

On the Usage of LDPC Codes in the McEliece Cryptosystem

Marco Baldi

Dipartimento di Elettronica, Intelligenza Artificiale e Telecomunicazioni
Università Politecnica delle Marche

Ancona, Italy

Email: m.baldi@univpm.it

Abstract—In this paper, a new variant of the McEliece cryptosystem, based on Low-Density Parity-Check (LDPC) codes, is studied. Random-based techniques allow to design large families of LDPC codes with equivalent error correction capability; therefore, in principle, such codes can substitute Goppa codes, originally used by McEliece in his cryptosystem. Furthermore, Quasi-Cyclic (QC) LDPC codes can be adopted in order to reduce the key length, thus overcoming the main drawbacks of the original cryptosystem. Their usage, however, must be subject to cryptanalytic evaluation to ensure sufficient system robustness. The author proves that some widespread families of QC-LDPC codes, based on circulant permutation matrices, are inapplicable in this context, due to security issues, whilst other families of codes, based on the “difference families” approach, are not exposed to the same risk. However, another attack is presented that obliges to adopt very large codes in order to ensure a good level of security against intrusions.

I. INTRODUCTION

Since many years, error correcting codes have gained an important place in cryptography. In particular, just in 1978, McEliece proposed a public-key cryptosystem based on algebraic coding theory [1] that revealed to be very secure. The rationale of the McEliece algorithm, that adopts a generator matrix as the private key and a linear transformation of it as the public key, lies in the difficulty of decoding a large linear code with no visible structure, that in fact is known to be an NP complete problem [2]. As a matter of fact, the original McEliece cryptosystem is still unbroken, in the sense that no algorithm able to realize a total break in an acceptable time has been presented up to now. On the other hand, a vast body of literature exists on local deduction attacks, i.e., attacks finalized to find the plaintext of intercepted ciphertexts, without knowing the secret key. Despite the advances in the field, however, the work factors required for this kind of violation remain very high, and quite intractable in practice. Moreover, the system is two or three orders of magnitude faster than RSA, the latter being, probably, the most popular public key algorithm. A variant of the McEliece cryptosystem, due to Niederreiter [3], is even faster.

As a counterpart, however, the McEliece system also shows some drawbacks, that can justify the limited interest most cryptographers have devoted to it till today; among them, the large length of the key and the low transmission rate.

The current scenario of error correcting codes is dominated by schemes, like turbo codes or low density parity check

(LDPC) codes, whose decoding exploits iterative exchange of messages among constituent components, based on soft-input soft-output (SISO) modules. Thus, it seems interesting to investigate possible application of this kind of codes in the framework of the McEliece cryptosystem. The idea to adopt LDPC codes in the public-key cryptosystem was first explored in [4]; however, the main task of that paper was to demonstrate that the usage of LDPC codes in place of Goppa codes does not permit to reduce the key length.

In this paper, the system proposed in [4] is slightly modified and the possible application of Quasi-Cyclic (QC) LDPC codes in that framework is studied. These codes are easily encodable, and have been included as an option in the IEEE 802.16e standard for Mobile Wireless MAN [5], thus providing a valuable reference in this kind of applications. An algebraic technique is suggested to design a very large number of equivalent codes with fixed length and rate, which is the pre-requisite for their application in cryptosystems. Such codes should, in principle, overcome both the major drawbacks of the original McEliece cryptosystem, but their adoption must be subject to cryptanalytic evaluation. Previous attacks to the McEliece cryptosystem and new attacks tailored to LDPC codes are presented: the first new attack is a total break attack that can be conducted on codes based on circulant permutation matrices, whilst it does not affect those designed with the proposed approach; the second new attack is targeted to the dual of the secret code and represents a serious threat for the considered cryptosystem, to the point it can compromise its practical feasibility.

The details of the code design method are given in Section II, where QC codes are described in general terms and an overview of different design techniques is reported. In Section III, the McEliece system using LDPC codes is reviewed, and the role of its matrix components discussed. In Section IV cryptanalysis of the new system is carried out, considering both classic attacks to the McEliece cryptosystem and new attacks specifically targeted to LDPC codes.

II. QUASI-CYCLIC LDPC CODES

Quasi-Cyclic codes have been studied since many years [6], but they did not find a great success in the past because of their inherent decoding complexity in classic implementations. Nowadays, however, the encoding facility of Quasi-Cyclic

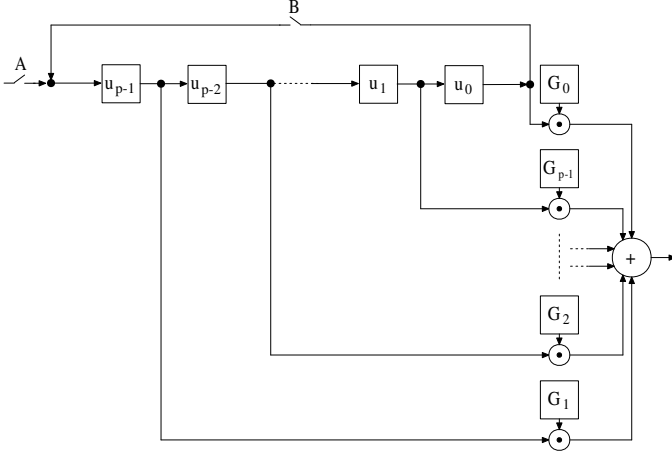


Figure 1. Encoder circuit for a QC code

codes can be combined with new efficient LDPC decoding techniques, thus yielding QC-LDPC codes, recently appeared even in telecommunication standards [5], [7].

The dimension k and the length n of a QC code are both multiple of a positive integer p , i.e. $k = p \cdot k_0$ and $n = p \cdot n_0$; the information vector $\mathbf{u} = [u_0, u_1, \dots, u_{k-1}]$ and the codeword vector $\mathbf{c} = [c_0, c_1, \dots, c_{n-1}]$ can be divided into p sub-vectors of size k_0 and n_0 , respectively, so that $\mathbf{u} = [\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_{p-1}]$ and $\mathbf{c} = [\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{p-1}]$.

The distinctive characteristic of QC codes is that every cyclic shift of n_0 positions of a codeword yields another codeword; since every cyclic shift of n_0 positions of a codeword is led by a cyclic shift of k_0 positions of the corresponding information word, it can be easily shown that Quasi-Cyclic codes are characterized by the following form of the generator matrix \mathbf{G} , where each block \mathbf{G}_i has size $k_0 \times n_0$:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \dots & \mathbf{G}_{p-1} \\ \mathbf{G}_{p-1} & \mathbf{G}_0 & \dots & \mathbf{G}_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_1 & \mathbf{G}_2 & \dots & \mathbf{G}_0 \end{bmatrix} \quad (1)$$

This leads to an efficient encoder implementation, consisting in a barrel shift register of size p , followed by a combinatorial network and an adder, as shown in Fig. 1.

