilinx 8000 Virtex-2 FPGA device. For the implementation of decoding/decryption of the new scheme, 52.2% of the logic slices and 83.9% of the memory blocks of this device are required. Thus, both encoding/encryption and decoding/decryption of the new scheme can be implemented on a single FPGA with the aforementioned resources, totally making use of 63% of the logic slices and 88.7% of the memory blocks of the device.

4.1.3 Error performance: The superior error performance of LDPC codes with iterative decoding based on BP have been shown to lie only a fraction of a decibel away from the Shannon limit [33, 34]. This discovery makes LDPC codes strong competitors with the previously predominant codes, as in the CCSDS recommended standard [52], for error control in many communication and digital storage systems where high reliability is required.

On the part of error correction efficiency of the proposed secure channel coding scheme, the error performance for the typical (2044, 1024, 12) QC-LDPC code with the sum-product decoding algorithm (SPA) is depicted in Fig. 1. The performance is evaluated with binary phase shift keying (BPSK) transmission over the AWGN channel. At a BER of 10^{-3}, it achieves an approximate coding gain of 4.9 dB over the uncoded BPSK.

Next, a comparison is brought out that highlights the decoding complexity of QC-LDPC codes against RS codes. In Fig. 2, the error performance of two most commonly used RS codes for error control in data

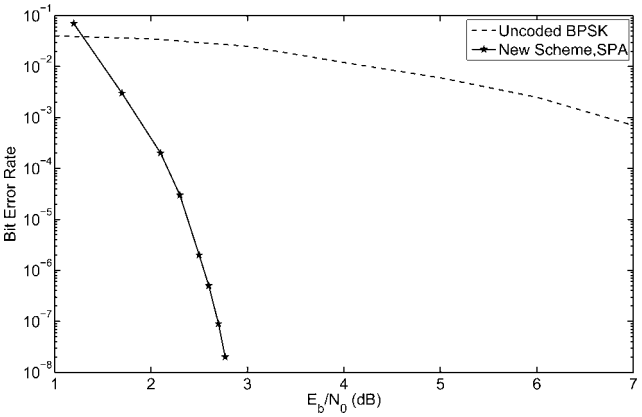

Figure 1 *Bit error performance of the proposed scheme with (2044, 1024, 12) QC-LDPC code with SPA decoding for ten iterations*

communication and storage systems is shown. These codes are the (255, 223, 33) and (255, 239, 17) RS codes over GF(2^8). The (255, 223, 33) is also a NASA standard code for space and satellite communications. Both codes are decoded with either the Berlekamp or the Euclidean algorithm. As it is observed in Fig. 2, for the (255, 223, 33) RS code, BER $= 10^{-5}$ is achieved approximately at $E_b/N_0 = 6.2$ dB. For binary transmission, each code symbol is expanded into an 8-bit byte. This symbol-to-binary expansion results in a (2040, 1784) binary code. At the receiving end, the received digits are grouped back into symbols in GF(2^8) for decoding. Their performance is evaluated with BPSK transmission over the AWGN channel.

In Fig. 3, the error performances of the proposed secure channel coding scheme on an AWGN channel with BPSK signalling for majority-logic decoding (MLD) and the SPA, as two of the common decoding algorithms for QC-LDPC codes [35], are shown. The employed code is a Euclidean geometry LDPC code of size (4088, 3068) from the construction family of the typical (2044, 1024) QC-LDPC code discussed in Section 2. In fact, it can be

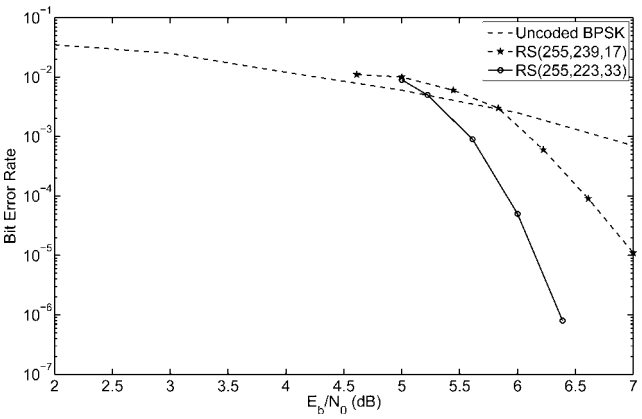

Figure 2 *Error performances for (255, 239, 17) and (255, 223, 33) RS codes [32]*

