A Reaction Attack against Cryptosystems Based on Quasi–Group MDPC Codes

Kirill Vedenev
Departament of Algebra and Discrete Mathematics
Southern Federal University
Rostov–on–Don, Russia
vedenevk@gmail.com

Yury Kosolapov
Departament of Algebra and Discrete Mathematics
Southern Federal University
Rostov–on–Don, Russia
puzzlestorage@gmail com

***Abstract*—Cryptosystems based on quasi–cyclic moderate density parity–check (QC–MDPC) codes are considered among the most perspective post–quantum public–key encryption schemes due to small public–key sizes and excellent performance. However, due to probabilistic decoding of MDPC codes, there is non–zero decryption failure rate. In 2016, Q. Guo, T. Johansson, P. Stankovski showed that decryption failures can be used to construct a key–recovery attack against QC–MDPC cryptosystems. Recently, in order to mitigate GJS attack, P. Santini, E. Persichetti, and M. Baldi proposed generalization of quasi–cyclic codes called quasi–reproducible (QR) codes and QR–MDPC cryptosystems. In this paper, we consider cryptosystems based on binary and non–binary quasi–group (QG) MDPC codes and propose a generalization of GJS reaction attack against these cryptosystems. We show that many efficient QR–MDPC cryptosystems are in fact equivalent to QG–MDPC cryptosystems, making the proposed attack applicable to them as well.**

***Index Terms*—Code–Based cryptography, MDPC codes, QC–MDPC, quasi–cyclic codes, reproducible codes, group algebras, DFR, reaction attack**

I. Introduction

Code-based cryptography is a type of public-key cryptography whose security is based on the hardness of decoding random linear codes. Since integer factorization and discrete logarithm problems used in traditional cryptosystems like RSA and elliptic-curve cryptosystems can be attacked in polynomial time using quantum computers with Shor's algorithm [1], code-based cryptography has received a lot of attention in recent years as a potential approach that can withstand both classical and quantum attacks [2].

The first code–based cryptosystem was proposed by R. McEliece in his seminal paper [3]. His approach involved using the matrix $\tilde{G} = SGP$ as the public key, where G is a generator matrix of a binary t-error correcting Goppa code, and S and P are random $k \times k$ invertible and $n \times n$ permutation matrices, respectively. Encryption of a message m was performed as $y = m\tilde{G} + \varepsilon$, where ε is a random error of weight t. Given the matrices S and P, it is straightforward to recover m. It should be noted that optimized modern version of Goppa code–based McEliece cryptosystem [4] is still considered secure and was selected as a finalist in round 3 of the NIST post–quantum standardization competition [5].

With the major drawback of the McEliece cryptosystem being its large public key size, there have been attempts to replace Goppa codes with more efficient ones and modify the protocol itself. However, many of these attempts have been proven insecure (see surveys [6], [7]).

One of the most promising approaches to constructing code-based public-key encryption with small public keys is based on using random quasi-cyclic codes with sparse parity-check matrices. Indeed, quasi-cyclic codes admit generator and parity-check matrices of block-circulant form, i.e., consisting of square circulant blocks. In addition, low-density parity-check codes (LDPC), proposed by R. Gallager [8], which are characterized by parity-check matrices with row weights of $O(1)$, are known for their extremely efficient decoding. However, for cryptographic applications, moderate-density parity check codes (MDPC) [9] are more preferable since MDPC codes do not require additional hiding mechanisms to prevent key-recovery attacks based on low-weight dual codewords search. Another significant advantage of MDPC codes, compared to traditional algebraic codes, is their general lack of algebraic structure. So, the use of MDPC codes helps in avoiding algebraic key-recovery attacks.

The generic QC–MDPC cryptosystem over the finite field $\mathbb{F}_q$ in the Nieddiriter form can be described as follows:

- *Key generation.* The secret key is the parity-check matrix
$$H = \left(H_1 \mid H_2 \mid \ \ldots \ \mid H_{l-1} \mid H_l\right) \tag{1}$$
of a random QC-MDPC $[n = ln', (l-1)n']_q$-code, with H_i being circulant $(n' \times n')$–blocks H_i having a row weight γ. The public key is the systematic form of H:
$$\tilde{H} = H_1^{-1}H = \left(I_{n'} \mid H_1^{-1}H_2 \mid \ \ldots \ \mid H_1^{-1}H_l,\right). \tag{2}$$
- *Encryption.* The plaintext is an error vector $\varepsilon \in \mathbb{F}_q^n$ of weight t, and the ciphertext is its syndrome $\tilde{s} = \varepsilon\tilde{H}^{\mathsf{T}}$.
- *Decryption.* To decrypt, the private syndrome $s = \varepsilon H^{\mathsf{T}} = \tilde{s}H_1^{\mathsf{T}}$ is computed and used as input for the MDPC decoder (e.g. bit-flipping or symbol flipping).

Since the decoding of LDPC and MDPC codes is probabilistic, it follows that there is always non–zero decoding failure rate (DFR). In [10], Q. Guo, T. Johansson, P. Stankovski showed that the secret key of a binary QC–MDPC cryptosystem can be recovered by exploiting decoding failures. Specifically, it was observed that certain error patterns can be decoded better or worse depending on distribution of ones in the secret key.

