Exploring Decryption Failures of BIKE: New Class of Weak Keys and Key Recovery Attacks

Tianrui Wang[1], Anyu Wang[2,3,4(✉)], and Xiaoyun Wang[2,3,4,5,6]

[1] Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China
wangtr22@mails.tsinghua.edu.cn
[2] Institute for Advanced Study, BNRist, Tsinghua University, Beijing, China
anyuwang@tsinghua.edu.cn, xiaoyunwang@tsinghua.edu.cn
[3] Zhongguancun Laboratory, Beijing, China
[4] National Financial Cryptography Research Center, Beijing, China
[5] Shandong Institute of Blockchain, Jinan, China
[6] Key Laboratory of Cryptologic Technology and Information Security (Ministry of Education), School of Cyber Science and Technology, Shandong University, Qingdao, China

Abstract. Code-based cryptography has received a lot of attention recently because it is considered secure under quantum computing. Among them, the QC-MDPC based scheme is one of the most promising due to its excellent performance. QC-MDPC based schemes are usually subject to a small rate of decryption failure, which can leak information about the secret key. This raises two crucial problems: how to accurately estimate the decryption failure rate and how to use the failure information to recover the secret key. However, the two problems are challenging due to the difficulty of geometrically characterizing the bit-flipping decoder employed in QC-MDPC, such as using decoding radius.

In this work, we introduce the gathering property and show it is strongly connected with the decryption failure rate of QC-MDPC. Based on this property, we present two results for QC-MDPC based schemes. The first is a new construction of weak keys obtained by extending the keys that have gathering property via ring isomorphism. For the set of weak keys, we present a rigorous analysis of the probability, as well as experimental simulation of the decryption failure rates. Considering BIKE's parameter set targeting 128-bit security, our result eventually indicates that the average decryption failure rate is lower bounded by $\mathrm{DFR}_{\mathrm{avg}} \geq 2^{-116.61}$. The second entails two key recovery attacks against CCA secure QC-MDPC schemes using decryption failures in a multi-target setting. The two attacks consider whether or not it is allowed to reuse ciphertexts respectively. In both cases, we show the decryption failures can be used to identify whether a target's secret key satisfies the gathering property. Then using the gathering property as an extra information, we present a modified information set decoding algorithm that efficiently retrieves the target's secret key. For BIKE's parameter set targeting 128-bit security, we show a key recovery attack with complexity $2^{116.61}$ can be mounted if ciphertexts reusing is not permitted, and the complexity can be reduced to $2^{98.77}$ when ciphertexts reusing is permitted.

Keywords: Post-quantum cryptography · Code-based cryptography · Decryption failure · BIKE · QC-MDPC · Information set decoding

1 Introduction

Shor's algorithm [46] can solve the problems of integer factorization and discrete logarithm in quantum polynomial time. Then once large-scale quantum computer that implements Shor's algorithm becomes a reality, traditional public-key systems based on factorization and discrete logarithm will run the risk of being broken. As a result, developing public-key systems that can withstand quantum attacks has become a pressing concern. In 2016, NIST (*National Institute of Standards and Technology*) was motivated to start a process of standardizing post-quantum public-key cryptographic algorithms [1]. In this process, code-based cryptography plays an important role.

Code-based cryptography can be traced back to the invention of McEliece public-key encryption scheme [39] and its variation Niederreiter scheme [42]. These schemes are built on Goppa codes, and their security can be reduced to the hardness of decoding binary linear codes [6]. Other types of codes can also be used to construct public key encryption schemes, and those based on QC-MDPC (*quasi-cyclic moderate density parity check*) codes are in an competitive class in terms of efficiency and bandwidth [40]. BIKE [3] is a representative QC-MDPC based scheme that has advanced to the fourth round of the NIST standardization process [2].

QC-MDPC based schemes are typically subject to decryption failures, which means that even when the protocol is correctly executed, it is still possible for the decryption to fail to recover the intended message. It is well known that decryption failures can leak information about the secret key, and different types of decryption failure attacks have been proposed for various lattice-based and code-based schemes. One type of such attacks was introduced by Jaulmes and Joux in [35] and extended in [34,27], which is against CPA (*chosen plaintext attack*) secure schemes and recovers the secret key by choosing certain ciphertexts that fail based on characteristics of the secret key. Another type of decryption failure attack can be carried out against CCA (*chosen ciphertext attack*) secure schemes, which is typically mounted in three stages: a precomputation stage in which special ciphertexts are generated randomly, a decryption stage in which the ciphertexts are submitted for decryption and some decryption failures are observed, and a key recovery stage in which the secret key is retrieved based on a statistical analysis of the decryption failures [14]. In [29], Guo et al. presented such an attack against the CCA secure QC-MDPC based scheme in [40] by using the "distance spectrum" to retrieve the secret key. Later, decryption failure attacks against other code-based schemes were also proposed for, e.g., HQC [28], QC-LDPC [24] and LRPC [4]. In [15,13], D'Anvers et al. investigated the decryption failure attacks for LWE-based schemes and proposed a technique called "directed failure boosting", which significantly speeds up the ciphertext search when several decoding failures have already been obtained. In addition to focusing solely on how to recover the secret key, Bindel and Schanck [7] demonstrated that successful decryption can also be utilized to speed up the search for ciphertexts. D'Anvers et al. [16] used the correlation of individual mistake bits to demonstrate that the decryption failure rate for specific algorithms could be underestimated.

To protect a scheme against decryption failure attacks, a natural solution is to reduce the probability of decoding failure so that it is unlikely to occur for an allowed number of decryptions. In QC-MDPC based schemes, the bit-flipping decoding algorithm [26] is employed to handle the errors involved in the decryption procedure. Since the bit-flipping algorithm is originally developed to decode LDPC (*Low Density Parity Check*) codes, numerous efforts have been made to enhance it to handle slightly denser errors in MDPC codes [40,31,37,11,10,9]. Another major problem is how to accurately estimate the failure rate of QC-MDPC based schemes. In [47], an asymptotic upper bound on the decoding failure rate is derived for MDPC codes. Sendrier and Vasseur [44] propose a framework to estimate the failure rate by adopting a Markov chain model. On the other hand, weak keys that result in higher decryption failure rate in QC-MDPC based schemes were also studied [45,48].

In BIKE, the Black-Gray-Flip (BGF) decoder [19] is adopted, and the decryption failure rate is believed to be low enough to make the scheme δ-correct, that is, the decryption failure rate δ is less than $2^{-\lambda}$ for λ-bit security. Under this condition, the CCA security of BIKE can be guaranteed via the Fujisaki-Okamoto transformation [25,32].

1.1 Our results

In this work, we reinvestigate the decryption failure rate for QC-MDPC based schemes by introducing the *gathering property.* $(y_0, y_1) \in \mathcal{R}^2$ is said to satisfy the (m, ϵ)-gathering property if there are $(w_H(y_0) - \epsilon)$ 1's of y_0 gathering in some m consecutive positions (see Fig. 1), where $\mathcal{R} = \mathbb{F}_2[x]/(x^r - 1)$. The gathering property exhibits a strong connection with the decryption failure rate of QC-MDPC. Experimental result demonstrates that when both the secret keys and the errors satisfy the gathering property, the decryption failure rate is significantly higher than the average. Based on the gathering property, we are able to give the following two results on the decryption failure rate for QC-MDPC based schemes.

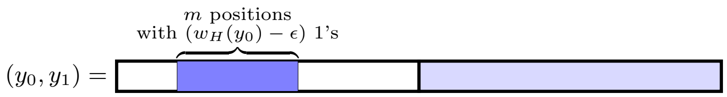

Fig. 1. An illustration of the gathering property.

A New Construction of Weak Keys. Our first contribution is a new construction of weak keys for QC-MDPC based schemes. Let $\mathrm{K}_{m,\epsilon}(w)$ be the set of secret keys satisfying the (m,ϵ)-gathering property. Through experiments, it can be proved that the decryption failure rate for the keys drawn from $\mathrm{K}_{m,\epsilon}(w)$ is higher than the average. Furthermore, $\mathrm{K}_{m,\epsilon}(w)$ can be extended to a larger set of weak keys $\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)$ by using the ring isomorphisms of $\mathcal{R}$.

We provide a rigorous approach to calculate the probability that a random key is in $\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)$, as well as experimental simulations of the decryption failure rates for keys drawing from $\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)$. These directly lead to a lower bound on the average decryption failure rate for BIKE's parameter set targeting 128-bit security, i.e.,

$$\mathrm{DFR}_{\mathrm{avg}} \geq 2^{-116.61} \quad . \tag{1}$$

Taking the simulation error into account, we can still conclude that $\mathrm{DFR}_{\mathrm{avg}} \geq 2^{-117.77}$ at 95% confidence level by using the normal approximation framework.

Key Recovery Attack. Our second contribution entails two key recovery attacks using decryption failures against CCA secure QC-MDPC schemes. The attacks are carried out in the multi-target setting, i.e., numerous targets are queried with the goal of recovering the secret key for at least one of these targets.

First, we consider an attack model that assumes multi-target protection where ciphertexts reusing is not allowed. For each target, the attacker randomly generates a set of ciphertexts, and then queries the target's decryption oracle to decrypt these ciphertexts. Once a decryption failure occurs for a target T, the attacker has an advantage of identifying whether T's secret key belongs the set of weak keys $\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)$, i.e., there exists an isomorphism of $\mathcal{R}$ such that the secret key satisfies the (m,ϵ)-gathering property after the action of isomorphism. Then we propose a *modified information set decoding* algorithm, which can efficiently recover T's secret key from the public key by using the gathering property as extra information. On the other hand, the current version of BIKE does not offer multi-target protection, prompting us to explore an attack model where ciphertexts reusing is permissible. In this model, the attacker first constructs a set of ciphertexts such that the errors satisfying the (m,ϵ)-gathering property. Then the attacker queries each target's decryption oracle to decrypt the same set of ciphertexts. For a target T that has a decryption failure, the attacker has an advantage of identifying whether T's secret key satisfies the (m,ϵ)-gathering property and can thus be efficiently recovered by the modified information set decoding algorithm.