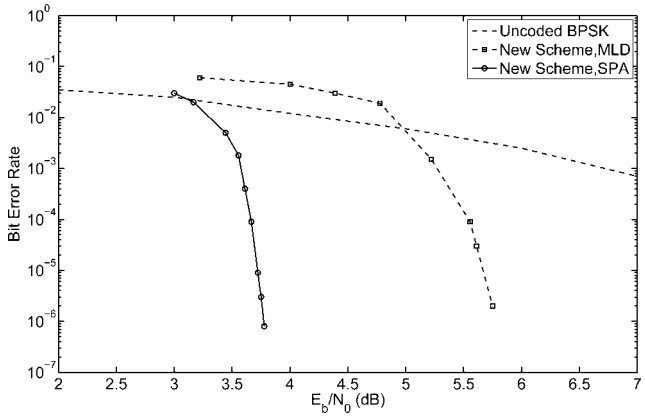

Figure 3 *Bit error performances of the proposed scheme with (4088, 3068) QC-LDPC code with MLD and SPA decoding for ten iterations*

proved that both these QC-LDPC codes have a minimum Hamming distance of at least 5 [35]. The QC-LDPC codes discussed above have no error floors down to the BER of 10^{-8}. As it is observed in Fig. 3, BER $= 10^{-5}$ for the proposed scheme with MLD algorithm is achieved approximately at $E_b/N_0 = 5.6$ dB. Now, we want to emphasise that the decoding complexity of QC-LDPC codes is considerably less than that of RS codes. The majority logic decoder for the QC-LDPC code can easily be implemented in hardware with FSRs. Even though the (4088, 3068) LDPC code is twice as long as the (255, 223, 33) RS code in binary form, its decoding complexity is much lower. This is because MLD requires only simple binary logic operations, whereas decoding of the (255, 223, 33) RS code with algebraic decoding algorithms requires computations in GF(2^8) to find the error-location polynomial and the error-value enumerator [35].

As we expected, simulation results indicate that the superior error performance of LDPC codes is unaffected in the proposed secure channel coding scheme. This implies the fact that there is no trade-off between the error performance and the security level of the proposed scheme unlike that of most previous secure channel coding schemes. Also shown in Fig. 3, the proposed scheme has achieved a much better error performance with the SPA which is an extremely efficient iterative decoding algorithm at the expense of increasing the decoding complexity compared with the MLD algorithm [32, 33, 35].

4.2 Security

The security of the proposed scheme is discussed in five categories including the analyses already developed for previous symmetric-key code-based cryptosystems. As mentioned in Section 1, some of these cryptosystems can be used as secure channel coding schemes, whereas others employ the total error-correcting capability of the code to obtain security. This is done because the security of the proposed channel coding scheme is lower-bounded by the security of the proposed code-based cryptosystem. In fact, the additive channel error vector will unpredictably corrupt the encrypted codeword and, hence, increase the computational complexity of the cryptanalysis. In each category, we first discuss the weakness(es) of the previous McEliece-like schemes, which make the proposed scenarios successful, and will then apply them to our scheme and discuss the results.

4.2.1 Brute force attacks: The first kind of known attacks are brute force attacks. The purpose is to enumerate the code set, the error vectors and the permutation. As mentioned in Section 2.2, the algorithms addressed to construct QC-LDPC codes can generate a large number of different codes with the same dimension, code length and identical row and column weights in the parity-check matrix. For instance, it has been demonstrated through combinatorial arguments that the number of different QC-LDPC codes with length $n = mn_0$, rate $R = (n_0 - 1)/n_0$ and identical row and column weight ρ, free of four-length cycles is [49]

$$N(n_0, \rho, m) \geq \frac{1}{m}\binom{m}{\rho}^{n_0} \prod_{i=0}^{n_0-1} \prod_{j=1}^{\rho-1} \frac{m}{m-j} + \frac{j\left[2 - m \bmod 2 + (j-1)^2/2 + i \cdot \rho \cdot (\rho - 1)\right]}{m-j} \tag{7}$$

This number can be considerably large from a security analysis point of view. For example, for $n_0 = 4$, $\rho = 11$, $m = 4032$, relation (6) results in $N(n_0, \rho, m) \geq 2^{391}$. Independent of their number, the codes in the family are equivalent as they share the characteristics that mostly influence LDPC decoding [49].