In order to prevent GJS attack [10], several approaches were proposed. The first one is to ensure that DFR would be sufficiently small making the complexity of the reaction attack to match the required security level. This approach is mainly based on obtaining theoretical estimates of DFR for QC–MDPC codes and choosing the appropriate cryptosystem parameters [11]–[14]. In [15] it was conjectured that replacing binary QC–MDPC codes with non–binary ones could counter GJS attack, however in [14] it was shown that GJS attack can be easily extended to the non–binary case as well.

With GJS attack being heavily based on the properties of quasi–cyclic codes, another conjecture proposed in [16] is that replacing quasi–cyclicity with different structure that would also allow compact representation of public key, could potentially mitigate reaction attacks. As a particular instance, in [16] it was proposed to replace circulant matrices in (1) with reproducible matrices. Specifically, *reproducible matrices* can be reconstructed from small set of its rows called the *signature* by applying to it certain linear transformations to obtain the remaining rows. Another possibility suggested in [14] is to replace quasi-cyclic codes with quasi-group codes, which generalize group codes (i.e., ideals in group algebras) in the same way that quasi-cyclic codes generalize cyclic codes.

In this paper, we consider cryptosystems that are based on quasi–group MDPC codes and propose a generalization of GJS reaction attack against these cryptosystems which exploits the group structure. We show that MDPC cryptosystems based on permutation–based quasi–reproducible codes with single–row signature proposed in [16], are indeed equivalent to quasi–group MDPC cryptosystems, thus implying the applicability of our attack to corresponding cryptosystems of [16] as well.

The paper is organized as follows. In Section II, we provide preliminaries on group algebras, quasi-group codes, and QG-MDPC cryptosystems. In Section III, we describe the proposed reaction attack. In Section IV, we prove that MDPC cryptosystems employing permutation-based quasi-reproducible codes with a single-row signature are equivalent to QG-MDPC cryptosystems.

II. QG–MDPC CRYPTOSYSTEMS

A. Group Algebras

Let G be a finite group and $\Bbbk$ be a field. The *group algebra of G over $\Bbbk$* is the algebra $\Bbbk G$ whose basis elements are indexed by elements of G such that the product of basis elements with indices $g, h \in G$ is the basis element with index gh. Usually, the elements of G and basis elements of $\Bbbk G$ are identified, so in that notation $\Bbbk G = \left\{ \sum_{g\in G} a_g g \mid a_g \in \Bbbk \right\}$ and

$$\left(\sum_{g\in G} a_g g\right)\left(\sum_{g\in G} b_g g\right) = \sum_{g\in G}\left(\sum_{h\in G} a_h b_{h^{-1}g}\right) g.$$

The support of $a = \sum_{g\in G} a_g g \in \Bbbk G$ is defined as $\operatorname{supp}(a) = \{g \in G \mid a_g \neq 0\}$. The Hamming weight $\operatorname{wt}(a)$ equals $|\operatorname{supp}(a)|$, the number of non-zero coefficients.

Let $G = \{g_1 = e, g_2, \ldots, g_n\}$, where e is identity, be a fixed listing of the elements of G. To avoid complicating the notation, we will identify group algebra elements $\sum_{g\in G} a_g g$ with vectors $(a_{g_1}, \ldots, a_{g_n})$. Let $h \in G$, we clearly have

$$h\left(\sum_{g\in G} a_g g\right) = \left(a_{h^{-1}g_1}, a_{h^{-1}g_2}, \ldots, a_{h^{-1}g_n}\right),$$

i.e., multiplying a group algebra element by a group element h simply permutes the coordinates of the corresponding vector.

Given $a = \sum_{g\in G} a_g g$ and $b = \sum_{g\in G} b_g g$, it can be easily verified that

$$\underbrace{a \cdot b}_{\text{in } \Bbbk G} = \underbrace{a \begin{pmatrix} g_1 b \\ g_2 b \\ \vdots \\ g_n b \end{pmatrix}}_{\text{vector by matrix}} = \underbrace{a \begin{pmatrix} b_{g_1^{-1}g_1} & b_{g_1^{-1}g_2} & \cdots & b_{g_1^{-1}g_n} \\ b_{g_2^{-1}g_1} & b_{g_2^{-1}g_2} & \cdots & b_{g_2^{-1}g_n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{g_n^{-1}g_1} & b_{g_n^{-1}g_2} & \cdots & b_{g_n^{-1}g_n} \end{pmatrix}}_{M(b)}.$$

The matrix $M(b)$ will be referred to as the *matrix representation of $b \in \Bbbk G$*, and matrices of this type will be referred to as *$\Bbbk G$-matrices*. It is easy to note that $\Bbbk G$-matrices are completely determined by the first row, with the other rows obtained by permuting the coordinates of the first row. Additionally, we have $M(a) \cdot M(b) = M(a \cdot b)$. Therefore, the map $a \mapsto M(a)$ is an isomorphism of $\Bbbk$-algebras between $\Bbbk G$ and $M(\Bbbk G) = \{M(a) \mid a \in \Bbbk G\}$.