Again we focus on BIKE's parameter set targeting 128-bit security. For the attack model that ciphertexts reusing is not allowed, we show that a key recovery attack can be performed with complexity

$$C_{\mathrm{total}} = 2^{116.61}. \tag{2}$$

Furthermore, when considering the attack model allowing for ciphertexts reusing, we show that the complexity of the key recovery attack can be reduced to $2^{98.77}$. Table 1 lists more detailed attack complexity of our attacks.

The source code for our experiments on decryption failure rates is available at `https://github.com/1234wangtr/BIKE_weakey`.

Table 1. The time complexity of the key recovery attacks against BIKE's parameter set targeting 128-bit security. The complexity of identifying failures is primarily determined by the complexity of accessing decryption oracles for all targets. The complexity of the key recovery step is determined by the total time complexity associated with all calls made to the ISD algorithm. The preprocessing complexity is associated with constructing the set of ciphertexts that satisfy the (m, ϵ)-gathering property.

	Attack model *without* ciphertexts reusing	Attack model *with* ciphertexts reusing
Total Complexity	$\mathbf{2^{116.61}}$	$\mathbf{2^{98.77}}$
Number of Targets	$2^{87.28}$	$2^{76.69}$
Queries per Target	$2^{29.33}$	$2^{22.08}$
Complexity of Identifying Failures	$2^{116.61}$	$2^{98.77}$
Complexity of the Key Recovery Step	$2^{111.96}$	$2^{94.81}$
Complexity of Preprocessing	—	$2^{97.66}$

1.2 Related Works

Guo et al's Attack on MDPC. In [29], Guo et al. find a strong correlation between the decryption failure rate and the secret key's distance spectrum, which is defined to be the set of distances between any two 1's in the secret key. From decryption failures, one can collect enough information about the secret key's distance spectrum, and then recover the secret key from the distance spectrum by the algorithm given in [29].

Weak Keys in QC-MDPC Schemes. In [18,45], the authors figure out that there exist weak secret keys in QC-MDPC for which the decryption failure rates are higher than the average. Vassuer [49] gives a classification of known weak keys in QC-MDPC, and presents simulation results for BIKE with parameter set targeting 128-bit security. In fact, lower bounds on the average decryption failure rate can be deduced via the formula $\mathrm{DFR}_{\mathrm{avg}} \geq \frac{|\mathrm{W}|}{|\mathrm{K}|} \cdot \mathrm{DFR}_{\mathrm{W}}$ where K is the set of keys and W is a set of weak keys, $\mathrm{DFR}_{\mathrm{W}}$ represents the decryption failure rate for the keys drawn uniformly from W. However, the lower bounds provided in [49] are all below 2^{-128}, which has no effect on BIKE's CCA security [2].

1.3 Summary of Our Work

As noted in [2,3], BIKE's security under decryption failures raises two critical questions: whether there exist weak keys that can affect the current estimate of the average decryption failure rate, and whether it is possible to launch a successful attack by utilizing the decryption failures. Our work confirms positive answers to both questions.

Weak Keys & Average DFR. We present a set of weak keys that impacts BIKE's current estimate of the average decryption failure rate for the first time. For BIKE's parameter set targeting 128-bit security, our results indicate that the average decryption failure rate is higher than 2^{-128}. This is concerning because the CCA security of BIKE relies on the δ-correctness assumption, which our findings suggest is not fully established.

Decryption Failure Attacks. Applying previous decryption failure attacks, such as [29], to BIKE faces a major challenge due to the low average decryption failure rate. Our new attack framework provides a solution by utilizing the gathering property. The gathering property exhibits a significant impact on the decryption failure rate, making our attack framework effective. Moreover, our framework permits a rigorous derivation or experimental confirmation of the relevant probability and decryption failure rates. This enables us to calculate the explicit attack complexity, and can give some insight into the concrete security of BIKE under decryption failure attacks.

1.4 Organizations

Section 2 introduces some preliminary concepts. In Section 3, we define the gathering property and provide experimental results on the decryption failure rate, assuming both the secret key and error satisfy this property. Section 4 presents a new construction of weak keys and derives lower bounds on the average decryption failure rate based on these weak keys. Section 5 and Section 6 describe key recovery attacks leveraging the gathering property, while Section 7 offers concluding remarks.

2 Preliminary

The following notations will be used in this paper.

- For a vector $\mathbf{y} = (y_0, \cdots, y_{n-1}) \in \mathbb{F}_2^n$, denote $w_H(\mathbf{y})$ to be the Hamming weight of $\mathbf{y}$, denote $\mathrm{Supp}(\mathbf{y})$ to be the support of $\mathbf{y}$, and denote $\mathbf{y}^{[a,b)} := (y_a, y_{a+1}, \cdots, y_{b-1})$, where the subscripts are taken mod n.
- Let $\mathcal{R} := \mathbb{F}_2[x]/(x^r - 1)$. An element in the ring $\mathcal{R}$ is represented in a polynomial form, i.e, $y = y_0 + y_1 x + \cdots + y_{r-1} x^{r-1}$, and the bold-case letter

$\mathbf{y} = (y_0, \cdots, y_{r-1})$ will be used to denote the coefficient vector of y. A circulant matrix corresponding the coefficients of y can be defined as

$$\mathtt{rot}(\mathbf{y}) = \begin{pmatrix} y_0 & y_{r-1} & \cdots & y_1 \\ y_1 & y_0 & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_{r-1} & y_{r-2} & \cdots & y_0 \end{pmatrix}. \tag{3}$$

Then for any $y, z \in \mathcal{R}$, the coefficient vector of yz equals to $\mathtt{rot}(\mathbf{y}) \cdot \mathbf{z}$.

- Suppose i is co-prime to r, then the map

$$\phi_i : y(x) \to y(x^i) \tag{4}$$

defines an isomorphism of $\mathcal{R}$ to $\mathcal{R}$. Particularly, this isomorphism preserves the Hamming weight.

- Denote
  - $\mathrm{K}(w) := \{(h_0, h_1) \in \mathcal{R}^2 | w_H(h_0) = w_H(h_1) = w/2\}$,
  - $\mathrm{E}(t) := \{(e_0, e_1) \in \mathcal{R}^2 | w_H(e_0) + w_H(e_1) = t\}$,
  - $\mathrm{E}(t_0, t_1) := \{(e_0, e_1) \in \mathcal{R}^2 | w_H(e_0) = t_0, w_H(e_1) = t_1\}$.

  Let $\mathrm{K}_{m,\epsilon}(w)$ and $\mathrm{E}_{m,\epsilon}(t_0, t_1)$ to be subsets of $\mathrm{K}(w)$ and $\mathrm{E}(t_0, t_1)$ respectively, such that their elements satisfy the (m, ϵ)-gathering property in Definition 1. Denote

$$p_{m,\epsilon} := \frac{|\mathrm{K}_{m,\epsilon}(w)|}{|\mathrm{K}(w)|} \text{ and } q_{m,\epsilon} := \frac{|\mathrm{E}_{m,\epsilon}(t/2, t/2)|}{|\mathrm{E}(t)|}. \tag{5}$$

2.1 Estimate of the Probability from the Frequency

How to estimate the probability from the frequency is a basic problem in statistics. In this paper, we mainly focus on the simulation of the decryption failures, which can be treated as a Bernoulli trial. Suppose we repeat the decryption for N times and find F failures while the actual decryption failure rate is p. Then the ratio F/N is an estimate of p. In the framework of normal approximation, the standard deviation of this estimate is

$$\sigma = \frac{\sqrt{F(N-F)}}{N\sqrt{N}} \approx \frac{\sqrt{F}}{N} \text{ for } F \ll N. \tag{6}$$

Then it has

$$\Pr[F/N - 2\sigma < p < F/N + 2\sigma] \approx 95\%, \tag{7}$$

and the confidence level will increase to 99.7% if 3σ is adopted in (7).

2.2 BIKE

In this work, we use BIKE to demonstrate our results. The gathering property and the key recovery attack can be directly applied to the QC-MDPC based schemes such as [41]. BIKE is built by first constructing a PKE (*public-key* *encryption*) using the Niederreiter framework, and then obtaining a KEM (*key encapsulation encapsulation*) following the method proposed in [20]. Let $n = 2r, w = 2v = O(\sqrt{n}), t = O(\sqrt{n})$ be a set of parameters, and let $\mathtt{H}$, $\mathtt{L}$, $\mathtt{K}$ be hash functions with proper outputs. Then BIKE KEM can be described as follows.

- $\mathtt{KeyGen}$ ():
  - Randomly generate $h_0, h_1 \in \mathcal{R}$ such that $w_H(h_0) = w_H(h_1) = w/2$.
  - Compute $h = h_1 h_0^{-1} \in \mathcal{R}$.
  - Output (h_0, h_1, σ) as the secret key, and h as the public key.
- $\mathtt{Encaps}$ (h):
  - Randomly choose $m \in \{0,1\}^{256}$.
  - Compute $(e_0, e_1) = \mathtt{H}(m) \in \mathcal{R}^2$ such that $w_H(e_0) + w_H(e_1) = t$.
  - Output the ciphertext $c = (e_0 + e_1 h, m \oplus \mathtt{L}(e_0, e_1))$, and the shared secret $\mathcal{K} = \mathtt{K}(m, c)$.
- $\mathtt{Decaps}$ $((h_0, h_1, \sigma), c)$:
  - Compute $e' = \mathtt{decoder}(c_0 h_0, h_0, h_1) \in \mathcal{R}^2$.
  - Compute $m' = c_1 \oplus L(e')$.
  - If $e' = \mathtt{H}(m')$ then output $\mathtt{K}(m', c)$, else output $\mathtt{K}(\sigma, c)$.

