On Decoding Schemes for the MDPC-McEliece Cryptosystem

Hannes Bartz and Gianluigi Liva
Institute of Communication and Navigation, Deutsches Zentrum für Luft- und Raumfahrt (DLR), Wessling, Germany
{hannes.bartz,gianluigi.liva}@dlr.de

***Abstract*—In this paper, classical (iterative) decoding schemes for moderate-density parity-check (MDPC) codes are considered. The algorithms are analyzed with respect to their error-correction capability as well as their resilience against a recently proposed reaction-based key-recovery attack on a variant of the MDPC-McEliece cryptosystem by Guo, Johansson and Stankovski (GJS). New message-passing decoding algorithms are presented and analyzed. The proposed decoding algorithms have an improved error-correction performance compared to existing hard-decision decoding schemes and can reduce the effectiveness of the GJS reaction-based attack for an appropriate choice of the algorithm's parameters.**

I. Introduction

In 1978, Rivest-Shamir-Adleman (RSA) proposed a pubilic-key cryptosystem whose security is based on the hard problem of factoring large integers. In 1999, Shor presented a factorization algorithm for quantum computers that is able to factor large integers in polynomial time [1] and thus renders the RSA cryptosystem insecure if quantum computer of sufficient scale can be built one day. This result gives rise to developing cryptosystems that are *post-quantum* secure.

McEliece proposed a code-based cryptosystem [2] that relies on the hardness of decoding an unknown linear error-correcting code and thus is resilient against efficient factorization attacks on quantum computers. One drawback of the scheme is the large key size and the rate-loss compared to the RSA cryptosystem. Many variants of the McEliece cryptosystem based on different code families were considered in the past. In particular, McEliece cryptosystems based on low-density parity-check (LDPC) allow for very small keys but suffer from feasible attacks on the low-weight dual code due to the sparse parity-check matrix [3]. Variants based on quasi-cyclic (QC)-LDPC codes that use sparse column scrambling matrices to increase the density of the public code parity-check matrix were proposed in [4], [5]. However, unfortunate choices of the column scrambling matrix allow for structural attacks [6].

The family of moderate-density parity-check (MDPC) codes admit a parity-check matrix of *moderate* density,[1] yielding codes with large minimum distance [7]. In [8] a McEliece cryptosystem based on QC-MDPC codes that defeats information set decoding attacks on the dual code due to the moderate density parity-check matrix is presented. For a given security level, the QC-MDPC cryptosystem allows for very small key sizes compared to other McEliece variants.

Recently, Guo, Johansson and Stankovski (GJS) presented a reaction-based key-recovery attack on the QC-MDPC system [9] which was modified in [10], [11] to attack the QC-LDPC cryptosystems [5], [12]. These attacks reveal the secret sparse parity-check matrix by observing the decoding failure probability for chosen ciphertexts that are constructed with error patterns of a specific structure. Modified versions of the attack can even break a system that uses CCA-2 secure conversions [13].

In this paper we analyze iterative decoding algorithms for (QC-) MDPC codes with respect to their error-correction capability and their resilience against the GJS attack [9]. We present novel hard-decision message-passing (MP) algorithms that can reduce the effectiveness of the GJS key-recovery attack from [9] and have an improved error-correction capability compared to existing hard-decision decoding schemes. Density evolution (DE) analysis for the novel decoding schemes is performed which allows to predict decoding thresholds as well as to optimize the parameters of the algorithm.

II. Preliminaries

Denote the binary field by $\mathbb{F}_2$ and let the set of $m \times n$ matrices over $\mathbb{F}_2$ be denoted by $\mathbb{F}^{m\times n}$. The set of all vectors of length n over $\mathbb{F}_2$ is denoted by $\mathbb{F}_2^n$. Vectors and matrices are denoted by bold lower-case and upper-case letters such as $\boldsymbol{a}$ and $\boldsymbol{A}$, respectively. A binary circulant matrix $\boldsymbol{A}$ of size Q is a $Q \times Q$ matrix with coefficients in $\mathbb{F}_2$ obtained by cyclically shifting its first row $\boldsymbol{a} = (a_0, a_1, \ldots, a_{Q-1})$ to right. The set of $Q \times Q$ circulant matrices together with the matrix multiplication and addition forms a commutative ring and it is isomorphic to the polynomial ring $\left(\mathbb{F}_2[X]/\left(X^Q - 1\right), +, \cdot\right)$. In particular, there is a bijective mapping between a circulant matrix $\boldsymbol{A}$ and a polynomial $a(X) = a_0 + a_1 X + \ldots + a_{Q-1}x^{Q-1} \in \mathbb{F}_2[X]$. We indicate the vector of coefficients of a polynomial $a(X)$ as $\boldsymbol{a} = (a_0, a_1, \ldots, a_{Q-1})$. The weight of a polynomial $a(X)$ is the number of its non-zero coefficients, i.e., it is the Hamming weight of its coefficient vector $\boldsymbol{a}$. We indicate both weights with the operator $\mathsf{wht}(\cdot)$, i.e., $\mathsf{wht}(a(X)) = \mathsf{wht}(\boldsymbol{a})$. In the remainder of this paper we use

[1] The existence of a moderate-density parity-check matrix for a binary linear block code does not rule out the possibility that the same code fulfills a (much) sparser parity-check matrix. As in most of the literature, we neglect the probability that a code defined by a randomly-drawn moderate parity check matrix admits a sparser parity-check matrix. Guarantees in this sense shall be derived based on random code ensemble arguments.

the polynomial representation of circulant matrices to provide an efficient description of the structure of the codes.