Given an element $a = \sum_{g\in G} a_g g \in \Bbbk G$, the *$\Bbbk G$–transpose* of a is defined as

$$a^* = \sum_{g\in G} a_g g^{-1} = \sum_{g\in G} a_{g^{-1}} g.$$

It can be easily verified that $M(a^*) = M(a)^{\mathsf{T}}$ and the map $a \to a^*$ is an anti-automorphism of $\Bbbk G$, i.e., $(a+b)^* = a^* + b^*$ and $(a \cdot b)^* = b^* a^*$.

B. Quasi–G codes

Cyclic codes are known to be ideals in the (cyclic) group algebras $\mathbb{F}_q\mathbb{Z}_n$. In 1967, S. Berman proposed the notion of *group codes (G-codes)*, which are defined as *left (or right) ideals* of group algebras $\mathbb{F}_q G$ [17]. In [17] he also showed that binary Reed-Muller codes can be represented as ideals in the *dyadic group algebra* $\mathbb{F}_2\mathbb{Z}_2^r$. The additional structure provided by group algebras allows improving encoding and decoding algorithms and studying codes using powerful algebraic techniques. Furthermore, many types of efficient codes are now known to be group codes (see [18]–[21]).

Extending the concept of quasi-cyclic codes, which are $\mathbb{F}_q\mathbb{Z}_n$–submodules of $(\mathbb{F}_q\mathbb{Z}_n)^l$, *quasi-group (quasi-G, QG) codes* are defined as left (or right) $\mathbb{F}_q G$-submodules of $(\mathbb{F}_q G)^l = \mathbb{F}_q G \oplus \cdots \oplus \mathbb{F}_q G$ (see [22], [23]). This generalization mirrors the relationship between cyclic codes and quasi-cyclic codes. In essence, $C \subset (\mathbb{F}_q G)^l$ is a quasi–G code iff

1) C is a vector subspace of $\mathbb{F}_q$–vector space $(\mathbb{F}_q G)^l$;
2) for all $\lambda \in \mathbb{F}_q G$ and $c = (c^{(1)}, \ldots, c^{(l)}) \in C$ we have $\lambda c = \left(\lambda c^{(1)}, \ldots, \lambda c^{(l)}\right) \in C$.

For cryptographic applications, we are particularly interested in the quasi–G codes of the following form:

$$\mathrm{Ann}\left(h^{(1)},\dots,h^{(l)}\right)=\left\{\left(\underbrace{c^{(1)}}_{\in\mathbb{F}_qG},\dots,c^{(l)}\right)\mid\sum_{i=1}^{l}c^{(i)}h^{(i)}=0.\right\},$$

where $h^{(1)},\dots,h^{(l)}\in\mathbb{F}_qG$, which are referred to as 1–*checkable codes*. Note that, the condition $\sum_{i=1}^{l}c^{(i)}h^{(i)}=0$ can be written in the matrix form as:

$$\left(c^{(1)}\mid\cdots\mid c^{(l)}\right)\cdot H^{\mathsf{T}}=0,$$

where

$$H=\left(M(h^{(1)})^{\mathsf{T}}\mid M(h^{(2)})^{\mathsf{T}}\mid\cdots\mid M(h^{(l)})^{\mathsf{T}}\right).\qquad(3)$$

Thus, (3) is a parity–check matrix of $\mathrm{Ann}\left(h^{(1)},\dots,h^{(l)}\right)$.

Remark 1. *Let a be invertible element in $\mathbb{F}_qG$. We have*

$$\mathrm{Ann}\left(h^{(1)},\dots,h^{(l)}\right)=\mathrm{Ann}\left(h^{(1)}a,\dots,h^{(l)}a\right)$$

and

$$\left(M(h^{(1)}a)^{\mathsf{T}}\mid M(h^{(2)}a)^{\mathsf{T}}\mid\cdots\mid M(h^{(l)}a)^{\mathsf{T}}\right)=$$
$$=M(a^*)\cdot\left(M(h^{(1)})^{\mathsf{T}}\mid M(h^{(2)})^{\mathsf{T}}\mid\cdots\mid M(h^{(l)})^{\mathsf{T}}\right)$$

is another parity–check matrix of $\mathrm{Ann}\left(h^{(1)},\dots,h^{(l)}\right)$.

C. QG–MDPC cryptosystems

Below, we provide a description of code-based cryptosystem based on 1–checkable quasi–group codes with moderate density parity–check matrices (QG–MDPC codes). To simplify the description, we will stick to the case $l=2$, however the generalization to arbitrary l is straightforward. We will also use the Niederreiter form as it provides shorter keys and ciphertexts.

1) *Key generation.* Randomly sample $h^{(1)},h^{(2)}\in\mathbb{F}_qG$, s.t. $\mathrm{wt}(h^{(1)})=\mathrm{wt}(h^{(2)})=\gamma$ and $h^{(1)}$ being invertible. The *secret key* is $\left(h^{(1)},h^{(2)}\right)$, and the *public key* is
$$h=h^{(2)}\cdot\left(h^{(1)}\right)^{-1}.$$
2) *Encryption.* Let $\varepsilon^{(1)},\varepsilon^{(2)}\in\mathbb{F}_qG$, s.t. $\mathrm{wt}(\varepsilon^{(1)})+\mathrm{wt}(\varepsilon^{(2)})=t$, be a plaintext, then the ciphertext is $\tilde{s}=\varepsilon^{(1)}+\varepsilon^{(2)}h$.
3) *Decryption.* Compute $s=\tilde{s}h^{(1)}=\varepsilon^{(1)}\cdot h^{(1)}+\varepsilon^{(2)}\cdot h^{(2)}$ and try to recover the plaintext $\left(\varepsilon^{(1)},\varepsilon^{(2)}\right)$ using any suitable decoder of MDPC codes.