The `decoder` in BIKE is the Black-Gray-Flip (BGF) algorithm proposed in [19]. BIKE provides three classes of parameters targeting 128-bit, 192-bit and 256-bit security respectively, which are listed in Table 2.

Table 2. BIKE parameter sets.

Security Level	r	w	t	Decryption Failure Rate
128-bit	12323	142	134	2^{-128}
192-bit	24659	206	199	2^{-192}
256-bit	40973	274	264	2^{-256}

2.3 The Bit-Flipping Algorithm

The bit-flipping algorithm is initially introduced in [26] for the decoding of LDPC codes. Taking inputs as an LDPC matrix $\mathbf{H} \in \mathbb{F}_2^{m \times n}$ and a syndrome vector $\mathbf{s} \in \mathbb{F}_2^m$, it iteratively finds the error vector $\mathbf{e} \in \mathbb{F}_2^n$ s.t. $\mathbf{He} = \mathbf{s}$ as follows. The algorithm starts with a zero vector $\mathbf{e} = \mathbf{0}$. In each iteration, it computes the number of *unsatisfied parity checks*

$$\mathtt{UPC}(\mathbf{e}, i) \coloneqq |\mathrm{Supp}(\mathbf{He} + \mathbf{s}) \cap \mathrm{Supp}(\mathbf{h_i})| \tag{8}$$

for each position $i \in [0, n-1]$, where $\mathbf{h_i}$ is the i-th column of $\mathbf{H}$, and flips the i-th position of $\mathbf{e}$ if $\mathtt{UPC}(\mathbf{e}, i)$ exceeds a pre-set threshold τ. The algorithm terminates when the maximum number of iterations NIter is achieved. We refer to Algorithm 1 for the details.

Algorithm 1: The Bit Flipping Algorithm

```
Input: $\mathbf{H} \in \mathbb{F}_2^{m \times n}, \mathbf{s} \in \mathbb{F}_2^m$
Output: $\mathbf{e} \in \mathbb{F}_2^n$ s.t. $\mathbf{He} = \mathbf{s}$
$\mathbf{e} = \mathbf{0}$
while $i <$ NIter do
    for $j$ from 0 to $n - 1$ do
        if $\texttt{UPC}(\mathbf{e}, j) \geq \tau$ then
            Flip the $j$-th position of $\mathbf{e}$
        end if
    end for
    $i = i + 1$
end while
return $\mathbf{e}$
```

When dealing with QC-MDPC codes, the bit-flipping algorithm takes $s, h_0, h_1 \in \mathcal{R}$ as inputs and finds the error vector $e_0, e_1 \in \mathcal{R}$ by solving

$$\begin{bmatrix}\texttt{rot}(\mathbf{h}_0) \; \texttt{rot}(\mathbf{h}_1)\end{bmatrix} \begin{bmatrix}\mathbf{e}_0 \\ \mathbf{e}_1\end{bmatrix} = \mathbf{s} \tag{9}$$

via Algorithm 1. Several optimizations have been proposed for MDPC codes, such as improved selection of the threshold and flipping the bits in parallel. The BGF algorithm in BIKE adopts a BG iteration, in which the positions with UPCs that exceed a high threshold (black) are flipped and checked first, and the positions with UPCs that are close but below a high threshold (gray) are flipped afterwards. Then some standard bit-flipping iterations are performed. The complete BGF algorithm can be found in Appendix D.

3 The Gathering Property for QC-MDPC

In this section we focus on the decryption failure rate of QC-MDPC based schemes when both the secret key (h_0, h_1) and the error (e_0, e_1) satisfy the *gathering property* defined as below.

Definition 1 (gathering property). *Let $m < r$ be a positive integer and let $\epsilon \geq 0$ be a small integer, then $(y_0, y_1) \in \mathcal{R}^2$ is said to have the (m, ϵ)-gathering property if there exists an integer a such that*

$$w_H(\mathbf{y}_0^{[a, a+m)}) = w_H(\mathbf{y}_0) - \epsilon. \tag{10}$$

The gathering property means that all but ϵ 1's of y_0 gather in some m consecutive positions (in the cyclic sense). We note that there is no requirement on the right side element y_1. In this paper, we are particularly interested in the case $\epsilon = 0, 1$.

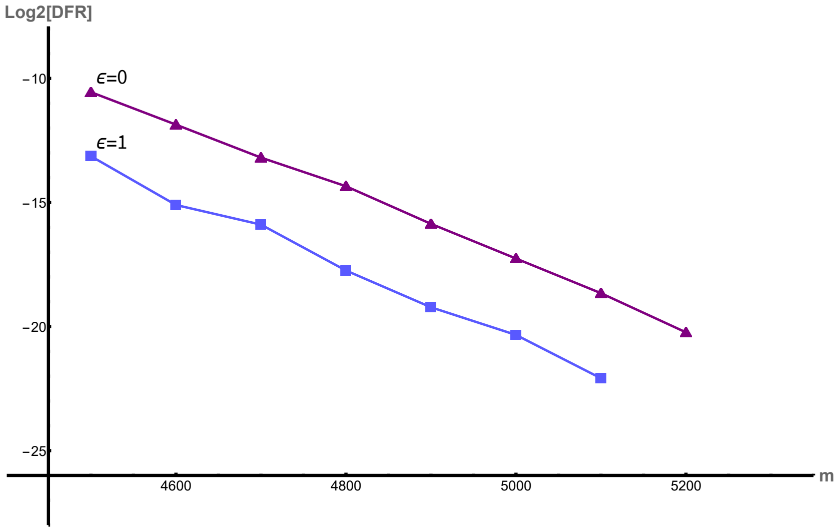

Fig. 2. The decryption failure rates for (h_0, h_1) and (e_0, e_1) satisfy the (m, ϵ)-gathering property.

3.1 The Frequency of Decryption Failures

The gathering property has a significant impact on the decryption failure rates of QC-MDPC based schemes. We experimentally simulate the decryption failure rates for BIKE with BGF algorithm and parameter set targeting 128-bit security. In the experiment, we sample (h_0, h_1) from $\mathrm{K}_{m,\epsilon}(w)$ and sample (e_0, e_1) from $\mathrm{E}_{m,\epsilon}(t/2, t/2)$ uniformly at random, where the specific sampling method is discussed in Section 3.1. By taking $(s = h_0e_0 + h_1e_1, h_0, h_1)$ as input, the BGF algorithm outputs a vector $\mathbf{e}$. Then we count a decryption failure if and only if $\mathbf{e} \neq (\mathbf{e_0}, \mathbf{e_1})$. For $4500 \leq m \leq 5100$ and $\epsilon = 0, 1$, the frequency of decryption failures in our experiment is listed in Table 3, and their trend is depicted in Fig. 2. From the figure, we can see that the decryption failure rates are significantly higher than the average when both the secret key (h_0, h_1) and the error (e_0, e_1) satisfy the gathering property.

How to Sample Keys and Errors. Directly sampling $(h_0, h_1), (e_0, e_1)$ from $\mathrm{K}_{m,\epsilon}(w)$ and $\mathrm{E}_{m,\epsilon}(t/2, t/2)$ uniformly is difficult. We observe that the decryption failures are preserved by cyclic shifting. That is, $(h_0, h_1), (e_0, e_1)$ gives a decryption failure if and only if $(x^i h_0, x^i h_1), (x^j e_0, x^j e_1)$ gives a decryption failure for any $i, j \in \mathbb{Z}$. As a result, we can take the following strategy, which samples the keys and errors such that 1's of h_0 and e_0 roughly gather in $[0, m)$. Note that we are interested in $m \leq r/2, \epsilon \in \{0, 1\}$ in experiments.

Firstly, we focus on the case $\epsilon = 0$. To sample (h_0, h_1), we first set the 0-th position of h_0 to 1 and then randomly choose $(w/2 - \epsilon - 1)$ positions from $[1, m)$

Table 3. The frequency of decryption failures for (h_0, h_1) and (e_0, e_1) satisfying the (m, ϵ)-gathering property. N represents the number of decryptions performed, and F represents the number of decryption failures observed.

(m, ϵ)	$(4500, 0)$	$(4600, 0)$	$(4700, 0)$	$(4800, 0)$	$(4900, 0)$	$(5000, 0)$	$(5100, 0)$	$(5200, 0)$
N	240393	595827	1496235	3330070	952115	2507712	6605312	19727185
F	160	160	160	160	16	16	16	16

(m, ϵ)	$(4500, 1)$	$(4600, 1)$	$(4700, 1)$	$(4800, 1)$	$(4900, 1)$	$(5000, 1)$	$(5100, 1)$
N	139443	491647	911967	2957650	9176502	19197539	28910000
F	15.5*	14	15	13.5*	15	14.5*	6.5*

* The '.5' comes from the rejection sampling in Algorithm 2, where a decryption failure in the overlapping area counts as 0.5.

and set them to 1. Denoting j to be the last position of h_0 with a value of 1, then we randomly choose ϵ positions from $[j+1, r)$ and set them to 1.[1] h_1 is just generated randomly such that $w_H(h_1) = w/2$. Such procedure is summarized in Algorithm 2. (e_0, e_1) is sampled in the same way as (h_0, h_1), while the input of the algorithm is set to be $(t, m, \epsilon = 0)$.