For the pseudorandom $(n - k)$-bit syndrome, we also have the large number of non-zero values of 2^{n-k} because of the large code parameters. This will, as a result, generate a large number of error vectors, which will be discussed in more detail in the following sections.

The number of permutations $\boldsymbol{P}$ in a block diagonal form is $l!$, where l is the number of rows/columns of the permutation submatrix $\pi_{l\times l}$. As the employed code is QC, there always exists an $1 < l < n$ so that it is a divisor of the codelength n. It is recommended a value be chosen for l which the enumeration of all $l!$ possible permutations $\pi_{l\times l}$ becomes a demanding task with regard to the design parameters of the code. For instance, $l \geq 50$ yields $l! \geq 3 \times 10^{64}$, which denotes an impractical amount of preliminary work.

4.2.2 RN attack: The symmetric-key variant of the McEliece scheme proposed by Rao [11] uses a set of errors that is restricted to the vectors of weight $t \leq \lfloor (d-1)/2 \rfloor$, where d is the minimum distance of the (n, k) code. This

doi: 10.1049/iet-com:20080050

type of cryptosystem is vulnerable to a majority voting analysis in which the secret encryption matrix is obtained in an efficient manner [12]. However, a chosen-plaintext attack of the above nature can succeed only when t/n is considerably small.

Based on the computation of the error vectors in our scheme, the generated error vectors will have a Hamming weight of at most n and $t \simeq n/2$ on average. This will make the proposed attack to our scheme infeasible.

4.2.3 Struik–Tilburg attacks: The RN scheme was proposed to increase the information rate that was one of the drawbacks of the McEliece scheme in a symmetric-key scenario. Rao and Nam modified the previous scheme [11] and used the error-correcting properties of the code to determine predefined error patterns [12]. The error patterns used in the RN scheme have an average Hamming weight of half the code length. Rao and Nam claimed that the determination of the encryption matrix of the modified scheme in the chosen-plaintext attack has a work factor of at least $O(n^{2k})$ for the (n, k) code. However, Struik and Tilburg proposed a chosen-plaintext attack that showed that the RN scheme is insecure for practical code lengths. They mentioned three weaknesses of the RN scheme. ST attacks [15] and Hin's attack with adjacent errors [13] were made possible because of these weaknesses.

The first weakness of the RN scheme is the low number of different syndromes for their recommended code parameters. This number is at most n for adjacent errors and it is q^{n-k} for a q-ary Goppa code. If the number of different error patterns is N, then one has to encrypt on average $O(N \log N)$ times to obtain all error patterns [15]. This work factor makes the proposed chosen-plaintext attack feasible for recommended code parameters of the RN scheme.

For a large (n, k) QC-LDPC code over GF(2) in our scheme (as in Table 2), we have an astronomical number of 2^{1020} error vectors for the (2044, 1024, 12) code with a reasonable key size. Therefore the attack fails because of the large amount of preliminary work involved.

The second weakness is because of the leakage of information about the permutation matrix $\boldsymbol{P}$ if adjacent errors are used. This case, which is the basis for either Hin's attack or ST's second attack, cannot be applied to our scheme because of the structure of the error vectors.

The third weakness is the possibility of estimating the rows of the encryption matrix because of the linear transformation of the scheme. The linearity of the transformation can be a drawback only when both the code length and the weight of the error vectors allow for the majority voting analysis. None of these conditions exists in our scheme as we are able to use a large code and error vectors with relatively large weights together with a small key size. In fact, the design of nonlinear transformations is not a straightforward task. There are a number of known pitfalls that should be avoided [25]. Thus, we can deduce that the use of nonlinear transformations complicates the design and is not necessarily an advantage.

4.2.4 Barbero–Ytrehus attacks: The attacks proposed by Barbero and Ytrehus [25] can be categorised into three classes. The first and foremost class includes the attacks on the nonlinearising transformation. As mentioned in their paper, the actual security of their scheme depends on the implementation of the nonlinear functions involved. As discussed in [25], the nonlinear function $f(.)$ acting on the message should be indexed by both the syndrome s and the permutation $\boldsymbol{P}$ in order to achieve the expected level of security. Some of the design criteria, discussed in [25], to obtain a secure nonlinear transformation with these conditions obviously remain as open problems or impose certain restrictions on the function. These restrictions may be unwanted since this secure nonlinear function should be at the same time as unpredictable as possible.