A. QC MDPC-based Cryptosystems

The QC-MDPC McEliece cryptosystem [8] allows for a very simple description without the need for row and column scrambling matrices. Due to the moderate density of the parity-check matrix, known decoding attacks on the dual code [3] are defeated. The parity-check matrix consists of blocks of $Q \times Q$ circulant matrices which allows for very small key sizes due to the compact description of the circulant blocks.

A binary MDPC code of length n, dimension k and row weight d_c is defined by a binary parity-check matrix $\boldsymbol{H}$ that contains a moderate number of $d_c \approx \mathcal{O}(\sqrt{n \log(n)})$ ones per row. For $n = N_0 Q$, dimension $k = K_0 Q$, redundancy $r = n - k = R_0 Q$ with $R_0 = N_0 - K_0$ for some integer Q, the parity-check matrix $\boldsymbol{H}(X)$ of a QC-MDPC code in polynomial form is a $R_0 \times N_0$ matrix.

Without loss of generality we consider in the following codes with $r = Q$ (i.e. $R_0 = 1$). The parity-check matrix of QC-MDPC codes with $r = Q$ has the form

$$\boldsymbol{H}(X) = \begin{pmatrix} h_0(X) & h_1(X) & \dots & h_{N_0-1}(X) \end{pmatrix}. \qquad (1)$$

Let $\mathrm{DEC}_{\boldsymbol{H}}(\cdot)$ be an efficient decoder for the code defined by the parity-check matrix $\boldsymbol{H}$.

Key generation:

- Randomly generate a parity-check matrix $\boldsymbol{H} \in \mathbb{F}_2^{r \times n}$ of the form (1) with $\mathsf{wht}\,(h_i(X)) = d_c^{(i)}$ for $i = 0, \dots, N_0 - 1$. The matrix $\boldsymbol{H}$ with row weight $d_c = \sum_{i=0}^{N_0-1} d_c^{(i)}$ is the *private* key.
- The *public* key is the corresponding binary $k \times n$ generator matrix in systematic form, i.e.,

$$\boldsymbol{G}(X) = \left(\begin{array}{c|c} \boldsymbol{I} & (g_0(X), \dots, g_{K_0-1}(X))^T \end{array} \right).$$

  $\boldsymbol{G}(X)$ can be described by $K_0 Q$ bits (public key size).

Encryption:

- To encrypt a plaintext[2] $\boldsymbol{u} \in \mathbb{F}_2^k$ a user computes the ciphertext $\boldsymbol{c} \in \mathbb{F}_2^n$ using the public key $\boldsymbol{G}$ as

$$\boldsymbol{c} = \boldsymbol{u}\boldsymbol{G} + \boldsymbol{e} \qquad (2)$$

  where $\boldsymbol{e}$ is an error vector uniformly chosen from all vectors from $\mathbb{F}_2^n$ of Hamming weight $\mathsf{wht}\,(\boldsymbol{e}) = e$.

Decryption:

- To decrypt a ciphertext $\boldsymbol{c}$ the authorized recipient uses the private key $\mathrm{DEC}_{\boldsymbol{H}}(\cdot)$ to obtain

$$\boldsymbol{u}\boldsymbol{G} = \mathrm{DEC}_{\boldsymbol{H}}(\boldsymbol{u}\boldsymbol{G} + \boldsymbol{e}).$$

- Since $\boldsymbol{G}$ is in systematic form the plaintext $\boldsymbol{u}$ corresponds to the first k bits of $\boldsymbol{u}\boldsymbol{G}$.

[2]As in [8] we assume that the CCA-2 security conversions from [13] are used to allow for systematic encoding without security reduction.

B. A Reaction-Based Attack on the QC-MDPC McEliece Cryptosystem

GJS proposed a reaction-based key-recovery attack on the QC-MDPC McEliece cryptosystem [8] which is currently the most critical attack against the scheme [14]. Efficient iterative decoding of LDPC/MDPC codes comes at the cost of decoding failures. The GJS attack exploits the observation that the decoding failure probability for some particular error patterns is correlated with the structure of the parity-check matrix $\boldsymbol{H}$ (secret key). We now describe briefly how the attack proceeds.

The *Lee distance* d_L between two entries at position i and j of a binary vector $\boldsymbol{a} = (a_0\ a_1\ \dots\ a_{n-1})$ is defined as [15]

$$d_L(i,j) \stackrel{\text{def}}{=} \min\{|i-j|, n-|i-j|\}.$$

The *Lee distance profile*[3] of a binary vector $\boldsymbol{a}$ of length Q is

$$D(\boldsymbol{a}) \stackrel{\text{def}}{=} \{d : \exists i,j \in (0, Q-1) \text{ s.t. } a_i = a_j = 1 \text{ and } d_L(i,j) = d\}$$

where the maximum distance in $D(\boldsymbol{a})$ is $U = \lfloor \frac{Q}{2} \rfloor$. The *multiplicity* $\mu(d)$ is defined as the number of occurrences of distance d in the vector $\boldsymbol{a}$. A binary vector $\boldsymbol{a}$ is fully specified by its distance profile $D(\boldsymbol{a})$ and thus can be reconstructed with high probability from $D(\boldsymbol{a})$ [9] (up to cyclic shifts).