The parameters of the cryptosystem are G, $r=|G|$, the density parameter $\gamma\approx\sqrt{2r}/2$, and the error weight $t\approx\sqrt{2r}$.

Remark 2. *The cryptosystem above can be also described in matrix notation. Indeed, the secret key is in fact a sparse parity–check matrix of* $\mathrm{Ann}(h^{(1)},h^{(2)})$*:*

$$H_{sec}=(H_1\mid H_2)=\left(M\left(h^{(1)^*}\right)\mid M\left(h^{(2)^*}\right)\right),$$

where each column is of weight γ and each row is of weight 2γ. The public key is the systematic parity–check matrix

$$H_{pub}=H_1^{-1}\cdot H_{sec}=\left(M(1)\mid M(h^*)\right),$$

of the same code, which generally isn't sparse. Encryption is syndrome computation:

$$\tilde{s}=\left(\varepsilon^{(1)}\mid\varepsilon^{(2)}\right)H_{pub}^{\mathsf{T}},$$

and the decryption is recovering an error from the private syndrome $s=\tilde{s}\cdot H_1^{\mathsf{T}}=\left(\varepsilon^{(1)}\mid\varepsilon^{(2)}\right)\cdot H_{sec}^{\mathsf{T}}$.

Remark 3. *The cryptosystem described in this section is designed to be one-way (OW-CPA) public-key encryption (PKE). This means an adversary cannot efficiently recover the plaintext from the ciphertext, assuming the plaintext is chosen uniformly at random. However, for many practical applications, OW-CPA is not enough. The most secure encryption schemes provide indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2). In this adversary model, the attacker has access to a decryption oracle that can decipher any valid ciphertext except the one being transmitted. Thus, many practical cryptosystems are obtained by applying special conversions (see e.g. [24]) of OW-CPA PKE into IND-CCA2 key encapsulation mechanisms (KEM). An example of such a conversion applied to the QC-MDPC Niederreiter cryptosystem can be found in [25], and the generalization to the QG-MDPC cryptosystem is straightforward. Note that applying such conversions usually implies that $(\varepsilon^{(1)},\varepsilon^{(2)})$ cannot be chosen arbitrarily.*

D. Decoding of MDPC codes

This subsection explains the parallel symbol flipping decoding algorithm for regular MDPC codes over $\mathbb{F}_q$ and presents the bit-flipping decoder as a special case. A MDPC code of length n with parity-check matrix $H=(h_{i,j})$ is (γ,δ)-regular if each column and row of H has weight γ and δ respectively. Recall that codes used in QG–MDPC cryptosystems are $(\gamma,2\gamma)$–regular (see Remark 2).

Let ε be the error vector and $s=\varepsilon H^{\mathsf{T}}$ its syndrome. Since $s_i=\langle\varepsilon,h_i\rangle$, where h_i is i–th row of H, if $h_{i,j}\neq0$, then

$$s_ih_{i,j}^{-1}=\varepsilon_j+\sum_{j'\in\mathrm{supp}(h_i)}h_{i,j'}\varepsilon_{j'}.$$

Since h_i and ε are sparse, it follows that $s_ih_{i,j}^{-1}$ equals ε_j with high probability. Hence the values $s_ih_{i,j}^{-1}$ can be used to estimate the error. Indeed, given a enumeration $\mathbb{F}_q=\{\alpha_0=0,\alpha_1,\dots,\alpha_{q-1}\}$ of the field elements, the *error counters* are defined as $\sigma_{j,v}=\left|\left\{i\mid h_{i,j}\neq0,\ s_ih_{i,j}^{-1}=\alpha_v\right\}\right|$.

One can easily note that $\sigma_{j,v}$ denotes the number of parity–check equation "voting" that error value in position j is α_v. So, larger $\sigma_{j,0}$ indicates lower error probability in position j, while larger $\sigma_{j,v}$ means ε_j is more likely to be α_v. In addition, flipping $z_j\mapsto z_j-\alpha_v$ in the received noisy codeword $z=c+\varepsilon$, decreases the syndrome weight by $\sigma_{j,v}-\sigma_{j,0}$ if $\sigma_{j,v}>\sigma_{j,0}$.