It can be proved that the decryption failure rate for keys and errors sampled as above equals to that for $\mathrm{K}_{m,\epsilon}(w)$ and $\mathrm{E}_{m,\epsilon}(t/2, t/2)$. Let $\mathrm{K}^{(0)}_{m,\epsilon}(w)$ and $\mathrm{E}^{(0)}_{m,\epsilon}(t)$ denote the set of (h_0, h_1) and (e_0, e_1) generated as above respectively. Denote $\mathrm{K}^{(b)}_{m,\epsilon}(w) = \{(x^b h_0, x^b h_1) : (h_0, h_1) \in \mathrm{K}^{(0)}_{m,\epsilon}(w)\}$ and $\mathrm{E}^{(b)}_{m,\epsilon}(t) = \{(x^b e_0, x^b e_1) : (e_0, e_1) \in \mathrm{E}^{(0)}_{m,\epsilon}(t)\}$. Then it is clear that

$$\mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}^{(0)}_{m,\epsilon}(w)\\ (e_0,e_1)\xleftarrow{\$}\mathrm{E}^{(0)}_{m,\epsilon}(t)}} = \mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}^{(b)}_{m,\epsilon}(w)\\ (e_0,e_1)\xleftarrow{\$}\mathrm{E}^{(b')}_{m,\epsilon}(t)}} \tag{11}$$

for any $b, b' \in \mathbb{Z}$ because of the cyclic property. Additionally, for $\epsilon = 0$, $\mathrm{K}_{m,\epsilon}(w)$ is exactly the disjoint union of $\mathrm{K}^{(b)}_{m,\epsilon}(w), 0 \le b < r$, and $\mathrm{E}_{m,\epsilon}(t/2, t/2)$ is the disjoint union of $\mathrm{E}^{(b)}_{m,\epsilon}(t), 0 \le b < r$. As a result, for $\epsilon = 0$ we can deduce that

$$\mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)\\ (e_0,e_1)\xleftarrow{\$}\mathrm{E}_{m,\epsilon}(t/2,t/2)}} = \mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}^{(0)}_{m,\epsilon}(w)\\ (e_0,e_1)\xleftarrow{\$}\mathrm{E}^{(0)}_{m,\epsilon}(t)}} . \tag{12}$$

For $\epsilon = 1$, Equation (11) still holds. However, the sets $\mathrm{K}^{(b)}_{m,\epsilon}(w), 0 \le b < r$, intersect with each other, which makes Equation (12) no longer holds. We solve this issue by rejection sampling. Specifically, we observe that there is no (h_0, h_1) simultaneously drops into more than 2 of $\mathrm{K}^{(b)}_{m,\epsilon}(w)$, i.e., each element in the overlapping area $\mathrm{K}^{(0)}_{m,\epsilon}(w) \cap (\cup_{1 \le b < r} \mathrm{K}^{(b)}_{m,\epsilon}(w))$ appears exactly twice in the complete union $\cup_{0 \le b < r} \mathrm{K}^{(b)}_{m,\epsilon}(w)$. A similar conclusion also holds for the sets $\mathrm{E}^{(b)}_{m,\epsilon}(t)$.

[1] For $\epsilon = 0$ there is nothing to do in this step.

Therefore, by accepting the (h_0, h_1) and (e_0, e_1) that drop in the overlapping area with probability 1/2 (see Algorithm 2),[2] we can obtain a result similar to Equation (12). A complete proof can be found in Appendix A.

Algorithm 2: Sampling the keys and errors

```
Input: $w, m$ and $\epsilon = 0$ or 1
Output: $(h_0, h_1) \in \mathcal{R}^2$
Randomly generate $h_1 \in \mathcal{R}$ such that $w_H(h_1) = w/2$
$h_0 \leftarrow 0 \in \mathcal{R}$
Set the 0-th position of $h_0$ to 1
Randomly choose $w/2 - \epsilon - 1$ positions $\subseteq [1, m)$ and set them to 1
$j \leftarrow$ the last position of $h_0$ whose value is 1
Randomly choose $\epsilon$ positions $\subseteq [j + 1, r)$ and set them to 1.
if $\epsilon = 1$ then
    if $(h_0, h_1)$ is in the overlapping area then
        Accept $(h_0, h_1)$ with probability 1/2
    end if
end if
return $(h_0, h_1)$
```

Lemma 1. *Denote* $\mathtt{F}(w, m, \epsilon)$ *to be the random function corresponding to Algorithm 2. Then for* $m \leq \frac{r}{2}$ *and* $\epsilon \in \{0, 1\}$*, it has*

$$\mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)\\ (e_0,e_1)\xleftarrow{\$}\mathrm{E}_{m,\epsilon}(t/2,t/2)}} = \mathrm{DFR}_{\substack{(h_0,h_1)\leftarrow\mathtt{F}(w,m,\epsilon)\\ (e_0,e_1)\leftarrow\mathtt{F}(t,m,\epsilon)}} . \tag{13}$$

For $m > \frac{r}{2}$ or $\epsilon \geq 2$, the rejection sampling approach may be extended by first determining how many sets $\mathrm{K}^{(b)}_{m,\epsilon}(w)$ an element (h_0, h_1) drops in, and then accepting it with proper probability. However, determining the exact number of sets is complicated and time consuming, and better solutions to this problem have yet to be devised.

3.2 An Explanation of the Gathering Property

Our basic observation is that the gathering property considerably raises the numbers of unsatisfied parity checks for partial positions. Then the number of bits that are incorrectly flipped can be increased for these positions, which makes the decoding more likely to fail.

Specifically, we consider an example that both $(h_0, h_1), (e_0, e_1) \in \mathcal{R}^2$ satisfy the (m, ϵ)-gathering property such that $\epsilon = 0$ and all the 1's of h_0 and e_0 gather in their first m positions. As depicted in Fig. 3, there is a diagonal area of the

[2] An equivalent way is to not perform reject sampling but to count each decryption failure in the overlapping area as 0.5.

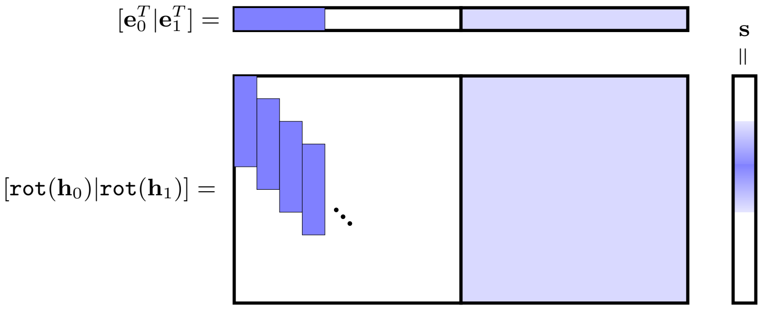
$[\mathbf{e}_0^T|\mathbf{e}_1^T] =$
$[\mathtt{rot}(\mathbf{h}_0)|\mathtt{rot}(\mathbf{h}_1)] =$
$\mathbf{s}$
$||$

Fig. 3. An explanation of the gathering property. The shadow parts represent that the 1's gather in these positions.

matrix $\mathbf{H} = [\mathtt{rot}(\mathbf{h}_0)|\mathtt{rot}(\mathbf{h}_1)]$ in which the 1's gather. As a result, the syndrome $s = h_0e_0 + h_1e_1$ will be denser around the m-th position. Now we consider the first iteration of the bit-flipping algorithm, in which the number of unsatisfied parity checks $\mathtt{UPC}(\mathbf{0}, i) = |\mathrm{Supp}(\mathbf{s}) \cap \mathrm{Supp}(\mathbf{h_i})|$ is computed at first. Due to the property of $\mathbf{s}$, the numbers of unsatisfied parity checks $\mathtt{UPC}(\mathbf{0}, i), 0 \leq i \leq m-1$, are more likely to be greater than other positions. This eventually increases the risk of a correct bit being mistakenly flipped in the first m positions. In fact, this phenomenon does happen when decryption failure occurs. For BIKE's parameter set targeting 128-bit security, we set $m = 4500$ and test the average unsatisfied parity check number of the first m positions and that of all positions, and find that the local unsatisfied parity check number is much higher than the global one in each iteration of the BGF algorithm, see Table 4.

Table 4. A comparison of the local UPC and the global UPC.

Iteration	Average UPC of the first m positions	Average UPC of all positions
0	31.3864	26.4111
1	57.2082	42.7164
2	83.5507	56.5557
3	114.588	73.0108
4	148.179	93.1936

3.3 Number of Keys & Errors Satisfying the Gathering Property

The purpose of this subsection is to prove the following statement.

Lemma 2. *Suppose $m < r/2$ and $\epsilon \in \{0, 1\}$, then it has*

$$1 - \xi \leq \frac{|\mathrm{K}_{m,\epsilon}(w)|}{\tau(w/2, m, \epsilon)} \leq 1 \textit{ and } 1 - \xi \leq \frac{|\mathrm{E}_{m,\epsilon}(t/2, t/2)|}{\tau(t/2, m, \epsilon)} \leq 1 \ , \tag{14}$$

where

$$\tau(x, m, \epsilon) \coloneqq r \sum_{d=x-\epsilon}^{m} \binom{d-2}{x-2-\epsilon}\binom{r-d}{\epsilon}\binom{r}{x} \ . \tag{15}$$

For $\epsilon = 0$ it has $\xi = 0$. For $\epsilon = 1, m \leq 5200$ and BIKE's parameter set with 128-bit security, it has $\xi < 0.05$.