Let Ψ_d be a set containing all binary vectors of length n with exactly t ones that are placed as $\lfloor \frac{t}{2} \rfloor$ pairs with Lee distance d in the first Q positions of the vector. By limiting the errors to the first Q positions, only the first circulant block $h_0(X)$ of the matrix $\boldsymbol{H}(X)$ will determine the result of the decoding procedure. The GJS attack proceeds as follows:

- For $d = 1, \dots, U$ generate error sets Ψ_d of size M each (with M being a parameter defining, together with U, the number of attempts used by the attacker).
- Send M ciphertexts (2) with $\boldsymbol{e} \in \Psi_d$ for all $d = 1, \dots, U$ and measure the frame error rate (FER).

Since the decoding failure probability is lower for $\boldsymbol{e} \in \Psi_d$ with $d \in D(\boldsymbol{h}_0)$, i.e. if $\mu(d) > 0$, for sufficiently large M the measured FER can be used to determine the distance profile $D(\boldsymbol{h}_0)$. The vector $\boldsymbol{h}_0$ can then be reconstructed from the distance profile $D(\boldsymbol{h}_0)$ using the methods from [9].

The remaining blocks of $\boldsymbol{H}(X)$ in (1) can then be reconstructed via the generator matrix $\boldsymbol{G}(X)$ using linear algebraic relations. The success on the attack depends on how the systems deals with decoding failures since the FER can only be measured if retransmissions are requested. Another important factor is which decoding scheme is used. In [9], [10] it is shown that the GJS attack succeeds if bit-flipping (BF) or belief propagation (BP) decoding algorithms are used.

The attack can be defeated on a protocol level by limiting the lifetime of a key (see [12]). However, this protocol-based fix affects the performance of the cryptosystem since it requires a recurring generation of key-pairs. Hence, decoding algorithms that are more robust against this attack can increase the lifetime of a key and thus improve the performance of the cryptosystem.

[3]We use the term "Lee distance profile" (instead of "distance spectrum" as in [8]) to avoid confusion with the distance spectrum of linear block codes.

C. Classical Decoding Algorithms

In the following we describe classical decoding algorithms for LDPC codes and analyze their error-correction capability for MDPC codes as well as their resilience against the GJS attack. For decoding we map each ciphertext bit c_i to $+1$ if $c_i = 0$ and -1 if $c_i = 1$ yielding (with some abuse of notation) a ciphertext $\boldsymbol{c} \in \{+1, -1\}^n$. We consider next iterative MP decoding on a bipartite graph consisting of n variable nodes (VNs) and r check nodes (CNs). A VN v_j is connected to a CN c_i if the corresponding entry $h_{i,j}$ in the parity-check matrix is equal to 1. We consider next only regular graphs, i.e, graphs for which the number of edges emanating from each VN equals d_v and the number of edges emanating from each CN equals d_c. We refer to d_v and d_c as variable and check node degree, respectively. The neighborhood of a variable node v is $\mathcal{N}(\mathsf{v})$, and similarly $\mathcal{N}(\mathsf{c})$ denotes the neighborhood of the check node c. We denote the messages from VN v_j to CN c_i by $m_{\mathsf{v}_j \to \mathsf{c}_i}$ and the messages from c_i to v_j by $m_{\mathsf{c}_i \to \mathsf{v}_j}$. In the following we omit the indices of VNs and CNs whenever they are clear from the context. For the following algorithms (except BP), each VN v is initialized with the corresponding ciphertext bit $c \in \{+1, -1\}$.

1) Bit-Flipping: For decryption in the QC-MDPC cryptosystem [8] an efficient BF algorithm for LDPC codes (see e.g. [16, Alg. 5.4]) is considered. This algorithm is often referred to as "Gallager's bit-flipping" algorithm although it is *different* from the algorithm proposed by Gallager in [17].

Given a ciphertext $\boldsymbol{c}$, a threshold $b \leq r$ and a maximum number if iterations $I_{\max}$, the BF algorithm proceeds as follows. Each VN sends the message $m_{\mathsf{v}\to\mathsf{c}} = c$ to all neighboring CNs $\mathsf{c} \in \mathcal{N}(\mathsf{v})$. The CNs send the messages

$$m_{\mathsf{c}\to\mathsf{v}} = \prod_{\mathsf{v}' \in \mathcal{N}(\mathsf{c})} m_{\mathsf{v}'\to\mathsf{c}}$$

to all neighboring VNs $\mathsf{v} \in \mathcal{N}(\mathsf{c})$. Each variable node counts the number of unsatisfied check equations and sends to its neighbors the "flipped" ciphertext bit if at least b parity-check equations are unsatisfied, i.e.

$$m_{\mathsf{v}\to\mathsf{c}} = \begin{cases} -c & \text{if } |\{\mathsf{c}' \in \mathcal{N}(\mathsf{v}) : m_{\mathsf{c}'\to\mathsf{v}} = -1\}| \geq b \\ c & \text{otherwise} \end{cases}.$$

The algorithm terminates if either all checks are satisfied or the maximum number of iterations $I_{\max}$ is reached.