The second class of attacks are aimed at the so-called public key of their secondary scheme. It is worth noting that their secondary scheme, as a modification of their main scheme, requires a public key while being a symmetric-key cryptosystem. One of the potential attacks that may be thought of and was not possible at that time can take advantage of this public key with the help of the support-splitting algorithm [55]. This scenario can be successful if some information regarding the encryption matrix in their scheme is obtained.

The above two classes of attacks cannot be applied to our scheme as it requires neither the nonlinearising transformation nor the public key.

The third class of their attacks is based on the condition that the number of q^{n-k} syndromes for a q-ary code is small. These attacks, as discussed in previous scenarios, are infeasible to our scheme because of the large number of possible syndromes.

5 Conclusions

In this paper, we proposed a symmetric-key secure channel coding scheme based on QC LDPC codes. This scheme utilises a specific form of error vectors (perturbation vectors) that can be calculated from a random syndrome by the sender and receiver. An algorithm was proposed for the computation of such error vectors based on the right inverse of the parity-check matrix of the code. The error vectors computed in this way are very effective for cryptographic use since they are not necessarily small weight words (like those of the McEliece scheme) and cannot be decoded. However, they can be calculated by the authorised receiver. The security of the proposed scheme was discussed on the basis of various known cryptanalytic scenarios. Simulation results indicate that there is no

trade-off between the error performance and the security level of the proposed scheme unlike that of most previous secure channel coding schemes. This implies that it can not only take advantage of the superior error performance of LDPC codes but also provide an acceptable resistance against known security analyses.

This scheme employs QC-LDPC codes based on four distinctive reasons: (1) this class of LDPC codes have been shown to perform as well as the superior computer-generated random LDPC codes, regular or irregular, in terms of the bit/block error performance and the error floor; (2) the algorithms addressed to construct these codes can generate a large number of different codes (from the point of view of security analysis) with the same characteristics, which mainly influence LDPC decoding; (3) the encoder/decoderof these codes can be implemented in an efficient way; and (4) the low-density structure of these codes makes it possible to employ large codes in the proposed scheme and still have a small key size to be exchanged.

In addition to the above merits, the proposed scheme also takes advantage of linear codes and transformations and it can be implemented with reasonable complexity in an efficient manner, which makes it suitable for high-speed communications.

6 Acknowledgment

This work was supported in part by Iran Telecommunications Research Center (ITRC) grant T/500/1894 and by Iranian NSF grant 84.5 193.

7 References

[1] MCELIECE R.J.: 'A public-key cryptosystem based on algebraic coding theory'. DSN Progress Report, 42-44, 1978, pp. 114–116

[2] BERLEKAMP E.R., MCELIECE R.J., VAN TILBORG H.C.A.: 'On the inherent intractability of certain coding problems', *IEEE Trans. Inf. Theory*, 1978, **24**, (3), pp. 384–386

[3] PRENEEL B., BOSSELAERS A., GOVAERTS R., VANDEWALLE J.: 'A software implementation of the Mceliece cryptosystem'. Proc. 13th Symp. Information Theory in the Benelux, 1992, pp. 119–126

[4] KABATIANSKY G., KROUK E., SEMENOV S.: 'Error correcting coding and security for data networks' (John Wiley Sons Ltd., 2005)

[5] GOLDWASSER S., MICALI S.: 'Probabilistic encryption and how to play mental poker keeping secret all partial information'. Proc. 14th ACM Symp. Theory of Computing, 1982, pp. 270–299

[6] PARK C.S.: 'Improving code rate of McEliece's public-key cryptosystem', *Electron. Lett*, 1989, **25**, (21), pp. 1466–1467

[7] LIN M.C., FU H.L.: 'Information rate of McEliece's public-key cryptosystem', *Electron. Lett*, 1990, **26**, (1), pp. 16–18

[8] GABORIT P.: 'Shorter keys for code based cryptography'. Proc. CLC2006, Technische Universitat Darmstadt, September 2006

[9] BALDI M., CHIARALUCE F.: 'Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes'. Proc. IEEE Int. Symp. Information Theory 2007, Nice, France, June 2007, pp. 2591–2595

[10] JORISSEN F.: 'A security evaluation of the public-key cipher system proposed by McEliece, used as a combined scheme'. Technical Report, Department of Elektrotechniek, Katholieke University Leuven, 1986