Proof. For $\epsilon = 0$, $\mathrm{K}_{m,\epsilon}(w)$ is the disjoint union of $\mathrm{K}^{(b)}_{m,\epsilon}(w), 0 \leq b < r$, then it has $|\mathrm{K}_{m,\epsilon}(w)| = \sum_{0 \leq b < r} |\mathrm{K}^{(b)}_{m,\epsilon}(w)| = r|\mathrm{K}^{(0)}_{m,\epsilon}(w)|$. To count the number of (h_0, h_1) in $\mathrm{K}^{(0)}_{m,\epsilon}(w)$, we enumerate the position $d \in (0, m)$ such that the d-th position of h_0 is 1 and there are exactly $(w/2 - \epsilon - 2)$ 1's in the $(0, d)$ positions of h_0. Clearly for a fixed d there are $\binom{d-2}{w/2-2-\epsilon}\binom{r-d}{\epsilon}\binom{r}{w/2}$ distinct (h_0, h_1) in $\mathrm{K}^{(0)}_{m,\epsilon}(w)$, and then the lemma follows directly.

For $\epsilon = 1$, $\cup_{0 \leq b < r} \mathrm{K}^{(b)}_{m,\epsilon}(w)$ is not a disjoint union. Then counting the overlapping area gives an estimate of $\mathrm{K}_{m,\epsilon}(w)$, where the complete proof can be found in Appendix B. □

In this paper, we only consider BIKE's parameter set with 128-bit security, and focus on the parameters $\epsilon = 1, m \leq 5200$ or $\epsilon = 0$. For $\epsilon = 1, m = 5200$, it can be calculated from Lemma 2 that $\xi < 0.05$, which means that the error is no more than $|\log_2(1 - \xi)| < 0.08$ bits. The error is further reduced when m decreases. Using Lemma 2, we calculate the probability $p_{m,\epsilon} = |\mathrm{K}_{m,\epsilon}(w)|/|\mathrm{K}(w)|$ and $q_{m,\epsilon} = |\mathrm{E}_{m,\epsilon}(t/2, t/2)|/|\mathrm{E}(t)|$ for the (m, ϵ) involved in our experiments of Section 3.1, which are listed in Table 5.

Table 5. The probability $p_{m,\epsilon}$ and $q_{m,\epsilon}$ for the (m, ϵ) involved in our experiment and BIKE's parameter set targeting 128-bit security. The numbers are presented in their logarithmic form.

(m, ϵ)	$(4500, 0)$	$(4600, 0)$	$(4700, 0)$	$(4800, 0)$	$(4900, 0)$	$(5000, 0)$	$(5100, 0)$	$(5200, 0)$
$p_{m,\epsilon}$	-96.09	-93.86	-91.67	-89.53	-87.43	-85.37	-83.36	-81.39
$q_{m,\epsilon}$	-94.17	-92.06	-90.0	-87.98	-86.0	-84.06	-82.16	-80.3

(m, ϵ)	$(4500, 1)$	$(4600, 1)$	$(4700, 1)$	$(4800, 1)$	$(4900, 1)$	$(5000, 1)$	$(5100, 1)$
$p_{m,\epsilon}$	-89.13	-86.95	-84.81	-82.72	-80.67	-78.66	-76.69
$q_{m,\epsilon}$	-87.29	-85.23	-83.22	-81.25	-79.32	-77.43	-75.58

4 A New Class of Weak Keys

In this section, we focus on the construction of weak keys that have decryption failure rate higher than the average. Our basic observation is that the set of keys satisfying the (m,ϵ)-gathering property (i.e., $\mathrm{K}_{m,\epsilon}(w)$) may have higher decryption failure rate. Specifically, denote

$$\mathrm{DFR}^{\mathrm{K}}_{m,\epsilon} \coloneqq \mathrm{DFR}_{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w),(e_0,e_1)\xleftarrow{\$}\mathrm{E}(t)} \tag{16}$$

as the decryption failure rate for the key drawn from $\mathrm{K}_{m,\epsilon}(w)$ and the error drawn from $\mathrm{E}(t)$. Then we can deduce a lower bound by

$$\mathrm{DFR}^{\mathrm{K}}_{m,\epsilon} \geq \mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)\\(e_0,e_1)\xleftarrow{\$}\mathrm{E}_{m,\epsilon}(\frac{t}{2},\frac{t}{2})}} \cdot \Pr_{(e_0,e_1)\xleftarrow{\$}\mathrm{E}(t)}[(e_0,e_1)\in \mathrm{E}_{m,\epsilon}(\frac{t}{2},\frac{t}{2})] \ . \tag{17}$$

For example, using the experimental results in Section 3.1, we have $\mathrm{DFR}^{\mathrm{K}}_{m,\epsilon} \geq 2^{-22.08}\cdot 2^{-75.58} = 2^{-97.66}$ for $m = 5100$ and $\epsilon = 1$, which is much higher than the average decryption failure rate 2^{-128} given in BIKE. However, the probability of a random key (h_0,h_1) falling in $\mathrm{K}_{m,\epsilon}(w)$ is too small to impact the average decryption failure rate. Next, we focus on the case $\epsilon = 1$, and show the following two facts.

(i) The set of weak keys $\mathrm{K}_{m,\epsilon}(w)$ can be extended by using the isomorphisms of the ring $\mathcal{R}$.
(ii) Based on the extended weak keys, a lower bound greater than 2^{-128} on the average decryption failure rate can be derived for BIKE's parameter set targeting 128-bit security.

4.1 Extending Weak Keys Using Isomorphism

Let ϕ_i be an isomorphism of $\mathcal{R}$, and denote $\mathrm{K}^{\phi_i}_{m,\epsilon}(w)$ to be the set of keys obtained by applying ϕ_i to $\mathrm{K}_{m,\epsilon}(w)$, i.e.,

$$\mathrm{K}^{\phi_i}_{m,\epsilon}(w) \coloneqq \{(\phi_i(h_0),\phi_i(h_1)) : (h_0,h_1)\in \mathrm{K}_{m,\epsilon}(w)\}. \tag{18}$$

Note that ϕ_i preserves decryption failures, i.e., $(h_0,h_1),(e_0,e_1)$ gives a decryption failure if and only if $(\phi_i(h_0),\phi_i(h_1)),(\phi_i(e_0),\phi_i(e_1))$ gives a decryption failure. Thus we have the following lemma.

Lemma 3. *For any isomorphism $\phi_i : \mathcal{R}\to\mathcal{R}$, it has*

$$\mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}^{\phi_i}_{m,\epsilon}(w)\\(e_0,e_1)\xleftarrow{\$}\mathrm{E}(t)}} = \mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)\\(e_0,e_1)\xleftarrow{\$}\mathrm{E}(t)}} = \mathrm{DFR}^{\mathrm{K}}_{m,\epsilon} \ . \tag{19}$$

Next we focus on the decryption failure rate for the keys drawn uniformly from the union

$$\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w) \coloneqq \bigcup_{1\leq i<r/2} \mathrm{K}^{\phi_i}_{m,\epsilon}(w). \tag{20}$$

Note that only half of the isomorphisms are considered due to $\mathrm{K}^{\phi_i}_{m,\epsilon}(w) = \mathrm{K}^{\phi_{-i}}_{m,\epsilon}(w)$. To begin with, we show for proper choices of (m,ϵ), the above union is 'roughly' disjoint. That is, denote $\mathrm{K}^{\texttt{overlap}}_{m,\epsilon}(w) = \cup_{i,j}(\mathrm{K}^{\phi_i}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_j}_{m,\epsilon}(w))$ as the overlapping area of the union in (20), then it has

$$\frac{|\mathrm{K}^{\texttt{overlap}}_{m,\epsilon}(w)|}{|\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)|} \le \frac{|\mathrm{K}^{\phi_i}_{m,\epsilon}(w) \cap \mathrm{K}^{\texttt{overlap}}_{m,\epsilon}(w)|}{|\mathrm{K}^{\phi_i}_{m,\epsilon}(w)|} \le \delta \ , \tag{21}$$

where δ is a small number. The detailed proof can be found in Appendix C. For example, when $m = 4000, \epsilon = 1$, δ is as small as 2^{-35}. Based on this observation, we can prove the following result.

Theorem 1. *Suppose the error* (e_0, e_1) *is drawn from* $\mathrm{E}(t)$ *uniformly at random, then for the set of keys* $\mathrm{K}^{\mathit{union}}_{m,\epsilon}(w) = \cup_{1\le i<r/2}\mathrm{K}^{\phi_i}_{m,\epsilon}(w)$ *it has*

$$\mathrm{DFR}_{(h_0,h_1)\xleftarrow{\$}\mathrm{K}^{\mathit{union}}_{m,\epsilon}(w)} \ge \mathrm{DFR}_{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)} - \delta \ , \tag{22}$$

and $|\mathrm{K}^{\mathit{union}}_{m,\epsilon}(w)| \ge (1-\delta)(r-1)/2 \cdot |\mathrm{K}_{m,\epsilon}(w)|$.