In [8] it is suggested to compute b according to [17, p. 46, Eq. 4.16] which will lead to suboptimal results since the BF decoder is different from the decoder analyzed in [17].

2) Gallager B: An efficient binary MP decoder for LDPC codes, often referred to as *Gallager B*, was presented an analyzed in [17]. The VN send the messages

$$m_{\mathsf{v}\to\mathsf{c}} = \begin{cases} -c & \text{if } |\{\mathsf{c}' \in \mathcal{N}(\mathsf{v}) \setminus \mathsf{c} : m_{\mathsf{c}'\to\mathsf{v}} = -c\}| \geq b \\ c & \text{else} \end{cases}. \tag{3}$$

This means that in the first iteration VN v sends the message $m_{\mathsf{v}\to\mathsf{c}} = c$ to all neighboring CNs $\mathsf{c} \in \mathcal{N}(\mathsf{v})$. The CNs send the messages

$$m_{\mathsf{c}\to\mathsf{v}} = \prod_{\mathsf{v}' \in \mathcal{N}(\mathsf{c})\setminus\mathsf{v}} m_{\mathsf{v}'\to\mathsf{c}} \tag{4}$$

to the neighboring VNs. After iterating (3), (4) at most $I_{\max}$ times, the final decision is given by

$$\hat{c} = \begin{cases} -c & \text{if } |\{m_{\mathsf{c}\to\mathsf{v}} = -c\}| > b \\ c & \text{else} \end{cases}. \tag{5}$$

For fixed (d_v, d_c) the average error correction capability over the binary symmetric channel (BSC) and the optimal value for b (see [17, Eq. 4.16]) for the ensemble of (d_v, d_c) LDPC codes can be analyzed, in the limit of large block lengths, using the density evolution (DE) analysis [17], [18].

3) Miladinovic-Fossorier (MF) Algorithms: Two probabilistic variants of Gallager's algorithm B were proposed by Miladinovic and Fossorier in [19, Sec. III.A]. At each iteration ℓ the VN to CN messages (3) in Gallager B are modified with a certain probability $p_e^{(\ell)}$. By defining an initial value $p_e^{(0)} = p^*$ and a decrement $p_{\text{dec}} \leq p^*$, one can compute $p_e^{(\ell)}$ by

$$p_e^{(\ell)} = \begin{cases} p_e^{(\ell-1)} - p_{\text{dec}} & \text{if } p_e^{(\ell-1)} > p_{\text{dec}} \\ 0 & \text{else} \end{cases}. \tag{6}$$

Variant 1 (Miladinovic and Fossorier (MF)-1): If the number of incoming CN messages different from c that do not agree with c exceeds the threshold b, i.e. if $|\{\mathsf{c}' \in \mathcal{N}(\mathsf{v}) \setminus \mathsf{c} : m_{\mathsf{c}'\to\mathsf{v}} = -c\}| \geq b$, the VNs send the messages

$$m_{\mathsf{v}\to\mathsf{c}} = \begin{cases} -c & \text{with probability } 1 - p_e^{(\ell)} \\ c & \text{with probability } p_e^{(\ell)} \end{cases}$$

and $m_{\mathsf{v}\to\mathsf{c}} = c$ otherwise.

Variant 2 (MF-2): With respect MF-1, we shall now introduce the iteration counter for the messages that are output by VNs and by CNs. At iteration ℓ, is the number of message at the input of a VN v sent by its neighboring CNs exceeds the threshold b, i.e. if $|\{\mathsf{c}' \in \mathcal{N}(\mathsf{v}) \setminus \mathsf{c} : m_{\mathsf{c}'\to\mathsf{v}}^{(\ell-1)} = -c\}| \geq b$, the VN sends the message

$$m_{\mathsf{v}\to\mathsf{c}}^{(\ell)} = \begin{cases} -c & \text{with probability } 1 - p_e^{(\ell)} \\ m_{\mathsf{v}\to\mathsf{c}}^{(\ell-1)} & \text{with probability } p_e^{(\ell)} \end{cases}$$

while $m_{\mathsf{v}\to\mathsf{c}}^{(\ell)} = c$ otherwise.

The check node operation as well as the final decision remains the same as in Gallager B (see (4) and (5)). By definition the probability $p_e^{(\ell)}$ has two degrees of freedom, namely p^* and p_{dec}, which are subject to optimization.

4) Algorithm E: A generalization of Gallager B that exploits erasures, further referred to as *Algorithm E*, was introduced and analyzed in [18], [20]. To incorporate erasures the decoder requires a ternary message alphabet $\{-1, 0, +1\}$, where 0 indicates an erasure. The VNs send the messages

$$m_{\mathsf{v}\to\mathsf{c}} = \text{sign}\left[\omega c + \sum_{\mathsf{c}' \in \mathcal{N}(\mathsf{v})\setminus\mathsf{c}} m_{\mathsf{c}'\to\mathsf{v}}\right]. \tag{7}$$

Here, ω is a heuristic weighting factor that was proposed in [18] improve the performance of Algorithm E which may vary over iterations. We consider next the simple case where ω is kept constant through all iterations. The check nodes operate the same way as in Gallager B, i.e the CNs send the messages $m_{\mathsf{c}\to\mathsf{v}}$ according to (4). After iterating (4) and (7) at most $I_{\max}$ times, the final decision is made as

$$\hat{c} = \operatorname{sign}\left[\omega c + \sum_{\mathsf{c}\in\mathcal{N}(\mathsf{v})} m_{\mathsf{c}\to\mathsf{v}}\right].$$

In [18] a DE analysis for Algorithm E was derived which allows to compute an estimate of the optimal weight ω. For odd d_v Algorithm E is equivalent to Gallager B with threshold $b = \lceil \frac{\omega+d_v-1}{2} \rceil$ and thus is also vulnerable against the GJS attack.