Let $\sigma_j^*=\max_v\sigma_{j,v}$, and let $[\![x,y]\!]=\{z\in\mathbb{Z}\mid x\leq z\leq y\}$. In Algorithm 1, we describe the generic parallel threshold–based symbol flipping (bit flipping if $q=2$) decoder for MDPC codes. Note that there are various ways to define *computeThreshold*, in this paper for the sake of simplicity we stick to determining thresholds by $th=$

Algorithm 1: Threshold parallel flipping decoder

Input: $s = \varepsilon H^\mathsf{T}$, $H \in \mathbb{F}_q^{(n-k)\times n}$
Output: estimated error $\tilde{\varepsilon} \in \mathbb{F}_q^n$
$s' \leftarrow s$; $\tilde{\varepsilon} \leftarrow 0^n$;
for $i \in [\![1, num\_it]\!]$ **do**
  compute syndrome and the errors counters for $\varepsilon - \tilde{\varepsilon}$;
  if *syndrome is 0* **then return** $\tilde{\varepsilon}$;
  $th \leftarrow \mathrm{computeThreshold}(context)$;
  for $j \in [\![1, n]\!]$ **do**
    if $\sigma_j^* - \sigma_{j,0} \geq th$ **then**
      $v^* \leftarrow \mathrm{argmax}_v\, \sigma_{j,v}$;
      $\tilde{\varepsilon}_j \leftarrow \tilde{\varepsilon}_j + \alpha_{v^*}$;
    end
  end
end
return fail;

$\max\left\{1, \max_j\{\sigma_j^* - \sigma_{j,0}\} - \delta\right\}$ for binary codes, where δ is some small number (in our experiments $\delta = 10$), and $th = \max\left\{1, 20th\_largest_j(\sigma_j - \sigma_j^*)\right\}$ for non–binary codes. However, more advanced strategies are possible.

III. The attack

A. Description of GJS attack

In [10] Q. Guo, T. Johansson, P. Stankovski considered the binary QC–MDPC cryptosystem with secret sparse matrix $(H_1 \mid H_2)$, where $H_i = M(h^{(i)}) \in M(\mathbb{F}_2\mathbb{Z}_n)$, and showed that it is possible to recover H_1 (up to circular shift of rows). The key concept of GJS attack is based on the observation that average DFR for error patterns from the set

$$\Psi_d = \big\{(\varepsilon, f) \in (\mathbb{F}_2\mathbb{Z}_n)^2 \mid \mathrm{wt}(f) = 0, \mathrm{supp}(\varepsilon) = \{s_1, \ldots, s_t\}, \\ s_{2j} = (s_{2j-1} + d) \mod n \text{ for } j \in [\![1, t/2]\!] \big\}$$

is strongly dependent on whether d is in distance spectrum of $h^{(1)}$. The *distance spectrum* $D(h^{(1)})$ for $h^{(1)}$ is the set containing the distances between non–zero positions of $h^{(1)}$, i.e. $D(h^{(1)}) = \{d(i,j) \mid i,j \in \mathrm{supp}(h^{(1)}),\ i \neq j\}$, where $d(i,j) = \min\{(i-j) \mod n, (j-i) \mod n\}$. Specifically, in [10] it was discovered that if $d \in D(h^{(1)})$, then DFR on errors from Ψ_d is noticeably lower compared to the case $d \notin D(h^{(1)})$. Hence $D(h^{(1)})$ can be recovered by estimating DFR on Ψ_d for all $d \in [\![1, n/2]\!]$. In addition, in [10], a procedure to restore $h^{(1)}$ from its distance spectrum was described. This attack was extended to the non–binary case as well [14].

Since GJS attack exploits properties of quasi–cyclic codes, in [14], it was conjectured replacing it with quasi–group codes to avoid it. However, below we show that GJS attack can be extended to this case as well.

B. Attack against QG-MDPC cryptosystem

To extend the GJS attack [10] to the case of arbitrary group G, we propose a generalization of distance spectrum. We also show that it is possible to construct special error patterns such that by estimating DFR on that patterns it is possible to distinguish the presence of certain distance in the generalized distance spectrum of $h^{(1)^*} \in \mathbb{F}_qG$.

Let G' be a subset of G such that

1) $g, g^{-1} \in G' \iff g = g^{-1}$,
2) $G = \{g, g^{-1} \mid g \in G'\}$.

For all $g, g' \in G$ we define the G–distance $d(g, g')$ as follows

$$d(g, g') = \begin{cases} g^{-1}g', & g^{-1}g' \in G', \\ (g')^{-1}g, & (g')^{-1}g \in G'. \end{cases}$$

We see that $d(g''g, g''g') = d(g, g')$ for any $g'' \in G$. Given any $a \in \mathbb{F}_qG$, we define its G–distance spectrum as:

$$D(a) = \{d(g, g') \mid g, g' \in \mathrm{supp}(a),\ g \neq g'\}.$$

One can easily note that $D(ga) = D(a)$ for any $g \in G$ since $\mathrm{supp}(ga) = g\,\mathrm{supp}(a)$. Hence all rows in $M(a)$, $a \in \mathbb{F}_qG$, have the same G–distance spectrum. In addition, if G is abelian, then $D(a) = D(a^*)$. Denote the multiplicity of the G–distance g in $D(a)$ as $\mu_g(a) = |\{\{x, x'\} \subset \mathrm{supp}(a) \mid x \neq x',\ d(x, x') = g\}|$.