It can be easily proved that the parity check matrix \mathbf{H} is also characterized by the same ‘‘circulant of blocks’’ form, in which each block \mathbf{H}_i has size $(n_0 - k_0) \times n_0 = r_0 \times n_0$. A row and column rearrangement can be applied that yields the alternative ‘‘block of circulants’’ form, here shown for the parity check matrix \mathbf{H} :

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{0,0}^c & \mathbf{H}_{0,1}^c & \dots & \mathbf{H}_{0,n_0-1}^c \\ \mathbf{H}_{1,0}^c & \mathbf{H}_{1,1}^c & \dots & \mathbf{H}_{1,n_0-1}^c \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}_{r_0-1,0}^c & \mathbf{H}_{r_0-1,1}^c & \dots & \mathbf{H}_{r_0-1,n_0-1}^c \end{bmatrix} \quad (2)$$

In this expression, each block $\mathbf{H}_{i,j}^c$ is a $p \times p$ circulant matrix:

$$\mathbf{H}_{i,j}^c = \begin{bmatrix} a_0^{i,j} & a_1^{i,j} & a_2^{i,j} & \dots & a_{p-1}^{i,j} \\ a_{p-1}^{i,j} & a_0^{i,j} & a_1^{i,j} & \dots & a_{p-2}^{i,j} \\ a_{p-2}^{i,j} & a_{p-1}^{i,j} & a_0^{i,j} & \dots & a_{p-3}^{i,j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_1^{i,j} & a_2^{i,j} & a_3^{i,j} & \dots & a_0^{i,j} \end{bmatrix} \quad (3)$$

and can be, therefore, associated to a polynomial $a^{i,j}(x) \in GF(2)[x] \bmod (x^p + 1)$, with maximum degree $p - 1$ and coefficients taken from the first row of $\mathbf{H}_{i,j}^c$:

$$a^{i,j}(x) = a_0^{i,j} + a_1^{i,j}x + a_2^{i,j}x^2 + a_3^{i,j}x^3 + \dots + a_{p-1}^{i,j}x^{p-1} \quad (4)$$

A. QC-LDPC codes based on Circulant Permutation Matrices

A well-known family of QC-LDPC codes has parity-check matrices in which each block $\mathbf{H}_{i,j}^c = \mathbf{P}_{i,j}$ is a circulant permutation matrix or the null matrix of size p ; circulant permutation matrices can be represented through the value of their first row shift $p_{i,j}$. Many authors have proposed code construction techniques based on this approach, like Tanner et al. in 2001 [8] and Fossorier in 2004 [9], and LDPC codes based on permutation matrices have been also included in the IEEE 802.16e standard [5], thus confirming their recognized error correction performance. Another reason why these codes have known good fortune is their implementation simplicity [10], [11].

The parity-check matrix of these codes can be represented through a ‘‘model’’ matrix \mathbf{H}_m , of size $r_0 \times n_0$, containing the shift values $p_{i,j}$ ($p_{i,j} = 0$ represents the identity matrix while, conventionally, $p_{i,j} = -1$ the null matrix). The code rate of such codes is $R = k_0/n_0$ and it can be varied arbitrarily through a suitable choice of r_0 and n_0 . On the other hand, the local girth length for these codes cannot exceed twelve [8], [9], and the imposition of a lower bound on the local girth length reflects on a lower bound on the code length [9].

The rows of a permutation matrix sum into the all-one vector, so these parity-check matrices cannot have full rank. Precisely, every parity-check matrix contains at least $r_0 - 1$ rows that are linearly dependent on the others, and the maximum rank is $r_0(p - 1) + 1$. A common solution to ensure the full rank is that of imposing the lower triangular (or quasi-lower triangular) form of the matrices, similarly to what done in the IEEE 802.16e standard [5].

B. QC-LDPC codes based on Difference Families

When designing a QC-LDPC code, the parity check matrix \mathbf{H} should have some properties that optimize the behavior of the belief propagation-based decoder. First of all, \mathbf{H} must be sparse, and this reflects on the maximum number of non-zero coefficients of each polynomial $a^{i,j}(x)$. Then, short length cycles must be avoided in the Tanner graph associated with the code.

The latter requirement can be ensured, as demonstrable through algebraic considerations, when \mathbf{H} is a single row of

circulants (*i.e.* $r_0 = 1$), and the corresponding code rate is $R = (n_0 - 1)/n_0$:

$$\mathbf{H} = [\mathbf{H}_0^c \quad \mathbf{H}_1^c \quad \cdots \quad \mathbf{H}_{n_0-1}^c] \quad (5)$$

Let $(G, +)$ be a finite group, and D a subset of G ; then, $\Delta D \equiv [x - y : x, y \in D, x \neq y]$ is the collection of all differences of distinct elements of D . Given a positive integer s , and a multi-set M , $sM \equiv \bigcup_{i=1}^s M$ is defined as s copies of M .

Let v be the order of the group G , μ and λ positive integers, with $v > \mu \geq 2$. A (v, μ, λ) -difference family (DF) is a collection $[D_1, \dots, D_t]$ of μ -subsets of G , called “base blocks”, such that

$$\bigcup_{i=1}^t \Delta D_i = \lambda (G \setminus \{0\}) \quad (6)$$

In other terms, every non-zero element of G appears exactly λ times as a difference of two elements from a base block. As already shown in the literature [12], difference families can be used to construct QC-LDPC codes. In particular, if we consider the difference family $[D_1, \dots, D_{n_0}]$, a code based on it has the following form for the polynomials of the circulant matrices $\{\mathbf{H}_0^c, \dots, \mathbf{H}_{n_0-1}^c\}$:

$$a^i(x) = \sum_{j=1}^{\mu} x^{d_{ij}} \quad , \quad i \in [1; n_0] \quad (7)$$

where d_{ij} is the j -th element of D_i whose dimension is μ . With this choice, the designed matrix \mathbf{H} is regular and all the elements in the difference family are used. It can be shown [12] that, by using difference families with $\lambda = 1$ in construction (7), the resulting code has a Tanner graph free of 4-length cycles.

Some theorems ensure the existence of difference families with $\lambda = 1$, but they apply only for particular values of the group order, so putting heavy constraints on the code length. Such constraints can be overcome by relaxing part of the hypotheses of these theorems and then refining the outputs through simple computer searches, in the so-called “Pseudo Difference Families” approach [13]; other authors have proposed an alternative technique based on “Extended Difference Families”, that ensures great flexibility in the code length [14]. Finally, a multi-set with the properties of a difference family can be constructed by (constrained) random choice of its elements; let us call this a “Random Difference Family” or RDF.