5) Belief Propagation (BP) Decoding: BP decoding is a soft-decision decoding algorithm that is optimum in the maximum a posteriori (MAP) sense over a cycle-free graph. Each VN v is initialized with the log-likelihood ratios

$$m_{\text{ch}} = c \ln \frac{n-e}{n}$$

where c is ciphertext bit corresponding to v. The VNs send the messages

$$m_{\mathsf{v}\to\mathsf{c}} = m_{\text{ch}} + \sum_{\mathsf{c}'\in\mathcal{N}(\mathsf{v})\setminus\mathsf{c}} m_{\mathsf{c}'\to\mathsf{v}} \tag{8}$$

to the CNs. In turn, the CNs send the messages

$$m_{\mathsf{c}\to\mathsf{v}} = 2\tanh^{-1}\left[\prod_{\mathsf{v}'\in\mathcal{N}(\mathsf{c})\setminus\mathsf{v}} \tanh\left(\frac{m_{\mathsf{v}'\to\mathsf{c}}}{2}\right)\right]. \tag{9}$$

After iterating (8), (9) at most $I_{\max}$ times, the final decision at each VN is made as

$$\hat{c} = \operatorname{sign}\left[m_{\text{ch}} + \sum_{\mathsf{c}\in\mathcal{N}(\mathsf{v})} m_{\mathsf{c}\to\mathsf{v}}\right].$$

It was conjectured for QC-MDPC codes [8] and finally shown for QC-LDPC codes [10] that the GJS attack is also successful for QC-MDPC McEliece cryptosystems under BP decoding.

D. Simulation Results

We now present simulation results of the GJS attack on variants of the QC-MDPC cryptosystem using the above described schemes. We consider next an QC-MDPC code ensemble $\mathscr{C}$ with $n = 9602$ and $k = 4801$ and parity-check matrix in the form

$$\boldsymbol{H}(X) = (h_0(X)\ h_1(X))$$

where $h_0(X)$ and $h_1(X)$ are two polynomials of degree less than 4801 and $\mathsf{wht}(h_0) = \mathsf{wht}(h_1) = 45$. The ensemble $\mathscr{C}$ was proposed in [8] for 80 bit security. To analyze the resilience against the GJS attack, we performed Monte Carlo simulations for codes randomly picked from $\mathscr{C}$ collecting up to

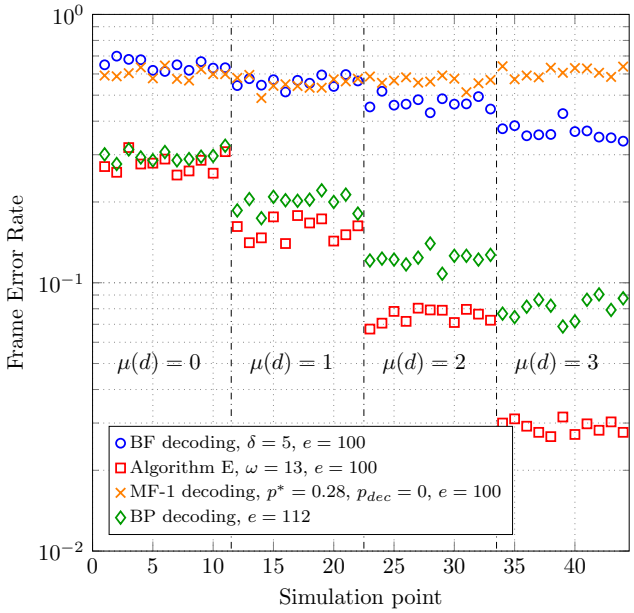


Fig. 1. GJS reaction-based attack on the code ensemble $\mathscr{C}$ with BF decoding, MF decoding, Algorithm E and BP decoding. For the MF-2 decoder an attacker needs to collect much more samples to reconstruct $D(\boldsymbol{h}_0)$.

200 decoding failures (frame errors) with $I_{\max} = 50$ iterations. For each multiplicity in $D(\boldsymbol{h}_0)$, 11 different error sets Ψ_d (simulation points) were simulated. As in [10] the weight of the error patterns was chosen such that the FER is high enough to be easily observable in the simulations.

The simulation results in Figure 1 for one code from $\mathscr{C}$ show that all considered schemes are vulnerable against the GJS attack. For the MF decoding scheme the probability $p_e^{(\ell)}$ was chosen such that the FER for all multiplicities appearing in $D(\boldsymbol{h}_0)$ are similar. To be able to reconstruct the distance profile $D(\boldsymbol{h}_0)$ if the MF decoding scheme with the appropriate choice of $p_e^{(\ell)}$ is used, the attacker needs to collect much more samples compared to the other approaches. Since simulations of different codes from $\mathscr{C}$ show very similar results we conjecture that the choice of $p_e^{(\ell)}$ rather depends on the code ensemble than on the code itself.