Proof. Since $|\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)| \ge \sum_i(|\mathrm{K}^{\phi_i}_{m,\epsilon}(w)| - |\mathrm{K}^{\phi_i}_{m,\epsilon}(w) \cap \mathrm{K}^{\texttt{overlap}}_{m,\epsilon}(w)|)$, then it follows directly from (21) that $|\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)| \ge (1-\delta)(r-1)/2 \cdot |\mathrm{K}_{m,\epsilon}(w)|$. It remains to show (22). Denote $\bar{h} = (h_0, h_1)$, $\tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w) = \mathrm{K}^{\phi_i}_{m,\epsilon}(w) \cap \mathrm{K}^{\texttt{overlap}}_{m,\epsilon}(w)$ and $\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w) = \mathrm{K}^{\phi_i}_{m,\epsilon}(w) - \tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)$, then

$$\mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)} \ge \sum_{1\le i\le \frac{r}{2}} \mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)} \cdot \mathop{\Pr}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)}[\bar{h} \in \bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)] \ . \tag{23}$$

Note that

$$\begin{aligned}
\mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\phi_i}_{m,\epsilon}(w)} &= \mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)} \cdot \mathop{\Pr}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\phi_i}_{m,\epsilon}(w)}[\bar{h} \in \bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)] \\
&\qquad + \mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)} \cdot \mathop{\Pr}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\phi_i}_{m,\epsilon}(w)}[\bar{h} \in \tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)] \\
&\le \mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)} \cdot \mathop{\Pr}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\phi_i}_{m,\epsilon}(w)}[\bar{h} \in \bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)] + \delta \ ,
\end{aligned}$$

then it follows from (23) that

$$\begin{aligned}
\mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)} &\ge \sum_{1\le i\le \frac{r}{2}} \Big(\mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\phi_i}_{m,\epsilon}(w)} - \delta \Big) \cdot \mathop{\Pr}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)}[\bar{h} \in \mathrm{K}^{\phi_i}_{m,\epsilon}(w)] \\
&= \Big(\mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)} - \delta \Big) \cdot \sum_{1\le i\le \frac{r}{2}} \mathop{\Pr}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)}[\bar{h} \in \mathrm{K}^{\phi_i}_{m,\epsilon}(w)] \\
&\ge \mathop{\mathrm{DFR}}\limits_{\bar{h}\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)} - \delta,
\end{aligned}$$

which completes the proof. □

4.2 Lower Bound on the Average DFR

In this subsection we give lower bounds on the average decryption failure rate by using Theorem 1 and the formula

$$\mathrm{DFR}_{\mathrm{avg}} \geq 2 \cdot \mathrm{DFR}_{(h_0,h_1) \xleftarrow{\$} \mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)} \cdot \frac{|\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)|}{|\mathrm{K}(w)|} \ , \tag{24}$$

where the '2×' comes from the gathering property defined for the right side of (h_0, h_1), i.e, the 1's of h_1 are gathering while h_0 is chosen freely. We note that there is a very little chance that both sides of (h_0, h_1) have the property that their 1's gathering. However, the probability is too small to have any effect on (24), and a rigorous treatment can be performed in a way similar to that of Theorem 1.

Next, we consider BIKE's parameter set targeting 128-bit security, and focus on $2900 \leq m \leq 4000, \epsilon = 1$. By Theorem 1, $\mathrm{DFR}_{(h_0,h_1) \xleftarrow{\$} \mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)}$ can be estimated by simulating the decryption failure rate $\mathrm{DFR}_{(h_0,h_1) \xleftarrow{\$} \mathrm{K}_{m,\epsilon}(w)}$. In our experiments, we sample (h_0, h_1) and (e_0, e_1) from $\mathrm{K}_{m,\epsilon}(w)$ and $\mathrm{E}(t)$ uniformly at random. For each (m, ϵ), the number of decryption performed and the number of decryption failures are listed in Table 6. Note that for these set of parameters, it has $\delta \approx 2^{-35}$, which is negligible with respect to $\mathrm{DFR}_{(h_0,h_1) \xleftarrow{\$} \mathrm{K}_{m,\epsilon}(w)}$. On the other hand, lower bound on $|\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)|/|\mathrm{K}(w)|$ can be derived by using Theorem 1 and Lemma 2.

Table 6. Estimates of the decryption failure rates for the set of weak keys $\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)$. N represents the number of decryptions, and F represents the number of decryption failures. p is 2 times the probability that a random key (h_0, h_1) is in $\mathrm{K}^{\mathtt{union}}_{m,\epsilon}(w)$.

(m,ϵ)	$(2900,1)$	$(3100,1)$	$(3200,1)$	$(3400,1)$	$(3500,1)$	$(3600,1)$	$(4000,1)$
N	2996871	5459695	32903584	165860000	214960000	315470000	8745860000
F	16	16	31.5^*	25.5^*	13.5^*	11	13
DFR	−17.52	−18.38	−19.99	−22.63	−23.92	−24.77	−29.33
p	−119.45	−112.76	−109.58	−103.51	−100.62	−97.80	−87.28

* The '.5' comes from the rejection sampling in Algorithm 2, where a decryption failure in the overlapping area counts as 0.5.

Combining the above results, we can give estimated lower bounds on the average decryption failure rate by using (24). Fig. 4 depicts the corresponding results. From the figure, $(m = 4000, \epsilon = 1)$ yields an estimated lower bound for the average decryption failure rate such that

$$\mathrm{DFR}_{\mathrm{avg}} \geq 2^{-29.33} \cdot 2^{-87.28} = 2^{-116.61}. \tag{25}$$

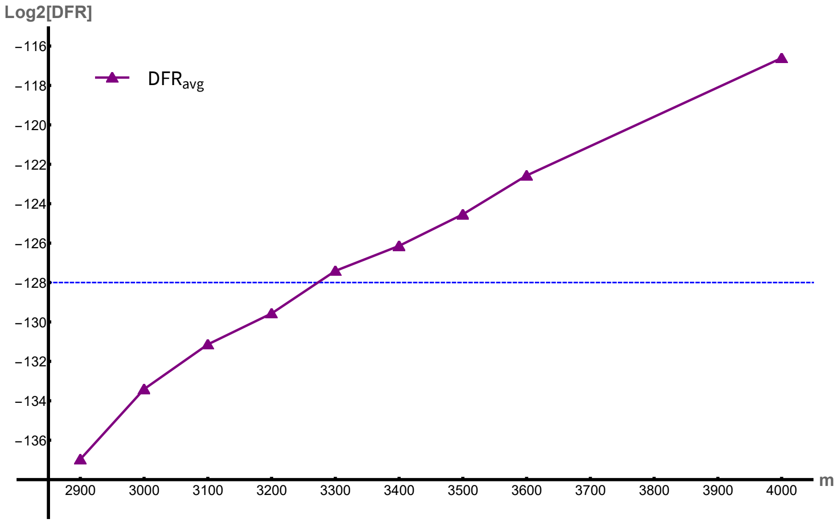

Fig. 4. Lower bounds on the average decryption failure rate for BIKE's parameter set targeting 128-bit security.

Taking the simulation error into consideration, we note that 13 decryption failures are observed from 8745860000 decryptions. Then using the framework in Section 2.1, we have

$$\text{DFR}_{(h_0,h_1)\overset{\$}{\leftarrow}\text{K}_{m,\epsilon}(w)} \geq F/N - 2\cdot\sqrt{F}/N \approx 2^{-30.49} \tag{26}$$

at 95% confidence level. As a result, we can deduce that the average decryption failure rate for BIKE's parameter set targeting 128-bit security is lower bounded by $\text{DFR}_{\text{avg}} \geq 2^{-30.49} \cdot 2^{-87.28} = 2^{-117.77}$ at 95% confidence level. For larger $m > 4000$, we should expect better lower bounds according to Fig. 4. However, in this case $\text{DFR}_{(h_0,h_1)\overset{\$}{\leftarrow}\text{K}^{\texttt{union}}_{m,\epsilon}(w)}$ is very small and difficult to simulate experimentally.

5 A Key Recovery Attack Using Decryption Failures

In this section, we demonstrate how the class of weak keys introduced in Section 4 can be utilized to launch a key recovery attack against QC-MDPC schemes with CCA security. The attack is in a multi-target mode, which means that numerous targets are queried, each with its own query limit, with the goal of recovering the key for at least one of these targets. Additionally, we assume that a ciphertext is only valid for a single target, i.e., the scheme provides multi-target protection.

5.1 Attack Model

The attack can be modeled as follows.

Step 0 (*Setup*). To begin with, we choose the proper parameters m, ϵ. Let $\mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)$ to be the set of weak keys defined in Section 4, i.e., the set of all $(h_0, h_1) \in \mathrm{K}_{m,\epsilon}(w)$ such that (h_0, h_1) or (h_1, h_0) satisfying the (m, ϵ)-gathering property. Denote

$$p^{\mathtt{weak}}_{m,\epsilon} := \frac{|\mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)|}{|\mathrm{K}(w)|} \tag{27}$$

to be the probability that a random secret key is in $\mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)$, and denote $\mathrm{DFR}^{\mathtt{weak}}_{m,\epsilon}$ to be the decryption failure rate for key drawn from $\mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)$ and error drawn from $E(t)$.

Step 1 (*Construct Ciphertexts for a Target*). For a target T, initialize the set of ciphertexts to be $\mathrm{C}_T = \{\}$. Randomly generate $1/\mathrm{DFR}^{\mathtt{weak}}_{m,\epsilon}$ seeds $m \in \{0,1\}^{256}$, then compute $(e_0, e_1) = \mathtt{H}(m)$ according to the error generation step of BIKE, and add the corresponding ciphertext to C_T. Clearly $|\mathrm{C}_T| = 1/\mathrm{DFR}^{\mathtt{weak}}_{m,\epsilon}$.

Step 2 (*Query and Collect Decryption Failures*). Query the target T's decryption oracle to decrypt all ciphertexts in C_T. If a decryption failure occurs, then go to the key recovery step. Else, choose another target and then repeat the ciphertexts construction and query steps.

Step 3 (*Recover the Secret Key*). For a target T that has a decryption failure, there is a probability $p_{\mathtt{true}}$ that T's secret key (h_0, h_1) is in $\mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)$. In the case that $(h_0, h_1) \in \mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)$, there exists an isomorphism $\phi_i, 1 \le i < r/2$, such that $(\phi_i^{-1}(h_0), \phi_i^{-1}(h_1))$ or $(\phi_i^{-1}(h_1), \phi_i^{-1}(h_0))$ satisfying the (m, ϵ)-gathering property. With the gathering property as extra information, the secret key can be efficiently recovered using a modified information set decoding (ISD) algorithm presented in the next subsection. Then the attacker enumerates the isomorphisms $\phi_i, 1 \le i < r/2$ and tries to recover $(\phi_i^{-1}(h_0), \phi_i^{-1}(h_1))$ or $(\phi_i^{-1}(h_1), \phi_i^{-1}(h_0))$ from

$$\phi_i^{-1}(h) = \phi_i^{-1}(h_1) \cdot (\phi_i^{-1}(h_0))^{-1} \tag{28}$$

utilizing the modified ISD algorithm. If T's secret key (h_0, h_1) is in $\mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)$, then for some ϕ_i the modified ISD algorithm can efficiently recovers T's secret key and the attack terminates. If T's secret key (h_0, h_1) is not in $\mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)$, it is unlikely the modified ISD algorithm can efficiently recover the secret key for any $1 \le i < r/2$. Then the attacker chooses another target and repeats the ciphertexts construction, query and key recovery steps.