[11] RAO T.N.R.: 'Joint encryption and error correction schemes'. Proc. 11th annual Int. symposium on Computer architecture', Ann Arbor, Mich., pp. 240–241

[12] RAO T.N.R., NAM K.H.: 'A private-key algebraic-coded cryptosystem'. Proc. Crypto'86, 1986, pp. 35–48

[13] HIN P.J.M.: 'Channel-error-correcting privacy cryptosystems', PhD, dissertation (in Dutch), Delft University of Technology, 1986

[14] RAO T.R.N., NAM K.H.: 'Private-key algebraic-code encryption', *IEEE Trans Inf. Theory*, 1987, **35**, (4), pp. 829–833

[15] STRUIK R., TILBURG J.: 'The Rao–Nam scheme is insecure against a chosen-plaintext attack'. Advances in Cryptology, Crypto'87, 1988, (*LNCS*), pp. 445–457

[16] BRICKELL E.F., ODLYZKO A.: 'Cryptanalysis: a survey of recent results', *Proc. IEEE*, 1988, **76**, (5), pp. 153–165

[17] DENNY W.F.: 'Encryptions using linear and non-linear codes: implementation and security considerations', PhD dissertation, The Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, 1988

[18] STRUIK R.: 'On the Rao-Nam scheme using nonlinear codes'. Proc. IEEE Int. Symp. Information Theory, 1991, p. 174

[19] ALENCAR F.M.R., LO A.M.P., CAMPELLO DE SOUZA R.M.: 'Private-key burst correcting code encryption'. Proc. IEEE Int. Symp. Information Theory, 1993, p. 227

[20] CAMPELLO DE SOUZA R.M., CAMPELLO DE SOUZA J.: 'Array codes for private-key encryption', *Electron. Lett.*, 1994, **30**, (17), pp. 1394–1396

doi: 10.1049/iet-com:20080050

[21] SUN H.M., SHIEH S.P.: 'Cryptanalysis of private-key encryption schemes based on burst-error-correcting codes'. Proc. Third ACM Conf. Computer and Communications Security, 1996, pp. 153–156

[22] AL JABRI A.: 'Security of private-key encryption based on array codes', *Electron. Lett.*, 1996, **32**, (24), pp. 2226–2227

[23] CAMPELLO DE SOUZA J., CAMPELLO DE SOUZA R.M.: 'Product codes and private-key encryption'. Proc. IEEE Int. Symp. Information Theory, 1995, p. 489

[24] SUN H.M., SHIEH S.P.: 'On private-key cryptosystems based on product codes'. Proc. 3rd Australasian Conf. Information Security and Privacy, 1998, pp. 68–79

[25] BARBERO A.I., YTREHUS O.: 'Modifications of the Rao–Nam cryptosystem'. Proc. Int. Conf. Coding Theory, Cryptography and Related Areas, Guanajuato, Mexico, 1998, pp. 1–13

[26] XU L.: 'A general encryption scheme based on MDS code'. Proc. IEEE Int. Symposium on Information Theory 2003, June 2003

[27] PAYANDEH A., AHMADIAN M., AREF M.R.: 'Adaptive secure channel coding based on punctured turbo codes', *IEE Proc. Commun.*, 2006, **153**, (2), pp. 313–316

[28] BARBULESCU S.A.: 'Secure satellite communications and turbo-like codes'. In JAGO, C., AMIS, K., BERROU, C. (EDS.), Proc. 3rd Int. Symp. Turbo Codes, ISTC 2003, Brest, France, 2003, pp. 227–230,

[29] MONICO C., ROSENTHAL J., SHOKROLLAHI A.: 'Using low density parity check codes in the McEliece cryptosystem'. Proc. IEEE Int. Symp. Information Theory, Italy, June 2000, p. 215

[30] BALDI M., CHIARALUCE F., GARELLO R., MININNI F.: 'Quasi-cyclic LDPC codes in the McEliece cryptosystem'. Proc. IEEE Int. Conf. Communications 2007, Glasgow, UK, June 2007, pp. 951–956

[31] GALLAGER R.G.: 'Low density parity check codes'. PhD thesis, MIT Press, 1963

[32] MACKAY D.J.C.: 'Good error correcting codes based on very sparse matrices', *IEEE Trans. Inf. Theory*, 1999, **45**, (1), pp. 399–431