III. Secret Key Concealment via Modified Iterative Decoding

In this section we propose new methods to modify MP decoding algorithms that admit erasures. The methods allow to modify MP decoding algorithms in a probabilistic manner to combat the GJS attack for an appropriate choice of the decoding parameters. The main idea is, that similar to the MF decoding scheme (see Sec. II-C3), we modify the VN to CN messages at each iteration with a given probability. In particular, we modify the MP decoder such that the messages $m_{\mathsf{v}\to\mathsf{c}}$ are erased (i.e., set to 0) under certain conditions with a given probability $p_e^{(\ell)}$. Remarkably, we will see how this results also in an improved error-correction capability. We will refer to this approach as random erasure message-passing (REMP) decoding and we apply it to modify Algorithm E.

A. First Modification of Algorithm E (REMP-1)

We modify Algorithm E such that any nonzero message $m_{\mathsf{v}\to\mathsf{c}}$ in iteration ℓ is erased with probability $p_e^{(\ell)}$. At the VNs we first compute a temporary output message

$$\tilde{m}_{\mathsf{v}\to\mathsf{c}} = \operatorname{sign}\left[\omega c + \sum_{\mathsf{c}'\in\mathcal{N}(\mathsf{v})\setminus\mathsf{c}} m_{\mathsf{c}'\to\mathsf{v}}\right].$$

If the message $\tilde{m}_{\mathsf{v}\to\mathsf{c}}$ is not an erasure, i.e. if $\tilde{m}_{\mathsf{v}\to\mathsf{c}} \neq 0$, the VN sends

$$m_{\mathsf{v}\to\mathsf{c}} = \begin{cases} \tilde{m}_{\mathsf{c}\to\mathsf{v}} & \text{with probability } 1-p_e^{(\ell)} \\ 0 & \text{with probability } p_e^{(\ell)} \end{cases} \tag{10}$$

and $m_{\mathsf{v}\to\mathsf{c}} = 0$ else. At the CNs we perform the same operation as in Algorithm E (see (4)). The final decision, after iterating (4) and (10) at most $I_{\max}$ times, is given by (12). As for the MF algorithm, the probability $p_e^{(\ell)}$ may be decreased as ℓ grows following (6).

B. Second Modification of Algorithm E (REMP-2)

In the second modification of Algorithm E from Sec. II-C4 the messages $m_{\mathsf{v}\to\mathsf{c}}$ at iteration ℓ are erased (i.e. set to $m_{\mathsf{v}\to\mathsf{c}} = 0$) with probability $p_e^{(\ell)}$ if they contradict the corresponding ciphertext bit c. At the VNs we first compute a temporary output message

$$\tilde{m}_{\mathsf{v}\to\mathsf{c}} = \operatorname{sign}\left[\omega c + \sum_{\mathsf{c}'\in\mathcal{N}(\mathsf{v})\setminus\mathsf{c}} m_{\mathsf{c}'\to\mathsf{v}}\right].$$

If the message $\tilde{m}_{\mathsf{v}\to\mathsf{c}}$ contradicts the ciphertext bit c, i.e. if we have $\tilde{m}_{\mathsf{v}\to\mathsf{c}} = -c$, the VN sends

$$m_{\mathsf{v}\to\mathsf{c}} = \begin{cases} \tilde{m}_{\mathsf{c}\to\mathsf{v}} & \text{with probability } 1-p_e^{(\ell)} \\ 0 & \text{with probability } p_e^{(\ell)} \end{cases} \tag{11}$$

and $m_{\mathsf{v}\to\mathsf{c}} = \tilde{m}_{\mathsf{c}\to\mathsf{v}}$ otherwise. At the check nodes we perform the same operation as in Algorithm E (see (4)). The final decision, after iterating (4) and (11) at most $I_{\max}$ times, is given by

$$\hat{c} = \operatorname{sign}\left[\omega c + \sum_{\mathsf{c}\in\mathcal{N}(\mathsf{v})} m_{\mathsf{c}\to\mathsf{v}}\right]. \tag{12}$$

Again, as for the MF algorithm, the probability $p_e^{(\ell)}$ may be decreased as ℓ grows following (6).

C. Performance Analysis & Simulation Results

1) Density Evolution Analysis: We first analyze the error-correction capability of the two modifications of Algorithm E from Sec III-A and Sec. III-B. As first estimate of the code performance, we employ the DE analysis [18] to determine the iterative decoding threshold of a (d_v, d_c) unstructured MDPC code ensemble over a BSC with error probability Δ (see full version of the paper [21]). The decoding threshold is denoted as $\Delta^\star$ and represents the largest channel error probability for which, in the limit of large n and large $I_{\max}$, the bit error probability of code picked randomly from the ensemble becomes vanishing small [18]. We then get a rough estimate on the error correction capability as[4] $\delta^\star = \lfloor n\Delta^\star \rfloor$. For a moderate block length n, $\delta^\star$ provides only a coarse estimate to the number of errors at which we expect the FER to rapidly decrease (so-called waterfall region), with the accuracy of the prediction improving as n grows large. With a slight abuse of the wording, we refer to $\delta^\star$ as decoding threshold as well. We further denote the decoding threshold under Algorithm E, REMP-1 and REMP-2 as $\delta_{\mathsf{E}}^\star$, $\delta_1^\star$ and $\delta_2^\star$, respectively. The decoding thresholds do not only depend on the selected algorithm, but also on the algorithm parameters. The results for the code ensembles with $N_0 = 2$ for the security levels (SLs) of $80, 128$ and 256 bit from [8, Tab. 2] are summarized in Table I. For Algorithm E, the value of ω has been chosen to maximize the decoding threshold. For REMP-2 we have that $p_{\mathrm{dec}} = 0$. In some cases, the variants REMP-1/2 provide gains for suitable choices of the parameters $(\omega, p^*, p_{\mathrm{dec}})$.