We propose constructing the set of special error patterns allowing recovering $D(h^{(1)^*})$ as follows:

$$\mathcal{E}_g = \big\{(\varepsilon, f) \in (\mathbb{F}_qG)^2 \mid \mathrm{wt}(f) = 0,\ \mathrm{supp}(\varepsilon) = \{s_1, \ldots, s_t\}, \\ s_{2j} = s_{2j-1} \cdot g \text{ for } j \in [\![1, t/2]\!] \big\}.$$

Indeed, as can be experimentally shown, there is a strong correlation between DFR on random errors from $\mathcal{E}_g$ and the presence of $g \in G$ in $D(h^{(1)^*})$ for various groups including non–abelian. Specifically, the larger $\mu_g(h^{(1)^*})$, the smaller the DFR on errors from $\mathcal{E}_g$. Thus, estimating DFR on $\mathcal{E}_g$ allows recovering $D(h^{(1)^*})$ with high probability.

In Table I, we present the results of numerical simulations showing the average DFR on $\mathcal{E}_g$ for various $\mu_g(h^{(1)^*})$. In order to validate the efficiency of our approach in the case of both abelian and non-abelian groups, for testing we used the group $\mathbb{Z}_2^r$, which is abelian, and the dihedral group

$$D_{2r} = \left\langle x, y \mid x^r = y^2 = e,\ x^i y = y x^{r-i} \right\rangle$$

which is a known non-abelian group of order $n = 2r$.

Once G–distance spectrum of $h^{(1)^*}$ is recovered, it is possible to reconstruct the support of $h^{(1)^*}$ using Algorithm 2. In essence, the algorithm recursively builds up potential supports by adding elements whose G–distances to the current partial support occur in the given spectrum. According to experiments, the resulting list contains only supports of elements of the following form: $g \cdot h^{(1)^*}$ for some $g \in G$, and $gh^{(1)}$ if G is abelian.

In the binary case, recovering the support is equivalent to recovering the element itself. So, after obtaining $g \cdot h^{(1)^*}$, we also get $h^{(1)}g^{-1}$. Therefore, the equivalent secret key can be obtained as

$$\left(h^{(1)} \cdot g^{-1},\ h \cdot h^{(1)} \cdot g^{-1}\right) = \left(h^{(1)}g^{-1}, h^{(2)}g^{-1}\right),$$

which only differs from the original key by the (weight-preserving) multiplication by $g^{-1} \in G$. Thus, the reconstructed secret key allows easily decrypting messages.

TABLE I
EXPERIMENTAL RESULTS

$\mathbb{F}_q G$	γ	t	DFR on $\mathcal{E}_g$, $\mu_g(h^{(1)^*}) = 0$	DFR on $\mathcal{E}_g$, $\mu_g(h^{(1)^*}) = 1$	DFR on $\mathcal{E}_g$, $\mu_g(h^{(1)^*}) = 2$	DFR on $\mathcal{E}_g$, $\mu_g(h^{(1)^*}) = 3$
$\mathbb{F}_2\mathbb{Z}_2^{12}$	45	74	0.2578	0.1745	0.1198	0.0847
$\mathbb{F}_4\mathbb{Z}_2^{11}$	35	80	0.0074	0.0029	0.0011	0.0003
$\mathbb{F}_2 D_{4802}$	45	92	0.1533	0.1071	0.05384	0.04602
$\mathbb{F}_4 D_{2050}$	35	80	0.0044	0.0018	0.0011	0.0006

Algorithm 2: Recovering $\mathrm{supp}(h^{(1)^*})$ from its G–distance spectrum

Input: distance spectrum $D(h^{(1)^*})$
Output: Set L of possible supports of $h^{(1)^*}$
Procedure $F(D, S, U)$
  Input: distance spectrum D, partial support S, potential support elements U
  if $|S| = \gamma$ **then** $L \leftarrow L \cup \{S\}$;
  else
    forall $h \in U$ **do**
      $U \leftarrow U \setminus \{h\}$;
      if $d(h, s) \in D$ *for all* $s \in S$ **then**
        make recursive call $F(D, S \cup \{h\}, U)$;
      end
    end
  end
$L \leftarrow \varnothing$; $\xi \leftarrow$ any element from $D(h^{(1)^*})$;
$F\left(D(h^{(1)}), \{e, \xi\}, G \setminus \{e, \xi\}\right)$;
return L;

In the non-binary case, recovering $\mathrm{supp}(h^{(1)^*})$ does not reveal $h^{(1)^*}$ itself, so this method does not directly apply. However, the secret key can still be efficiently recovered using the approach described in [14].

Since in IND–CCA2–converted MDPC cryptosystems, the adversary is usually limited in choosing arbitrary errors in the encryption step, the proposed attack can be modified to cover this versions using sampling and collecting error pattern with desired distance properties. Such a modification of the attack against QC–MDPC cryptosystems was described in [10], and the extension of our attack can be done analogously.

IV. APPLICABILITY OF THE ATTACK TO QUASI–REPRODUCIBLE CODES

One can easily notice that the parity–check matrix of codes used in QG–MDPC cryptosystems is built from $\mathbb{F}_q G$–matrix blocks (see Remark 2). Moreover, as previously noted, $\mathbb{F}_q G$–matrices have the following properties:

(1) each matrix is characterized by only the first row, and the remaining rows are obtained from the first one by applying certain set permutations;
(2) the set of such matrices form a ring.

Below, we show that the converse is also true, i.e., if there is a set of matrices, satisfying (1)–(2), then it coincides with $M(\mathbb{F}_q G)$ for some group G.