C. Equivalent codes based on Difference Families

A first requirement when using an error correcting code in a cryptosystem concerns the possibility to choose it at random among a very large class of equivalent codes. This way, an opponent, even aware of the code parameters (*i.e.* length and rate), neither knows (obviously) the private key nor is able to obtain it through a brute force attack.

When considering LDPC codes, their equivalence needs to be verified under message passing decoding, whose performance does not depend only on the weight spectrum. Generally speaking, two codes exhibit almost identical performance when they have equal (or very similar): i) code length and rate, ii) parity check matrix density, iii) nodes degree distributions and iv) girth length distribution in the Tanner graph associated with the code.

Let us adopt a family of codes with fixed length n and parity-check matrices in the form (5), so property i) is ensured. An integer $d_v > 2$ is then set as the column weight in matrix \mathbf{H} , so that property ii) is verified too. d_v integers are then chosen in such a way to ensure the difference family properties, and they are used to construct, through Eq. (7), a parity check matrix \mathbf{H} in the form (5), with rate $R = (n_0 - 1)/n_0$. Since circulant matrices are regular, the resulting parity check matrix is regular in both row and column weights; so all codes have the same nodes degree distributions and property iii) is verified. Finally, when a circulant matrix has row (column) weight greater than 2, and it does not induce 4-length cycles, it can be shown it corresponds to a Tanner graph with constant local girth length equal to 6, so property iv) is satisfied as well.

In a recent paper [15] the author has proposed, for the same objective of McEliece cryptosystem implementation, to design an EDF larger than needed, through a randomized version of the algorithm reported in [14], and then select randomly a subset of each block in order to have a QC-LDPC code. Each generated EDF hence produces as many different codes as the possible different random choices of its elements; in order to ensure large cardinality of the code family, however, this approach can require large code length.

Therefore, it is preferable to adopt Random Difference Families (RDFs), that do not derive from EDFs, but rather are constructed by direct (constrained) random selection of their elements. The constraints imposed serve to ensure the difference family character of each set, namely that the designed codes do not have 4-length cycles. A lower bound on the cardinality $C\{\text{RDF}(n_0, d_v, p)\}$ of a set of RDFs with fixed parameters n_0 , d_v and p can be evaluated, through probabilistic arguments, as follows [16]:

$$C\{\text{RDF}(n_0, d_v, p)\} \geq \frac{1}{p} \binom{p}{d_v} \prod_{l=0}^{n_0-1} \prod_{j=1}^{d_v-1} \frac{p}{p-j} + \frac{j \left[2 - p \bmod 2 + (j-1)^2 / 2 + l \cdot d_v \cdot (d_v - 1) \right]}{p-j} \quad (8)$$

According with this expression, very high values of $C\{\text{RDF}(n_0, d_v, p)\}$ can be obtained, even for (relatively) short codes. As an example, by assuming $n = 1060$, $n_0 = 4$ (which implies rate $R = 3/4$ and $p = 265$) and $d_v = 5$, we have $C\{\text{RDF}(4, 5, 210)\} \simeq 2^{111}$. From the cryptanalysis point of view, however, it should be observed that an attack could be made on each $\mathbf{H}_{i,j}^c$ block, which is equivalent to say that the maximum number of trials for an eavesdropper results

from setting $n_0 = 1$ in expression (8). In order to preserve high cardinalities, longer codes should be therefore considered, that however would remain feasible for application. As an example, by setting, at a parity of n_0 , $n = 8000$ (that implies $p = 2000$) and $d_v = 13$, we have $C \{RDF(1, 13, 2000)\} \simeq 2^{97}$, high enough to discourage a brute force attack on a single submatrix.

III. MCELIECE CRYPTOSYSTEM WITH LDPC CODES

A. System description

A possible implementation of the McEliece system using LDPC codes is as follows [4]. Bob, that must receive a message from Alice, chooses a parity check matrix \mathbf{H} among a family of a given class of LDPC codes. Let us suppose to know that the chosen code, like any other in the same class, is able to correct t errors with high probability, under belief propagation decoding. Bob also chooses an $r \times r$ non-singular dense circulant “transformation” matrix, \mathbf{T} , and obtains the new matrix $\mathbf{H}' = \mathbf{T} \cdot \mathbf{H}$, that, obviously, has the same null space of \mathbf{H} .

Bob then computes a generator matrix, \mathbf{G} , corresponding to \mathbf{H}' , in row reduced echelon form and makes it available in the public directory: \mathbf{G} is Bob’s public key and it is completely described by a single row of its non-systematic part, that is a k -bit vector. On the contrary, \mathbf{H} and \mathbf{T} form the private (or secret) key, that is owned by Bob only. The system requires also a $k \times k$ non-singular “scrambling” matrix \mathbf{S} , that is suitably chosen and publicly available (it can be even embedded in the algorithm implementation). Also matrix \mathbf{S} has the “circulants block” form, and its role is to cause propagation of residual errors at the eavesdropper’s receiver, leaving the opponent in the most uncertain condition (that is equivalent to guess the plaintext at random). For this purpose, it must be sufficiently dense in its turn.

When Alice wants to send an encrypted message to Bob, she fetches \mathbf{G} from the public directory and calculates $\mathbf{G}' = \mathbf{S}^{-1} \cdot \mathbf{G}$. Then she divides her message into k -bit blocks and encrypts each block \mathbf{u} as follows:

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G}' + \mathbf{e} \quad (9)$$

where \mathbf{x} is the encrypted version of \mathbf{u} and \mathbf{e} is a random vector of t intentional errors. The choice of t will be discussed next.

At the receiver side, Bob uses its private key for decoding. In the ideal case of a channel that does not introduce additional errors, by a suitable choice of t , all errors can be corrected with high probability (in the extremely rare case of an LDPC decoder failure, message resend is request, as discussed in Subsection IV-A). Belief propagation decoding, however, works only on sparse Tanner graphs, free of short-length cycles; therefore, in order to exploit the actual correction capability, the knowledge of the sparse parity-check matrix \mathbf{H} is essential.