[33] CHUNG S.Y., FORNEY J.G.D., RICHARDSON T., URBANKE R.: 'On the design of low-density parity-check codes within 0.0045dB of the Shannon limit', *IEEE Commun. Lett*, 2001, **5**, (2), pp. 58–60

[34] RICHARDSON T., SHOKROLLAHI A., URBANKE R.: 'Design of capacity-approaching irregular low-density parity check codes', *IEEE Trans. Inf. Theory*, 2001, **47**, (2), pp. 619–637

[35] LIN S., COSTELLO D.J.: 'Error control coding: fundamentals and applications' (Prentice-Hall, Upper Saddle River, NJ, 1983, 2nd edn., 2004)

[36] FONG W.: 'White paper for low density parity check (LDPC) codes for CCSDS channel coding blue book'. CCSDS P1B Channel Coding Meeting, Houston, TX, October 2002

[37] LIN S.: 'Quasi-cyclic LDPC codes', CCSDS Working Group White Paper, October 2003

[38] ANDREWS K., DOLINAR S., DIVSALAR D., THORPE J.: 'Design of low-density parity-check (LDPC) codes for deep space applications'. IPN Progress Report 42-159, November 2004

[39] CHEN L., XU J., DJURDJEVIC I., LIN S.: 'Near-shannon-limit quasicyclic low-density parity-check codes', *IEEE Trans. Commun.*, 2004, **52**, (7), pp. 1038–1042

[40] TANG H., XU J., LIN S., ABDEL-GHAFFAR K.: 'Codes on finite geometries', *IEEE Trans. Inf. Theory*, 2005, **51**, (2), pp. 572–596

[41] XU J., CHEN L., ZENG L.Q., LAN L., LIN S.: 'Construction of low-density parity-check codes by superposition', *IEEE Trans. Commun.*, 2005, **53**, (2), pp. 243–251

[42] CCSDS 131.1-O-1: 'Low density parity check codes for use in near-earth and deep space applications'. Research and Development for Space Data System Standards', CCSDS, Washington, DC, 2006, Orange Book, Issue 1

[43] LI Z., CHEN L., ZENG L., LIN S., FONG W.H.: 'Efficient encoding of quasi-cyclic low-density parity-check codes', *IEEE Trans. Commun.*, 2006, **54**, (1), pp. 71–81

[44] YU K., LIN S., FOSSORIER M.: 'Low density parity check codes based on finite geometries: A discovery and new results', *IEEE Trans. Inf. Theory*, 2001, **47**, (11), pp. 2711–2736

[45] PETERSON W.W., WELDON E.J.: 'Error correcting codes' (MIT Press, 1972, 2nd edn.)

[46] TANNER R.M.: 'A recursive approach to low-complexity codes', *IEEE Trans. Inf. Theory*, 1981, **27**, (5), pp. 533–547

[47] BALDI M., CHIARALUCE F.: 'New quasi cyclic low density parity check codes based on difference families'. Proc. 8th Int. Symp. Commun. Theory and Appl., ISCTA 05, Ambleside, UK, July 2005, pp. 244–249

[48] XIA T., XIA B.: 'Quasi-cyclic codes from extended difference families'. Proc. IEEE Wireless Commun. and Networking Conf., March 2005, pp. 1036–1040

[49] BALDI M.: 'Quasi-cyclic low-density parity-check codes and their application to cryptography'. PhD dissertation,

Università Politecnica delle Marche, Ancona, Italy, November 2006

[50] WU C.K., DAWSON E.: 'Existence of generalized inverse of linear transformations over finite fields', *Finite fields Appl.*, 1998, **4**, (4), pp. 307–315

[51] BEN-ISRAEL A., GREVILLE T.N.E.: 'Generalized inverses: theory and applications' (Springer-Verlag, New York, 2003, 2nd edn.)

[52] CCSDS 131.0-B-1: 'TM synchronization and channel coding'. Recommendation for Space Data System Standards', CCSDS, Washington, DC, 2003, Blue Book, Issue 1

[53] SALOMON D.: 'Data compression: the complete reference' (Springer, 2000, 2nd edn.)

[54] HEO J.: 'Analysis of scaling soft information on low density parity check code', *Electron. Lett.*, **39**, (2), pp. 219–221

[55] SENDRIER N.: 'Finding the permutation between equivalent linear codes: the support splitting algorithm', *IEEE Trans. Inf. Theory*, 2000, **46**, (4), pp. 1193–1203

doi: 10.1049/iet-com:20080050