Let S_n denote the group of permutations on $[\![1, n]\!]$. Given $\pi \in S_n$, an action of π on a vector $x = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ is defined as $\pi.x = \left(x_{\pi^{-1}(1)}, \ldots, x_{\pi^{-1}(n)}\right)$, i.e., π acts on standard basis $\{e_1, \ldots, e_n\}$ of $\mathbb{F}_q^n$ by $\pi.e_i = e_{\pi(i)}$. In addition, it is also possible to map π to the permutation matrix $P(\pi)$, where positions indexed by $(i, \pi(i))$ are ones and all the remaining positions are zeros. One can easily note that

$$\pi.x = x \cdot P(\pi), \quad P(\pi) \cdot x^{\mathsf{T}} = \left(\pi^{-1}.x\right)^{\mathsf{T}}.$$

The map $\pi \mapsto P(\pi)$ is an anti–monomorphism from S_n to $\mathrm{GL}_n(\mathbb{F}_q)$, i.e. $P(\pi\tilde{\pi}) = P(\tilde{\pi}) \cdot P(\pi)$.

Theorem 1. *Let* $\mathcal{F} = \{\pi_1 = e, \pi_2, \ldots, \pi_n\} \subset S_n$. *Define*

$$\mathcal{M}_{\mathcal{F}} = \left\{ \begin{pmatrix} \pi_1.x \\ \pi_2.x \\ \vdots \\ \pi_n.x \end{pmatrix} \in \mathbb{F}_q^{n \times n} \;\middle|\; x \in \mathbb{F}_q^n \right\}.$$

If $A \cdot B \in \mathcal{M}_{\mathcal{F}}$ *for any* $A, B \in \mathcal{M}_{\mathcal{F}}$, *i.e.* $\mathcal{M}_{\mathcal{F}}$ *is a ring, then* $\mathcal{F}$ *is a subgroup of* S_n, *s.t.* $\pi_i(1) = i$, *and* $\mathcal{M}_{\mathcal{F}} = M(\mathbb{F}_q \mathcal{F})$.

Proof. Theorem 3.9 of [16] implies that if $\mathcal{M}_{\mathcal{F}}$ is closed under multiplication, then $P(\pi_i) \cdot A = A \cdot P(\pi_i)$ for any $A \in \mathcal{M}_{\mathcal{F}}$ and $i \in [\![1, n]\!]$. Let $A_i = \pi_i.a$, where $a \in \mathbb{F}_q^n$, denote i–th row of A. Since

$$P(\pi_i) \cdot A = \begin{pmatrix} A_{\pi_i(1)} \\ \vdots \\ A_{\pi_i(n)} \end{pmatrix} = \begin{pmatrix} \pi_{\pi_i(1)}.a \\ \vdots \\ \pi_{\pi_i(n)}.a \end{pmatrix},$$

$$A \cdot P(\pi_i) = \begin{pmatrix} aP(\pi_1)P(\pi_i) \\ \vdots \\ aP(\pi_n)P(\pi_i) \end{pmatrix} = \begin{pmatrix} (\pi_i\pi_1).a \\ \vdots \\ (\pi_i\pi_n).a \end{pmatrix},$$

it follows that $\pi_i\pi_j = \pi_{\pi_i(j)}$, implying that $\mathcal{F}$ is closed under permutation composition. With $\mathcal{F}$ being finite, it is a subgroup of S_n by the finite subgroup test. Moreover, substituting $j = 1$ into $\pi_i\pi_j = \pi_{\pi_i(j)}$ gives $\pi_i = \pi_{\pi_i(1)}$ and thus $\pi_i(1) = i$.

Furthermore, since $A_{i,j} = a_{\pi_i^{-1}(j)}$, it follows that $A_{i,j} = a_{(\pi_i^{-1}\pi_j)(1)}$. As there is a one-to-one correspondence $\pi \mapsto \pi(1)$ between $\mathcal{F}$ and $[\![1, n]\!]$, we obtain that $A = M(\overline{a})$, where $\overline{a} = \sum_{i=1}^n a_i\pi_i \in \mathbb{F}_q\mathcal{F}$, i.e., A is a $\mathbb{F}_q\mathcal{F}$–matrix. □

Corollary 1. *Any code with parity–check matrix of the form*

$$(H_1 \mid \cdots \mid H_m), \quad H_i \in \mathcal{M}_{\mathcal{F}},$$

where $\mathcal{M}_{\mathcal{F}}$ *satisfies conditions of Theorem 1, is a* 1*–checkable quasi–*$\mathcal{F}$ *code.*

Note that in Section 4.3 of [16], it was proposed to replace circulant blocks in QC–MDPC cryptosystems with matrices from $\mathcal{M}_{\mathcal{F}}$. It was also noted that to obtain Niedirreiter–like cryptosystem with compact keys the set $\mathcal{M}_{\mathcal{F}}$ has to be a ring with identity. In addition, in [16] it was conjectured that replacing quasi–cyclic structure with this more general one would mitigate GJS attack. However, Theorem 1 and Corollary 1 imply that the obtained cryptosystems are instances of QG–MDPC cryptosystems, so the reaction attack proposed in this paper is also applicable to many cryptosystems with compact keys based on quasi–reproducible MDPC codes proposed in [16].