By applying the decoding algorithm on the ciphertext \mathbf{x} , Bob can derive $\mathbf{u} \cdot \mathbf{G}' = \mathbf{u} \cdot (\mathbf{S}^{-1} \cdot \mathbf{G})$. On the other hand, as \mathbf{G} is in row reduced echelon form, the first k coordinates of

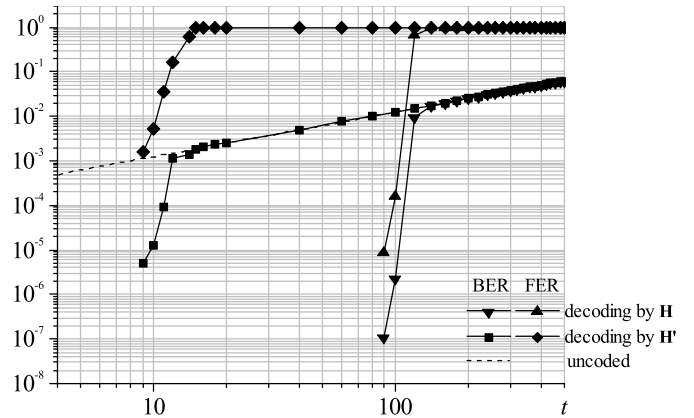


Figure 2. Performance attainable over the “McEliece channel” by \mathbf{H} and by \mathbf{H}' .

this product reveal directly $\mathbf{u} \cdot \mathbf{S}^{-1}$, and right multiplication by \mathbf{S} permits to extract the plaintext \mathbf{u} as desired.

The eavesdropper Eve, that wishes to intercept the message from Alice to Bob and is not able to derive matrix \mathbf{H} from the knowledge of \mathbf{G} , is, as expected, in a much less favorable position. Even if \mathbf{H}' can be derived from \mathbf{G} , it is made unsuitable for LDPC decoding through the action of matrix \mathbf{T} . In addition, \mathbf{H}' is dense and this implies, for Eve, large decoding complexity.

B. Choice of matrix S

The residual errors after decoding are “propagated” by the subsequent product by \mathbf{S} , so that, at the output of the eavesdropper’s decoder, not only the FER is equal to 1 (which means that all decoded sequences are erred) but also the BER is practically equal to $1/2$ (that is the most uncertain condition for an opponent). Similarly to \mathbf{T} , also \mathbf{S} has to be rather dense, for doing, at the best, this error propagating action.

The results of a numerical transmission simulation over the “McEliece channel” (a channel that introduces t errors on each frame) is shown in Fig. 2, where the error correction performance achieved by the private matrix \mathbf{H} and by a public matrix \mathbf{H}' is shown. The considered code has parameters $n = 8000$, $k = 6000$ ($p = 2000$) and $d_v = 13$. For further evidence, a sparse \mathbf{T} matrix (with same column weight as \mathbf{H}) has been used to obtain \mathbf{H}' ; employing denser \mathbf{T} matrices (as those requested for security issues) the performance of \mathbf{H}' would be even worse.

As mentioned, the value of t must be upper bounded by the code error correction capability. In the considered example, it is possible to foresee that authorized users have negligible error probability for $t \leq 40$, while, by assuming $t = 40$, unauthorized users have useless decoders (they achieve same BER as the uncoded transmission).

Despite the effect of matrix \mathbf{T} , the error correction performance achievable by unauthorized users is still good. Fig. 2 shows that, in our example, the uncoded transmission for $t = 40$ achieves BER of the order of $5 \cdot 10^{-3}$, which is quite unacceptable from the security viewpoint. A proper security

level is restored by the action of the scrambling matrix, that causes “propagation” of the residual errors. This can be explained in the following terms.

If we consider the information part of the vector decoded by an opponent, $\tilde{\mathbf{u}}$, it can be expressed as $\tilde{\mathbf{u}} = \mathbf{u} \cdot \mathbf{S}^{-1} + \tilde{\mathbf{e}}$, where $\tilde{\mathbf{e}}$ is corresponding part of the residual errors vector. After the descrambling process, the decoded message is $\hat{\mathbf{u}} = \tilde{\mathbf{u}} \cdot \mathbf{S} = \mathbf{u} + \tilde{\mathbf{e}} \cdot \mathbf{S}$; therefore the scrambling matrix \mathbf{S} operates directly on the residual errors. Really, the extent of the error propagation effect can be predicted through simple combinatorial arguments, under the hypothesis that \mathbf{S} is randomly generated. At the same time, this prediction permits to design the features of matrix \mathbf{S} (its density, in particular, for a given size) that are compatible with the achievement of a satisfactory security level, according with the procedure described below.

Let η be the Hamming weight of vector $\tilde{\mathbf{e}}$ and v the unknown value of the row (column) weight of matrix \mathbf{S} . Moreover, let us suppose that the value of k is fixed. Under the hypothesis that \mathbf{S} is randomly generated, the probability that j symbols 1 in $\tilde{\mathbf{e}}$ coincide with as many symbols 1 in a generic column of \mathbf{S} can be easily calculated as follows:

$$P_j(v) = \frac{\binom{v}{j} \binom{k-v}{\eta-j}}{\binom{k}{\eta}}$$

Now, the i -th element of vector $\tilde{\mathbf{e}} \cdot \mathbf{S}$ is equal to 1 (which means that a residual error is present) any time the number of coincidences j is odd. Consequently, the conditional probability that the i -th bit of $\hat{\mathbf{u}}$ is in error, when $\tilde{\mathbf{e}}$ has weight η and the row (column) weight of \mathbf{S} is v , equals:

$$P(\text{err}|\eta, v) = \sum_{m=0}^{\lfloor \frac{v-1}{2} \rfloor} P_{2m+1}(v) \quad (10)$$

Finally, the probability that the i -th bit of $\hat{\mathbf{u}}$ is in error, for the same value v , can be obtained as:

$$P(\text{err}|v) = \sum_{\eta=1}^k P(\text{err}|\eta, v) P_{\tilde{\mathbf{e}}}(\eta) \quad (11)$$

where $P_{\tilde{\mathbf{e}}}(\eta)$ represents the probability that vector $\tilde{\mathbf{e}}$ has weight η , and, for the McEliece cryptosystem, can be estimated as follows:

$$P_{\tilde{\mathbf{e}}}(\eta) = \frac{\binom{t}{\eta} \binom{n-t}{k-\eta}}{\binom{n}{k}} \quad (12)$$

Expression (12) can be used, together with expression (10), in expression (11). This way, it can be calculated, as an example, that, for $n = 8000$, $k = 6000$ and $t = 30$, an \mathbf{S} matrix with density of about 18% suffices to ensure maximum error “propagation” action (that consists in obtaining BER equal to 1/2). On the other hand, it is expected that the scrambling action has impact also for the authorized user (Bob). In other words, a penalty in the error correction performance must appear, that is compensated by the possibility to make the system secure against unauthorized intrusions.

IV. SYSTEM CRYPTANALYSIS

Cryptanalysis is carried out considering both attacks already developed for the original McEliece cryptosystems and new attacks targeted to the proposed instance of it.