TABLE I
DECODING THRESHOLDS OF ALGORITHM E AND IT VARIANTS FOR THE MDPC CODE ENSEMBLES WITH THE PARAMETERS FROM [8, TAB. 2].

	REMP-1			REMP-2		Alg. E
SL	p^*	p_{dec}	$\delta_1^\star\ (\omega)$	p^*	$\delta_2^\star\ (\omega)$	$\delta_{\mathsf{E}}^\star\ (\omega)$
80	10^{-3}	0	107(13)	0.1	108(13)	106(14)
128	10^{-1}	10^{-3}	153(18)	0.76	157(14)	153(18)
256	$2\cdot 10^{-3}$	$2\cdot 10^{4}$	296(27)	0.65	301(23)	294(26)

2) Simulation Results: To validate the performance estimates obtained through DE, we simulated the error-correction capability of the decoding schemes from Section II-C and Section III. The results in terms of FER as a function of the error pattern weight are depicted in Figure 2. The results confirm the trend predicted by the DE analysis that the error-correction capability improves upon existing decoding algorithms. Even for erasure probability values chosen to conceal the structure of $\boldsymbol{H}(X)$ (yielding a suboptimal choice with respect to the error correction performance), REMP-2 outperforms Algorithm E and the BF/MF algorithms.

D. Resilience Against the GJS Attack

We now analyze the resilience of the proposed decoding schemes against the GJS attack. For the REMP-1/2 decoding schemes we performed Monte Carlo simulations for codes randomly picked from $\mathscr{C}$ collecting up to 200 decoding failures (frame errors) with $I_{\max} = 50$ iterations. For each multiplicity in $D(\boldsymbol{h}_0)$, 11 different error sets Ψ_d (simulation points) were simulated. The simulation results in Figure 3 show that, for an appropriate choice of parameters, the REMP-1/2 decoding schemes have a similar FER for all multiplicities appearing in $D(\boldsymbol{h}_0)$. Hence, the reconstruction of the distance profile $D(\boldsymbol{h}_0)$ from the observed FER is much harder which

[4] At the decoding threshold $\Delta^\star$ a vanishing small bit error probability may not imply a vanishing small block error probability. For the regular MDPC ensembles under consideration the threshold on the bit error probability and the one on the block error probability do coincide over binary-input output-symmetric memoryless channel under BP decoding [22]. In our estimate, we implicitly assume that the result extends to Algorithm E and its variants.

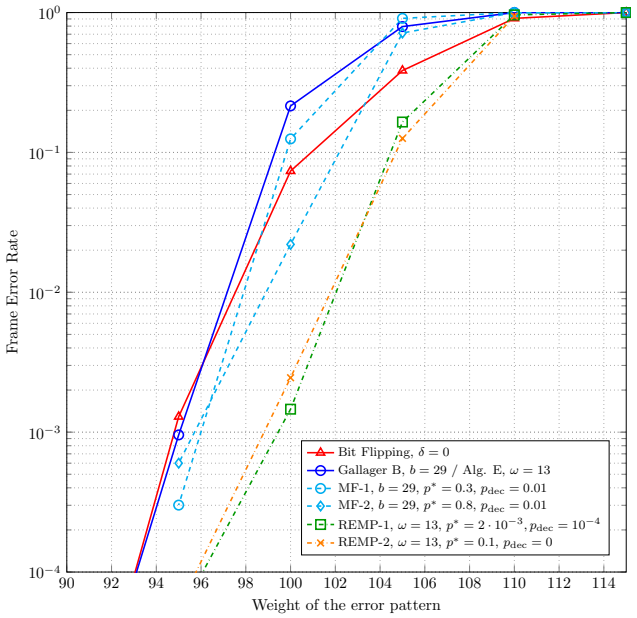
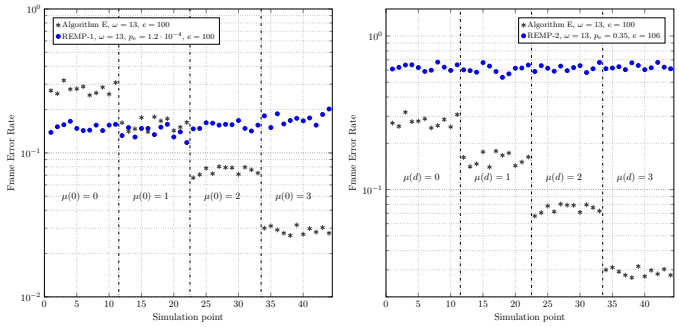


Fig. 2. Error-correction performance (FER) over the weight of the error patterns. The figure shows that the proposed REMP schemes significantly improve upon existing hard-decision decoding schemes.