Analysis of the Probability $p_{\mathtt{true}}$. Let 'FAIL' denote the event that a decryption failure occurs for the secret key $\bar{h} := (h_0, h_1)$, and denote $\mathrm{X} := \mathrm{K}(w) \times \mathrm{E}(t)$. Then it can be deduced that

$$\begin{aligned} p_{\mathtt{true}} &= \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\bar{h} \in \mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w) \mid \mathrm{FAIL}] \\ &= \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\mathrm{FAIL} \mid \bar{h} \in \mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)] \cdot \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\bar{h} \in \mathrm{K}^{\mathtt{weak}}_{m,\epsilon}(w)] / \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\mathrm{FAIL}] \\ &= \mathrm{DFR}^{\mathtt{weak}}_{m,\epsilon} \cdot p^{\mathtt{weak}}_{m,\epsilon} / \mathrm{DFR}_{\mathrm{avg}} \end{aligned}$$

by Bayes' theorem, where $\mathrm{DFR}_{\mathrm{avg}} = \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\mathrm{FAIL}]$ is the average decryption failure rate.

5.2 Information Set Decoding using Extra Information

Efficiently retrieving the secret key given extra information is a well-known topic in public key cryptography [12,8,17,30,50]. Several effects have been developed to address this issue in code-based cryptography. Horlemann et al. [33] present a general framework for recovering the key by employing hint information such as certain erroneous or error-free locations, or the Hamming weight of error blocks. Esser et al. [22] demonstrate that when a fraction of the secret key bits are erased or faulty, the key recovery for BIKE can be greatly accelerated. Kirshanova and May [36] show that a small portion of the secret key at any known positions can be used to successfully recover the entire secret key for McEliece. The purpose of this subsection is to develop an effective approach for using the gathering property to improve the key recovery in QC-MDPC.

Our goal is to recover a target's secret key (h_0, h_1) from the public key $h = h_1 h_0^{-1}$ using the fact that (h_0, h_1) satisfies the (m, ϵ)-gathering property. Denote $\mathbf{H} = (\mathbf{I}_r, \texttt{rot}(h))$, $\mathbf{e} = (\mathbf{h}_1^T, \mathbf{h}_0^T)^T$ and $\mathbf{s} = \mathbf{0}$. Then the problem can be stated as follows.

Problem 1. Given $\mathbf{H} \in \mathbb{F}_2^{r\times 2r}, \mathbf{s} \in \mathbb{F}_2^r$ and positive integers w, m and $\epsilon \geq 0$, find $\mathbf{e} = (\mathbf{h}_1^T, \mathbf{h}_0^T)^T$ such that $\mathbf{He} = \mathbf{s}$, $w_H(\mathbf{h_0}) = w_H(\mathbf{h_1}) = w/2$ and there exists an integer a such that $w_H(\mathbf{h}_0^{[a,a+m)}) = w/2 - \epsilon$.

Without the extra information of the gathering property, the above problem is typically solved using the information set decoding (ISD) algorithm.

Classical ISD Algorithm. The ISD algorithm iteratively finds a vector $\mathbf{e}$ such that $\mathbf{He} = \mathbf{s}$ and $w_H(\mathbf{e}) = w$. Suppose l, p are two integers, each iteration of the ISD algorithm is as follows.

- *Random Permutation*: Choose a random permutation matrix $\mathbf{P}$, and compute $\mathbf{HP}^{-1}$.
- *Gauss Elimination*: Apply Gauss elimination to the matrix $\mathbf{HP}^{-1}$ and obtain a matrix of the form

$$\mathbf{H}' \coloneqq \mathbf{THP}^{-1} = \begin{bmatrix} \mathbf{I}_{r-l} & \mathbf{H_1} \\ \mathbf{O} & \mathbf{H_2} \end{bmatrix}, \tag{29}$$

  where $\mathbf{T}$ is the corresponding Gauss elimination matrix, $\mathbf{O}$ is an $l \times (r-l)$ zero matrix. Denote $\mathbf{Pe} = (\mathbf{e}_1^T, \mathbf{e}_2^T)^T$ and $\mathbf{Ts} = (\mathbf{s}_1^T, \mathbf{s}_2^T)^T$ such that $\mathbf{e_1}, \mathbf{s_1} \in \mathbb{F}_2^{r-l}, \mathbf{e_2} \in \mathbb{F}_2^{r+l}$ and $\mathbf{s_2} \in \mathbb{F}_2^l$. Then the problem can be written as

$$\mathbf{H}_2\mathbf{e}_2 = \mathbf{s}_2, \mathbf{e}_1 = \mathbf{H}_1\mathbf{e}_2 + \mathbf{s}_1. \tag{30}$$

- *Column Match*: In this step, an algorithm $\texttt{COLUMNMATCH}(\mathbf{H_2}, \mathbf{s_2}, p)$ is called to generate a set $\mathrm{L} = \{\mathbf{e_2} : \mathbf{H_2 e_2} = \mathbf{s_2}, \mathrm{w}_H(\mathbf{e_2}) = p\}$. The $\texttt{COLUMNMATCH}$ algorithm differs depending on the ISD algorithms, e.g., Stern-Dumer [21], MMT[38], BJMM[5], etc.

– *Recover* $\mathbf{e}$: For all $\mathbf{e}_2 \in \mathrm{L}$, compute $\mathbf{e_1} = \mathbf{H_1 e_2} + \mathbf{s_1}$. If $w_H(\mathbf{e_1}) = w - p$, then the algorithm returns $\mathbf{e} = \mathbf{P}^{-1} \cdot (\mathbf{e}_1^T, \mathbf{e}_2^T)^T$. Else goes to the *Random Permutation* step and try another permutation matrix.

For a random $\mathbf{P}$, the probability that the Hamming weight of $\mathbf{Pe} = (\mathbf{e}_1^T, \mathbf{e}_2^T)^T$ splits to $w - p$ and p is

$$\mathrm{P}_{\mathrm{ISD}} = \frac{\binom{r-l}{w-p}\binom{n-r+l}{p}}{\binom{n}{w}} \quad . \tag{31}$$

Thus the ISD algorithm outputs the vector $\mathbf{e}$ after $\mathrm{P}_{\mathrm{ISD}}^{-1}$ iterations in average. For each iteration, the time and space costs are mainly the costs of COLUMNMATCH, which we denote by $T_{\mathtt{MATCH}}$ and $S_{\mathtt{MATCH}}$ respectively. For example, in the Stern-Dumer ISD algorithm [21], it has $T_{\mathtt{MATCH}} = S_{\mathtt{MATCH}} = \mathtt{max}\{\binom{(r+l)/2}{p/2}, \binom{(r+l)/2}{p/2}^2/2^l\}$. The total time complexity of the ISD algorithm can be represented as $T_{\mathtt{MATCH}} \cdot \mathrm{P}_{\mathrm{ISD}}^{-1}$ and the space complexity is $S_{\mathtt{MATCH}}$.

Remark 1. It is required that the Hamming weight of $\mathbf{e}$ can be split into $w/2$ and $w/2$ in Problem 1, whereas the ISD algorithm just returns a vector of Hamming weight w. In the context of key recovery, w is usually small enough such that the vector $\mathbf{e}$ is unique (up to cyclic shifting in the quasi-cyclic case), making the two conditions equivalent.

Using the Extra Information. Our basic idea is to use the extra information to increase the probability $\mathrm{P}_{\mathrm{ISD}}$. We note that the permutation $\mathbf{P}$ actually gives a random partition of all the positions $[0, 2r) = \mathrm{Q}_l \cup \mathrm{Q}_r$ such that $|\mathrm{Q}_l| = r - l$ and $|\mathrm{Q}_r| = r + l$, and the ISD algorithm outputs the vector $\mathbf{e}$ if $w_H(\mathbf{e}^{\mathrm{Q}_l}) = w - p$ and $w_H(\mathbf{e}^{\mathrm{Q}_r}) = p$.

Suppose $p \geq \epsilon$ and $l \leq r - m$. By using the extra information, we construct the partition as follows. First, randomly choose $b \in [0, r-1)$, then put the m positions $\mathbf{h}_0^{[b,b+m)}$ into Q_l, and put the other $r - m$ positions of $\mathbf{h}_0$ into Q_r. After that, randomly choose $r - l - m$ positions of $\mathbf{h}_1$ and put them into Q_l, and put the other $l + m$ positions of $\mathbf{h}_1$ into Q_r. The next steps are then carried out in the same manner as the classical ISD algorithm.

Note that with probability at least $1/r$, it has $w_H(\mathbf{h}_0^{[b,b+m)}) = w/2 - \epsilon$. In this case, the ISD algorithm outputs the correct vector $\mathbf{e}$ whenever the $r - l - m$ positions of $\mathbf{h}_1$ have exactly $(w/2 - p + \epsilon)$ 1's and the other $l + m$ positions of $\mathbf{h}_1$ have $(p - \epsilon)$ 1's. As a result, the success probability when using the extra information is

$$\mathrm{P}_{\mathrm{extra}} = \frac{1}{r} \frac{\binom{l+m}{p-\epsilon}\binom{r-l-m}{w/2-p+\epsilon}}{\binom{r}{w/2}} \quad , \tag{32}$$

And the total complexity is as follows.