A. Classic Attacks

The first kind of known attacks are brute force attacks. As already shown, enumeration of the code set is too demanding; therefore a brute force attack on \mathbf{H} (or a square sub-block of \mathbf{H}) should be excluded. Even the number of randomly chosen (dense) \mathbf{T} matrices is extremely high. Other kinds of brute force attacks (e.g. trying to decode the ciphertext, even using coset leaders) are unfeasible as well, like in the original McEliece cryptosystem.

A low complexity attack, in the form of a local deduction, yet mentioned in [1], has been further investigated and improved in the subsequent literature. Lee and Brickell, in [17], propose a generalization of such attack, characterized by reduced complexity. The work factor of Lee and Brickell’s algorithm can be evaluated as follows:

$$W_j = T_j (\alpha k^3 + N_j \beta k) \quad (13)$$

where $T_j = \left[\sum_{i=0}^j \binom{t}{i} \binom{n-t}{k-i} / \binom{n}{k} \right]^{-1}$, $N_j = \sum_{i=0}^j \binom{k}{i}$ and α , β and j are integers ≥ 1 . Assuming $\alpha = \beta = 1$ (the most favorable condition for an opponent), $n = 8000$ and $k = 6000$, we have that W_j is minimum for $j = 2$ and, for $t = 40$, $W_2 \simeq 2^{105.8}$, that would be high enough to discourage the attack.

Berson [18] proved that attacks based on Information Set Decoding become very easy when the same message is encrypted more than once (with different error vectors), or in the case of messages with a known linear relationship among them. When considering LDPC codes, it is not easy to determine the decoding radius under belief propagation; therefore, for a fixed value of t , Bob’s LDPC decoder may be occasionally unable to correct all the errors. In this case, that however occurs with extremely low probability, the parity check fails, and the uncorrected errors are detected. So, message resending could be necessary. It should be noted, however, that the same circumstance can occur in the original McEliece cryptosystem, though originated by different causes.

Berson-like attacks, however, can be avoided through a simple modification of the cryptosystem [19]: for example, the encryption map can be modified as follows: $\mathbf{x} = [\mathbf{u} + h(\mathbf{e})] \cdot \mathbf{G}' + \mathbf{e}$, where h is a one-way hash function with \mathbf{e} as input and a k -bit output. Its contribution must be obviously removed by Bob, after successful decoding: $\mathbf{u} = [\mathbf{u} + h(\mathbf{e})] + h(\mathbf{e})$.

Another attack can be derived considering that the problem of finding \mathbf{e} translates into that of finding the minimum weight codeword of the $(n, k+1)$ linear block code generated by $\mathbf{G}'' = \begin{bmatrix} \mathbf{G}' \\ \mathbf{x} \end{bmatrix}$. The problem of finding the minimum weight codeword is mostly unsolved for the case of LDPC codes. Actually, some algorithms have been studied [20], but they rarely succeed on long codes, like those adopted here.

Moreover, matrix \mathbf{H}' is not suitable for iterative decoding, while these algorithms exploit belief propagation.

An adaptation of the probabilistic algorithm originally proposed by Stern [21], that does not rely on iterative decoding, has been applied to LDPC codes by Hiroto et al. [22]. Even if this approach seems to outperform that based on iterative decoding, it is hard to estimate the minimum distance using Stern's algorithm for code length $n \geq 2048$, as the authors themselves acknowledge. For $n = 4096$ and $k = 2048$ (that are smaller than those here considered), the work factor reaches $2^{98.3}$. Even considering the quasi-cyclic property, such values of the work factor should discourage the attack.

B. Attacks to LDPC Codes

1) *Density Reduction Attacks*: These attacks, already conjectured in [4], are specifically targeted to LDPC codes.

Let h_i be the i -th row of matrix \mathbf{H} and h'_j the j -th row of matrix $\mathbf{H}' = \mathbf{T} \cdot \mathbf{H}$, and let $(GF_2^n, +, \times)$ be the vector space of all the possible binary n -tuples with the operations of addition (*i.e.* the logical "XOR") and multiplication (*i.e.* the logical "AND"). Let us define "orthogonality" in the following sense: two binary vectors u and v are orthogonal, *i.e.* $u \perp v$, iff $u \times v = 0$. From the cryptosystem description, it follows that: $h'_j = h_{i_1} + h_{i_2} + \dots + h_{i_z}$ where z represents the Hamming weight of each row (column) of \mathbf{T} .

Let us suppose that many h_i are mutually orthogonal, due to the sparsity of matrix \mathbf{H} . Let $h'_{j^a} = h_{i_1^a} + h_{i_2^a} + \dots + h_{i_z^a}$ and $h'_{j^b} = h_{i_1^b} + h_{i_2^b} + \dots + h_{i_z^b}$ be two distinct rows of \mathbf{H}' and $h_{i_1^a} = h_{i_1^b} = h_{i_1}$ [that happens when \mathbf{T} has two non-zero entries in the same column (i_1), at rows j^a and j^b .] In this case, it may happen that: $h'_{j^a} \times h'_{j^b} = h_{i_1}$ (that occurs, for example, when $h_{i_1} \perp h_{i_2^b}, \dots, h_{i_1} \perp h_{i_2^a}, \dots, h_{i_1} \perp h_{i_1^b}, \dots, h_{i_1} \perp h_{i_1^a}$). Therefore, a row of \mathbf{H} could be derived as the product of two rows of \mathbf{H}' . At this point, if the code is quasi-cyclic with the considered form, its whole parity-check matrix can be obtained, due to the fact that the other rows of \mathbf{H} are simply block-wise circular shifted versions of the one obtained through the attack.

Even when the analysis of all possible couples of rows of \mathbf{H}' does not reveal a row of \mathbf{H} , it may produce a new matrix, \mathbf{H}'' , sparser than \mathbf{H}' , able to allow efficient LDPC decoding. Alternatively, the attack can be iterated on \mathbf{H}'' and it can succeed after a number of iterations > 1 ; in general, the attack requires $\rho - 1$ iterations when not less than ρ rows of \mathbf{H}' have in common a single row of \mathbf{H} . This attack procedure can be even applied on a single circulant block of \mathbf{H}' , say \mathbf{H}'_i , to derive its corresponding block \mathbf{H}_i of \mathbf{H} , from which $\mathbf{T} = \mathbf{H}'_i \cdot \mathbf{H}_i^{-1}$ can be obtained.

The author has verified elsewhere [16] that the attack can be avoided through a proper selection of matrix \mathbf{T} , but this approach forces constraints on the code parameters. In this work, following [4], it is proposed instead to resort only to matrix \mathbf{T} density.