V. Conclusion

In this paper, we considered cryptosystems based on quasi–group MDPC codes, which generilize QC–MDPC cryptosystems. We also showed that it is possible to extend an approach of [10] to build a reaction attack against this cryptosystems. In addition, we proved that quasi-reproducible codes that can be used in Niederreiter–like cryptosystems and that have parity-check matrices defined only by their first rows with the remaining ones obtained by permutations are in fact QG–codes. Thus, they are also vulnerable to the proposed attack. It should be noted that other classes of binary reproducible MDPC code–based cryptosystems are likely to have larger public keys and be less efficient compared to QC–MDPC cryptosystems with theoretically estimated DFR [13].

References

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[2] D. J. Bernstein and T. Lange, "Post-quantum cryptography," *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

[3] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *Coding Thv*, vol. 4244, pp. 114–116, 1978.

[4] D. J. Bernstein, T. Chou, T. Lange, I. von Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier *et al.*, "Classic McEliece: conservative code-based cryptography," *NIST submissions*, vol. 1, no. 1, pp. 1–25, 2017.

[5] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta *et al.*, "Status report on the third round of the NIST post-quantum cryptography standardization process," *US Department of Commerce, NIST*, 2022.

[6] V. Weger, N. Gassner, and J. Rosenthal, "A survey on code-based cryptography," *arXiv preprint arXiv:2201.07119*, 2022.

[7] P. Gaborit and J.-C. Deneuville, "Code-based cryptography," in *Concise Encyclopedia of Coding Theory*, W. C. Huffman, J.-L. Kim, and P. Solé, Eds. Chapman and Hall/CRC, 3 2021.

[8] R. Gallager, "Low-density parity-check codes," *IRE Transactions on information theory*, vol. 8, no. 1, pp. 21–28, 1962.

[9] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. Barreto, "MDPC-McEliece: new McEliece variants from moderate density parity-check codes," in *2013 IEEE international symposium on information theory*. IEEE, 2013, pp. 2069–2073.

[10] Q. Guo, T. Johansson, and P. S. Wagner, "A key recovery reaction attack on QC-MDPC," *IEEE Transactions on Information Theory*, vol. 65, pp. 1845–1861, 3 2019.

[11] J.-P. Tillich, "The decoding failure probability of MDPC codes," in *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2018, pp. 941–945.

[12] P. Santini, M. Battaglioni, M. Baldi, and F. Chiaraluce, "Hard-decision iterative decoding of LDPC codes with bounded error rate," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[13] ——, "Analysis of the error correction capability of LDPC and MDPC codes under parallel bit-flipping decoding and application to cryptography," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4648–4660, 2020.

[14] K. Vedenev and Y. Kosolapov, "Theoretical analysis of decoding failure rate of non-binary QC-MDPC codes," in *Code-Based Cryptography - 11th International Workshop CBCrypto 2023*, ser. Lecture Notes in Computer Science, A. Esser and P. Santini, Eds., vol. 14311. Springer Nature Switzerland, 2023, (to appear, eprint: https://eprint.iacr.org/2023/1224).

[15] M. Baldi, G. Cancellieri, F. Chiaraluce, E. Persichetti, and P. Santini, "Using non-binary LDPC and MDPC codes in the mceliece cryptosystem," in *2019 AEIT International Annual Conference (AEIT)*. IEEE, 2019, pp. 1–6.

[16] P. Santini, E. Persichetti, and M. Baldi, "Reproducible families of codes and cryptographic applications," *Journal of Mathematical Cryptology*, vol. 16, no. 1, pp. 20–48, 2021.

[17] S. D. Berman, "On the theory of group codes," *Cybernetics*, vol. 3, pp. 25–31, 1967.

[18] A. Kelarev and P. Solé, "Error-correcting codes as ideals in group rings," *Contemporary Mathematics*, vol. 273, pp. 11–18, 2001.

[19] M. Borello and W. Willems, "Group codes over fields are asymptotically good," *Finite Fields and Their Applications*, vol. 68, p. 101738, 12 2020.

[20] W. Willems, "Codes in group algebras," in *Concise Encyclopedia of Coding Theory*, W. C. Huffman, J.-L. Kim, and P. Solé, Eds. Chapman and Hall/CRC, 3 2021.

[21] L. P. Natarajan and P. Krishnan, "Berman codes: A generalization of Reed-Muller codes that achieve BEC capacity," *IEEE Transactions on Information Theory*, pp. 1–1, 2023.

[22] S. T. Dougherty, J. Gildea, R. Taylor, and A. Tylyshchak, "Group rings, G-codes and constructions of self-dual and formally self-dual codes," *Designs, Codes and Cryptography*, vol. 86, pp. 2115–2138, 2018.

[23] M. Borello and W. Willems, "On the algebraic structure of quasi-group codes," *Journal of Algebra and its Applications*, p. 2350222, 2022.

[24] D. Hofheinz, K. Hövelmanns, and E. Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation," in *Theory of Cryptography Conference*. Springer, 2017, pp. 341–371.

[25] N. Aragon, P. S. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. A. Melchor *et al.*, "BIKE: bit flipping key encapsulation," 2017. [Online]. Available: bikesuite.org