(a) (b)

Fig. 3. GJS reaction-based attack on the code ensemble $\mathscr{C}$ with (a) REMP-1 and (b) REMP-2 decoding. The results show that an attacker has to collect much more samples to be able to reconstruct the distance profile $D(\boldsymbol{h}_0)$.

significantly delays the GJS attack and increases the lifetime of the public key. To conceal the structure of $\boldsymbol{H}(X)$ the choice of $p_e^{(\ell)}$ for a particular error weight e is crucial. If $p_e^{(\ell)}$ is chosen too large the picture is inverted, i.e. higher multiplicities have a higher FER than lower multiplicities. Thus the error weight e should be computed after decoding and ciphers generated with an error weight different from e should be rejected to prevent attacks that exploit this effect.

IV. Conclusions

Classical iterative decoding schemes for moderate-density parity-check (MDPC) codes were analyzed with respect to their error-correction capability as well as their resilience against the recent Guo, Johansson and Stankovski (GJS) key-recovery attack. A new decoding method called random erasure message-passing (REMP) that allows to improve existing message-passing (MP) decoding algorithms with respect to their error-correction capability as well as their resilience against the GJS attack was proposed. Two REMP variants of an existing MP decoder that have an improved error-correction performance for MDPC codes compared to existing schemes were presented and analyzed. The simulation results show that the proposed REMP schemes significantly reduce the effectiveness of the GJS attack for an appropriate choice of decoding parameters.

References

[1] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.

[2] R. J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Codes," *Deep Space Network Progr. Report*, vol. 44, pp. 114–116, 1978.

[3] C. Monico, J. Rosenthal, and A. Shokrollahi, "Using Low Density Parity Check Codes in the McEliece Cryptosystem," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Sorrento, Italy, Jun. 2000, p. 215.

[4] M. Baldi and F. Chiaraluce, "Cryptanalysis of a New Instance of McEliece Cryptosystem based on QC-LDPC Codes," in *2007 IEEE Int. Symp. on Inf. Theory*, June 2007, pp. 2591–2595.

[5] M. Baldi, M. Bodrato, and F. Chiaraluce, "A New Analysis of the McEliece Cryptosystem Based on QC-LDPC Codes," in *Int. Conf. on Sec. and Crypt. for Networks*. Springer, 2008, pp. 246–262.

[6] A. Otmani, J.-P. Tillich, and L. Dallot, "Cryptanalysis of two McEliece Cryptosystems based on Quasi-Cyclic Codes," *Math. in Comp. Science*, vol. 3, no. 2, pp. 129–140, 2010.

[7] S. Ouzan and Y. Be'ery, "Moderate-Density Parity-Check Codes," *arXiv preprint arXiv:0911.3262*, 2009.

[8] R. Misoczki, J. P. Tillich, N. Sendrier, and P. S. L. M. Barreto, "MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 2069–2073.

[9] Q. Guo, T. Johansson, and P. Stankovski, "A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors," in *ASIACRYPT 2016*, Hanoi, Vietnam, Dec. 2016, pp. 789–815.

[10] T. Fabšič, V. Hromada, P. Stankovski, P. Zajac, Q. Guo, and T. Johansson, "A Reaction Attack on the QC-LDPC McEliece Cryptosystem," in *Int. Workshop on Post-Quantum Cryptography*, 2017, pp. 51–68.

[11] T. Fabsic, V. Hromada, and P. Zajac, "A Reaction Attack on LEDApkc," Cryptology ePrint Archive, Report 2018/140, 2018, https://eprint.iacr.org/2018/140.

[12] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, "LEDApkc: Low-dEnsity parity-check coDe-bAsed public-key cryptosystem," 2018, https://www.ledacrypt.org/LEDApkc/.

[13] K. Kobara and H. Imai, "Semantically secure McEliece public-key cryptosystems-conversions for McEliece PKC," in *4th Int. Workshop on Pract. and Theory in PKC*, Cheju Island, South Korea, Feb. 2001, pp. 19–35.

[14] N. Sendrier, "Code-Based Cryptography: State of the Art and Perspectives," *IEEE Security & Privacy*, vol. 15, no. 4, pp. 44–50, Aug. 2017.

[15] C. Lee, "Some Properties of Nonbinary Error-Correcting Codes," *IRE Transactions on Information Theory*, vol. 4, no. 2, pp. 77–82, Jun. 1958.

[16] W. Ryan and S. Lin, *Channel codes – Classical and modern*. New York, NY, USA: Cambridge University Press, 2009.

[17] R. Gallager, *Low-density parity-check codes*. Cambridge, MA, USA: MIT Press, 1963.

[18] T. Richardson and R. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599 – 618, Feb. 2001.

[19] N. Miladinovic and M. P. Fossorier, "Improved Bit-Flipping Decoding of Low-Density Parity-Check Codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1594–1606, Apr. 2005.

[20] M. Mitzenmacher, "A Note on Low Density Parity Check Codes for Erasures and Errors," *SRC Technical Note*, vol. 1998, no. 17, 1998.

[21] H. Bartz and G. Liva, "On decoding schemes for the mdpc-mceliece cryptosystem," *arXiv preprint arXiv:1801.05659*, 2018.

[22] M. Lentmaier, D. V. Truhachev, K. S. Zigangirov, and D. J. Costello, "An analysis of the block error probability performance of iterative decoding," *IEEE Trans. Inf. Theory*, vol. 51, no. 11, pp. 3834–3855, Nov 2005.