Let us suppose that the attack is carried out on the single block \mathbf{H}'_i (it can be easily generalized for the whole \mathbf{H}'). The first iteration of the attack, for the considered case of \mathbf{H}'_i

circulant, is equivalent to calculate the periodic autocorrelation of the first row of \mathbf{H}'_i . When \mathbf{H}'_i is sparse (*i.e.* \mathbf{T} is sparse) the autocorrelation is everywhere null (or very small), except for a limited number of peaks that reveal the couple of rows of \mathbf{H}'_i able to give information on the structure of \mathbf{H}_i . On the contrary, when \mathbf{H}'_i is dense (suppose with one half of symbols 1), the autocorrelation is always high, and no information is available for the opponent. In this case, the eavesdropper Eve is in the same condition as to guess at random. The relevant point is that to have a dense matrix \mathbf{H}' , when the proposed system adopts QC-LDPC codes based on RDFs, does not affect the public key length: matrix \mathbf{G} remains described completely by a column of its non-systematic part.

2) *Attack to Circulant Permutation Matrices-based QC-LDPC codes*: In the previous subsection it has been explained how the assumption of a matrix \mathbf{T} sufficiently dense may guarantee inapplicability of an attack that aims at finding single rows of \mathbf{H} . In this subsection it is shown that, independently of the \mathbf{T} density, QC-LDPC codes having the form described in Section II-A cannot be used in the considered cryptosystem, since a total-break attack is possible able to recover the private key with high probability and low complexity.

Let the private key \mathbf{H} be formed by $r_0 \times n_0$ circulant permutation or null matrices $\mathbf{P}_{i,j}$ of size p and have lower triangular form, to ensure full rank. For the sake of clarity, let us consider a simple example with $r_0 = 3$ and $n_0 = 6$, and where all the blocks $\mathbf{P}_{i,j}$ are non-null:

$$\mathbf{H} = \begin{bmatrix} \mathbf{P}_{1,1} & \mathbf{P}_{1,2} & \mathbf{P}_{1,3} & \mathbf{P}_{1,4} & 0 & 0 \\ \mathbf{P}_{2,1} & \mathbf{P}_{2,2} & \mathbf{P}_{2,3} & \mathbf{P}_{2,4} & \mathbf{P}_{2,5} & 0 \\ \mathbf{P}_{3,1} & \mathbf{P}_{3,2} & \mathbf{P}_{3,3} & \mathbf{P}_{3,4} & \mathbf{P}_{3,5} & \mathbf{P}_{3,6} \end{bmatrix} \quad (14)$$

Let \mathbf{T} have the generic form of $r_0 \times r_0$ square circulant dense blocks $\mathbf{T}_{i,j}$, each of size p :

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{1,1} & \mathbf{T}_{1,2} & \mathbf{T}_{1,3} \\ \mathbf{T}_{2,1} & \mathbf{T}_{2,2} & \mathbf{T}_{2,3} \\ \mathbf{T}_{3,1} & \mathbf{T}_{3,2} & \mathbf{T}_{3,3} \end{bmatrix} \quad (15)$$

The public key is obtained as $\mathbf{H}' = \mathbf{T} \cdot \mathbf{H}$. Although the exact knowledge of the private key would certainly allow correct decoding, for the eavesdropper it suffices to find a couple of matrices $(\mathbf{T}_d, \mathbf{H}_d)$, with the same dimensions of (\mathbf{T}, \mathbf{H}) , such that $\mathbf{H}' = \mathbf{T}_d \cdot \mathbf{H}_d$, and \mathbf{H}_d is sparse enough to allow efficient belief propagation decoding. This can be accomplished considering that, given an invertible square matrix \mathbf{Z} of size $r = p \cdot r_0$, the following relationship holds:

$$\mathbf{H}' = \mathbf{T} \cdot \mathbf{H} = \mathbf{T} \cdot \mathbf{Z} \cdot \mathbf{Z}^{-1} \cdot \mathbf{H} = \mathbf{T}_d \cdot \mathbf{H}_d \quad (16)$$

where $\mathbf{T}_d = \mathbf{T} \cdot \mathbf{Z}$ and $\mathbf{H}_d = \mathbf{Z}^{-1} \cdot \mathbf{H}$. If we separate \mathbf{H} in its left ($r \times k$) and right ($r \times r$) parts, $\mathbf{H} = [\mathbf{H}_l | \mathbf{H}_r]$, a particular choice of \mathbf{Z} coincides with the lower triangular part of \mathbf{H} , *i.e.*, for the considered example:

$$\mathbf{Z} = \mathbf{H}_r = \begin{bmatrix} \mathbf{P}_{1,4} & 0 & 0 \\ \mathbf{P}_{2,4} & \mathbf{P}_{2,5} & 0 \\ \mathbf{P}_{3,4} & \mathbf{P}_{3,5} & \mathbf{P}_{3,6} \end{bmatrix} \quad (17)$$

With this choice of \mathbf{Z} , \mathbf{H}_d assumes a particular form:

$$\mathbf{H}_d = [\mathbf{H}_{d1}|\mathbf{H}_{dr}] = \mathbf{H}_r^{-1} \cdot [\mathbf{H}_1|\mathbf{H}_r] = [\mathbf{H}_r^{-1} \cdot \mathbf{H}_1|\mathbf{I}] \quad (18)$$

where the right part of \mathbf{H}_d is an identity matrix of size $r = p \cdot r_0$. From this, it follows that:

$$\mathbf{H}' = \mathbf{T}_d \cdot \mathbf{H}_d = [\mathbf{T}_d \cdot \mathbf{H}_{d1}|\mathbf{T}_d] \quad (19)$$

Eq. (19) is the starting point for the eavesdropper: looking at \mathbf{H}' , she immediately knows \mathbf{T}_d and, hence \mathbf{H}_d , both corresponding to the choice of \mathbf{Z} expressed by Eq. (17).

Moreover, it can be proved that \mathbf{H}_d , so found, can be sparse enough to allow efficient belief propagation decoding. In fact, because of the \mathbf{H}_d definition, its sparsity depends on that of \mathbf{Z}^{-1} . Actually, for the considered example, it is:

$$\mathbf{Z}^{-1} = \mathbf{H}_r^{-1} = \begin{bmatrix} \mathbf{Z}_{1,1} & 0 & 0 \\ \mathbf{Z}_{2,1} & \mathbf{Z}_{2,2} & 0 \\ \mathbf{Z}_{3,1} & \mathbf{Z}_{3,2} & \mathbf{Z}_{3,3} \end{bmatrix} \quad (20)$$

where

$$\begin{cases} \mathbf{Z}_{1,1} = \mathbf{P}_{1,4}^T \\ \mathbf{Z}_{2,2} = \mathbf{P}_{2,5}^T \\ \mathbf{Z}_{3,3} = \mathbf{P}_{3,6}^T \\ \mathbf{Z}_{2,1} = \mathbf{P}_{2,5}^T \mathbf{P}_{2,4} \mathbf{P}_{1,4}^T \\ \mathbf{Z}_{3,2} = \mathbf{P}_{3,6}^T \mathbf{P}_{3,5} \mathbf{P}_{2,5}^T \\ \mathbf{Z}_{3,1} = \mathbf{P}_{3,6}^T \mathbf{P}_{3,4} \mathbf{P}_{1,4}^T + \mathbf{P}_{3,6}^T \mathbf{P}_{3,5} \mathbf{P}_{2,5}^T \mathbf{P}_{2,4} \mathbf{P}_{1,4}^T \end{cases} \quad (21)$$

and the property of permutation matrices to have inversion coincident with transposition has been used too.

It follows that the main diagonal of \mathbf{Z}^{-1} contains permutation matrices; the underlying diagonal contains products of permutation matrices, *i.e.* permutation matrices again; the following one contains sums of permutation matrices. The same analysis holds for an arbitrary r_0 : the blocks $\mathbf{Z}_{i+j,1+j}$, $i \in [2; r_0]$, $j \in [0; r_0 - i]$ have column (row) weight 2^{i-2} . It follows that, when r_0 is small, \mathbf{Z}^{-1} is sparse and, consequently, \mathbf{H}_d is sparse as well; so it could be used by Eve for efficient decoding.

Furthermore, the attack can continue and aim at obtaining another matrix, \mathbf{H}_b , that has the same density of \mathbf{H} and, therefore, produces a total break of the cryptosystem. This new matrix corresponds to another choice of \mathbf{Z} , namely $\mathbf{Z} = \mathbf{Z}^*$, that, for the considered example, has the following form:

$$\mathbf{Z}^* = \begin{bmatrix} \mathbf{P}_{1,4} & 0 & 0 \\ 0 & \mathbf{P}_{2,5} & 0 \\ 0 & 0 & \mathbf{P}_{3,6} \end{bmatrix} \quad (22)$$

For $\mathbf{Z} = \mathbf{Z}^*$, \mathbf{H}_b (that is in row reduced echelon form) has the same density of \mathbf{H} , since each of its rows is a permuted version of the corresponding row of \mathbf{H} . An attack that aims at finding \mathbf{H}_b can be conceived by analyzing the structure of \mathbf{H}_d in the form (18), with \mathbf{Z}^{-1} expressed by Eq. (20) and (21). We notice that the first row of \mathbf{H}_d equals that of \mathbf{H} multiplied by $\mathbf{P}_{1,4}^T$, so it corresponds to the first row of \mathbf{H}_b . The second

row of \mathbf{H}_d equals that of \mathbf{H} multiplied by $\mathbf{P}_{2,5}^T$ plus a shifted version of the first row of \mathbf{H}_b . The third row of \mathbf{H}_d equals that of \mathbf{H} multiplied by $\mathbf{P}_{3,6}^T$ plus two shifted versions of the first row of \mathbf{H}_b and a shifted version of the second row of \mathbf{H}_b . Therefore, a recursive attack can be conceived.

When sparse matrices are added, it is highly probable that their symbols “1” do not overlap. For this reason, in a sum of rows of \mathbf{H}_b , the contributions of shifted versions of known rows can be isolated through a correlation operation and, therefore, eliminated. This permits to deduce each row of \mathbf{H}_b from the previous ones and, this way, obtain the entire \mathbf{H}_b . Obviously, the hypothesis of non-overlapping elements is most likely verified when the blocks of \mathbf{H} are sparse and r_0 is not too high. For example, it has been verified that matrices with $r_0 = 6$ and $p = 40$ (*i.e.* density of the blocks 0.025) are highly exposed to total break. This is not a trivial case; for example, the codes included in the IEEE 802.16e standard [5] have $p \in [24; 96]$ and $r_0 = 6$ when the code rate is 3/4.

An attack of this kind is addressed to LDPC codes based on circulant permutation matrices, whilst it is not applicable to LDPC codes based on difference families (described in Section II-B). For this reason, the latter appear more secure, at least at the present stage of cryptanalysis.

3) *Attacks to the Dual Code*: The most dangerous attack for every instance of the McEliece cryptosystem based on LDPC codes rises from the fact that an opponent knows the dual of the secret code contains very low weight codewords and can directly search for them, thus recovering \mathbf{H} .

The dual of the secret code can be generated by \mathbf{H} ; therefore it has at least $A_{d_c} \geq r$ codewords with weight d_c . Each of them completely describes \mathbf{H} and, if known, allows the opponent to break the system by gathering the private key. From a cryptographic point of view, A_{d_c} should be known in order to precisely evaluate the work factor of the attack, but this is not, in general, a simple task. However, it can be considered that it is $d_c \ll n$ and that sparse vectors most likely sum into vectors of higher weight. Therefore, it will be considered $A_{d_c} = r$ in the following.

The best known probabilistic algorithm for finding low weight codewords is due to Stern [21], and it has been recently applied to LDPC codes by Hiroto et al. [22]. The algorithm, that exploits an iterative procedure, works on the parity-check matrix of a code and has two parameters, p and l , that represent the number of matrix columns and rows considered at each iteration, respectively. Optimal values for p and l can be derived considering their influence on the total number of binary operations needed for finding a codeword of given weight. If we suppose that the algorithm is performed on the dual of the secret code, with length n and dimension k_d (*i.e.* redundancy $r_d = n - k_d$), the probability of finding, in one iteration, a codeword with weight w , supposed that it is unique, is $P_w = P_1 \cdot P_2 \cdot P_3$, with:

$$\begin{cases} P_1 = \binom{w}{p} \binom{n-w}{k_d/2-p} / \binom{n}{k_d/2} \\ P_2 = \binom{w-p}{p} \binom{n-k_d/2-w+p}{k_d/2-p} / \binom{n-k_d/2}{k_d/2} \\ P_3 = \binom{n-k_d-w+2p}{l} / \binom{n-k_d}{l} \end{cases}$$

If the code contains A_w codewords with weight w , it is $P_{w,A_w} \leq A_w P_w$; therefore, the average number of iterations needed in order to find one of them is $m \geq P_{w,A_w}^{-1}$. It can be considered that each iteration of the algorithm requires

$$N = \frac{r_d^3}{2} + k_d r_d^2 + 2pl \binom{k_d/2}{p} + \frac{2pr_d \binom{k_d/2}{p}^2}{2^l}$$

binary operations, so the total work factor can be estimated as $W = mN$.

If we consider the following choice of the system parameters: $n = 8000$, $k_d = 2000$, $w = d_c = n_0 \cdot d_v = 52$, $A_w = 2000$, the minimum work factor, that corresponds to $(p, l) = (3, 38)$, is $2^{35.65}$. It is evident that such system would be highly exposed to a total break.

This attack is particularly insidious since the work factor of Stern's algorithm mainly depends on the relative weight searched and decreases with the code rate (it is desirable, for the dual code, to have rate as low as possible, *i.e.* highest rate for the private and the public code). In order to increase the work factor, denser parity-check matrices and lower code rate should be adopted. On the other hand, however, such matrices must be sparse enough to ensure the absence of 4-length cycles and allow efficient belief propagation decoding. This means that it is possible to obtain high work factors only by employing relatively large codes. For example, by choosing $n = 84000$, $r = k_d = 28000$, $n_0 = 3$ ($R = 2/3$) and $d_v = 41$ ($d_c = 123$) it is $W = 2^{81.3}$ (minimal for $p = 3$, $l = 54$), that ensures satisfactory system robustness. With this choice of the parameters, the number of equivalent codes is still high¹. However, the complexity of such a system is high, and it could be very hard to implement. Alternatively, the cryptosystem should be modified in order not to expose the secret code, thus preventing the attack to be even attempted. Further work is in progress in this direction.

V. CONCLUSIONS

An instance of the McEliece cryptosystem based on QC-LDPC codes has been studied. Such codes, in principle, could be able to overcome the main drawbacks of the original McEliece cryptosystem, that are large keys and low transmission rate.

The new system has been cryptanalyzed both considering classic attacks and introducing new threats to its security. It has been shown that some structured configurations of the parity-check matrix, like those based on circulant permutation matrices, cannot be used in this system as highly vulnerable to total breaks. The attack proposed for demonstrating this conclusion is not applicable to different design methods, like that based on the use of difference families. However, another attack has been presented, targeted to the dual of the secret code, able to seriously threaten the system security. This attack forces to adopt larger codes, even if such choice can jeopardize the system feasibility.

¹The lower bound given in Subsection II-C becomes too loose, but such number can be estimated through different arguments.

ACKNOWLEDGMENTS

The author is greatly indebted to Professor F. Chiaraluce for his continued interest and his precious insights during the progress of this research.

REFERENCES

- [1] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory." *DSN Progress Report*, pp. 114–116, 1978.
- [2] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. 24, pp. 384–386, May 1978.
- [3] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Probl. Contr. and Inform. Theory*, vol. 15, pp. 159–166, 1986.
- [4] C. Monico, J. Rosenthal, and A. Shokrollahi, "Using low density parity check codes in the McEliece cryptosystem," in *Proc. IEEE ISIT 2000*, Sorrento, Italy, Jun. 2000, p. 215.
- [5] 802.16e (2005), *IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems - Amendment for Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE Std., Dec. 2005.
- [6] R. Townsend and E. J. Weldon, "Self-orthogonal quasi-cyclic codes," *IEEE Trans. Inform. Theory*, vol. 13, pp. 183–195, Apr. 1967.
- [7] IEEE P802.11, *Wireless LANs WWiSE Proposal: High throughput extension to the 802.11 Standard*, IEEE Std. IEEE 11-04-0886-00-000n, Aug. 2004.
- [8] R. Tanner, D. Sridhara, and T. Fuja, "A class of group-structured LDPC codes," in *Proc. ISCTA 2001*, Ambleside, UK, Jul. 2001.
- [9] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. Inform. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.
- [10] D. Hocevar, "LDPC code construction with flexible hardware implementation," in *Proc. IEEE ICC 2003*, vol. 4, Anchorage, Alaska, May 2003, pp. 2708–2712.
- [11] —, "Efficient encoding for a family of quasi-cyclic LDPC codes," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '03)*, vol. 7, 2003, pp. 3996–4000.
- [12] S. Johnson and S. Weller, "A family of irregular LDPC codes with low encoding complexity," *IEEE Commun. Lett.*, vol. 7, pp. 79–81, Feb. 2003.
- [13] M. Baldi and F. Chiaraluce, "New quasi cyclic low density parity check codes based on difference families," in *Proc. 8th Int. Symp. Commun. Theory and Appl., ISCTA 05*, Ambleside, UK, Jul. 2005, pp. 244–249.
- [14] T. Xia and B. Xia, "Quasi-cyclic codes from extended difference families," in *Proc. IEEE Wireless Commun. and Networking Conf.*, vol. 2, New Orleans, USA, Mar. 2005, pp. 1036–1040.
- [15] M. Baldi, F. Chiaraluce, and R. Garelo, "On the usage of quasi-cyclic low-density parity-check codes in the McEliece cryptosystem," in *Proc. First Int. Conf. on Commun. and Electron. (ICCE'06)*, Hanoi, Vietnam, Oct. 2006, pp. 305–310.
- [16] M. Baldi, "Quasi-cyclic low-density parity-check codes and their application to cryptography," Ph.D. dissertation, Università Politecnica delle Marche, Ancona, Italy, Nov. 2006.
- [17] P. Lee and E. Brickell, "An observation on the security of McEliece's public-key cryptosystem," in *Advances in Cryptology - EUROCRYPT 88*, Springer-Verlag, Ed., 1988, pp. 275–280.
- [18] T. A. Berson, "Failure of the McEliece public-key cryptosystem under message-resend and related-message attack," *Advances in Cryptology - CRYPTO '97, Lecture Notes in Computer Science*, vol. 1294, pp. 213–220, Aug. 1997.
- [19] H. M. Sun, "Improving the security of the McEliece public-key cryptosystem," in *ASIACRYPT*, 1998, pp. 200–213.
- [20] X.-Y. Hu, M. Fossorier, and E. Eleftheriou, "On the computation of the minimum distance of low-density parity-check codes," in *Proc. IEEE ICC 2004*, vol. 2, Paris, France, Jun. 2004, pp. 767–771.
- [21] J. Stern, "A method for finding codewords of small weight," in *G. Cohen and J. Wolfmann, Coding Theory and Applications*, Springer-Verlag, Ed., no. 388 in Lecture Notes in Computer Science, 1989, pp. 106–113.
- [22] M. Hiroto, M. Mori, and M. Mori, "A probabilistic computation method for the weight distribution of low-density parity-check codes," in *Proc. IEEE ISIT 2005*, Adelaide, Australia, Sep. 2005, pp. 2166–2170.