Proposition 4 (ISD with extra information)**.** *The above algorithm gives a solution to Problem 1 in average time complexity* $T_{\mathtt{MATCH}} \mathrm{P}_{extra}^{-1}$ *and space complexity*

$S_{\texttt{MATCH}}$. *When using the* `COLUMNMATCH` *in the Stern-Dumer ISD algorithm, it has* $T_{\texttt{MATCH}} = S_{\texttt{MATCH}} = max\{\binom{(k+l)/2}{p/2}, \binom{(k+l)/2}{p/2}^2/2^l\}$.

For the quasi-cyclic case, each cyclic shift of (h_0, h_1) is also a solution to $h = h_1 h_0^{-1}$ [23,43]. As a result, the step that randomly chooses $b \in [0, r-1)$ can be skipped, and we just put the first m positions $\mathbf{h}_0^{[0,m)}$ into Q_l, and put the other $r - m$ positions of $\mathbf{h}_0$ into Q_r. This enables us to obtain an $r\times$ speedup for the quasi-cyclic case.

Corollary 5. *Given $h \in \mathcal{R}$, then (h_0, h_1) satisfying $h = h_1 h_0^{-1}$ and the (m, ϵ)-gathering property can be recovered in time complexity $T_{\texttt{MATCH}} \mathrm{P}_{extra}^{-1}/r$ and space complexity $S_{\texttt{MATCH}}$.*

5.3 Complexity Analysis

We present an analysis of the complexity of the key recovery attack in Section 5.1. The total complexity can be divided into two parts: the complexity of identifying decryption failures (Step 1 and Step 2) and the complexity of the key recovery step (Step 3).

The complexity of identifying decryption failures. On average, $1/p_{m,\epsilon}^{\texttt{weak}}$ targets are required to obtain a secret key falling in $\mathrm{K}_{m,\epsilon}^{\texttt{weak}}(w)$, which should result in a successful key recovery in Step 3. Thus both Step 1 and Step 2 are called $1/p_{m,\epsilon}^{\texttt{weak}}$ times. On the other hand, for a single target, the complexity of the ciphertexts construction and the decryption query is $|\mathrm{C}_T| = 1/\mathrm{DFR}_{m,\epsilon}^{\texttt{weak}}$. Thus the complexity of identifying decryption failures is

$$1/p_{m,\epsilon}^{\texttt{weak}} \cdot 1/\mathrm{DFR}_{m,\epsilon}^{\texttt{weak}} \quad . \tag{33}$$

The complexity of the key recovery step. For the key recovery step, the attacker calls the ISD algorithm $2 \cdot r/2$ times for each secret key, and the probability that a secret key (in this step) falls in $\mathrm{K}_{m,\epsilon}^{\texttt{weak}}(w)$ is $p_{\texttt{true}}$. Thus the complexity of the key recovery step is

$$1/p_{\texttt{true}} \cdot r \cdot T_{\texttt{ISD}} \quad , \tag{34}$$

where $p_{\texttt{true}} = p_{m,\epsilon}^{\texttt{weak}} \cdot \mathrm{DFR}_{m,\epsilon}^{\texttt{weak}}/\mathrm{DFR}_{\mathrm{avg}}$ and $T_{\texttt{ISD}}$ is the time complexity of a single call of the ISD algorithm as in Corollary 5.

Besides, note that Step 1 and Step 2 can be performed in polynomial space complexity, thus the total space complexity of the key recovery attack is $S_{\texttt{ISD}}$, which is the space complexity of a single call of the ISD algorithm. Therefore, the total complexity of the key recovery attack is as follows.

Theorem 2. *The key recovery attack in Section 5.1 can be mounted in time complexity*

$$C_{total} = (\mathrm{DFR}_{m,\epsilon}^{weak} \cdot p_{m,\epsilon}^{weak})^{-1} + p_{true}^{-1} \cdot r \cdot T_{ISD}, \tag{35}$$

and space complexity S_{ISD}, where $p_{true} = p_{m,\epsilon}^{weak} \cdot \mathrm{DFR}_{m,\epsilon}^{weak} \cdot \mathrm{DFR}_{avg}^{-1}$. The number of targets required is $1/p_{m,\epsilon}^{weak}$, and the number of queries required for a single target is $1/\mathrm{DFR}_{m,\epsilon}^{weak}$.

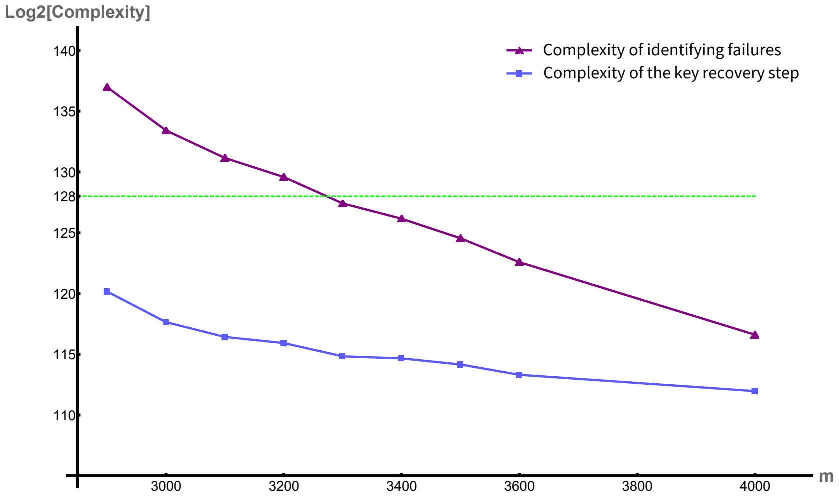

Fig. 5. The complexity of identifying a decryption failure and the complexity of ISD. The complexity of the key recovery attack can be viewed as their maximum.

Concrete Attack Complexity for BIKE. Next, we consider BIKE's parameter set targeting 128-bit security, and give the concrete complexity of the key recovery attack. We should note that the precise value of the average decryption failure rate $\mathrm{DFR}_{\mathrm{avg}}$ is yet unknown for BIKE. Nonetheless, an upper bound on $\mathrm{DFR}_{\mathrm{avg}}$ suffices for estimating the total complexity using Theorem 2. In the following analysis we assume that $\mathrm{DFR}_{\mathrm{avg}} < 2^{-80}$ for BIKE's parameter set targeting 128-bit security. We believe this upper bound is almost certain because the estimated value of $\mathrm{DFR}_{\mathrm{avg}}$ is about 2^{-128} in [3]. Despite the fact that our conclusion in Section 4 indicates that $\mathrm{DFR}_{\mathrm{avg}} \geq 2^{-116.61}$, there is no evidence that $\mathrm{DFR}_{\mathrm{avg}}$ can be as large as 2^{-80}.

In this attack, we initialize (m, ϵ) such that $(2900 \leq m \leq 4000, \epsilon = 1)$. The $\mathrm{DFR}^{\mathtt{weak}}_{m,\epsilon}$ and $p^{\mathtt{weak}}_{m,\epsilon}$ have been simulated experimentally in Section 4.2, and their values are listed in Table 6. In Fig. 5, we illustrate the complexity of identifying a decryption failure and the complexity of the key recovery step. The complexity of the key recovery attack can be viewed as their maximum. It can be deduced from the figure that the total complexity is less than 2^{128} for $3300 \leq m \leq 4000$. For example, when $m = 4000$, it has $\mathrm{DFR}^{\mathtt{weak}}_{m,\epsilon} = 2^{-29.33}$, $p^{\mathtt{weak}}_{m,\epsilon} = 2^{-87.28}, T_{\mathrm{ISD}} = 2^{75.35}$ and the total complexity is

$$C_{\mathrm{total}} = 2^{116.61}. \tag{36}$$

In this case, the number of targets required is $1/p^{\mathtt{weak}}_{m,\epsilon} = 2^{87.28}$, and the number of queries required for a single target is $1/\mathrm{DFR}^{\mathtt{weak}}_{m,\epsilon} = 2^{29.33}$ (see Table 1 for the details).

Moreover, it can be observed from Fig. 5 that both the complexity of identifying failures and the complexity of the key recovery step decrease as m increases.[3] So better total complexity can be expected when (m, ϵ)-gathering property is considered for m slightly larger than 4000. However, large m leads to low decryption failure rate which is difficult to be simulated via experiment.

6 A Key Recovery Attack with Ciphertexts Reusing

We note that BIKE does not offer protection against multiple targets in its current version. As described in Section 2.2, each target shares the same generation of error $(e_0, e_1) = \mathtt{H}(m)$ without binding to the public key. Thus it is possible to mount an attack by reusing the ciphertexts. Specifically, the results in Section 3.1 demonstrate that the decryption failure rate is greatly increased when both the secret key (h_0, h_1) and the error (e_0, e_1) satisfy the (m, ϵ)-gathering property. Thus the attacker first constructs a set of ciphertexts C of which the corresponding errors satisfy the (m, ϵ)-gathering property. Then by querying a target T's decryption oracle to decrypt the ciphertexts C, the attacker has an advantage of identifying whether T's secret key (h_0, h_1) satisfies the (m, ϵ)-gathering property by identifying whether a decryption failure occurs. Then use the (m, ϵ)-gathering property as an extra information, the attacker is able to recover T's secret key by using the ISD algorithm in Section 5.2.

6.1 Attack Model (with Ciphertexts Reusing)

The attack can be modeled as follows.

Step 0 (*Setup*). To begin with, we choose the proper parameters m, ϵ. Let $\mathrm{DFR}_{m,\epsilon}$ denote the decryption failure rate when both the secret key and error satisfying the (m, ϵ)-gathering property. Denote

$$p_{m,\epsilon} := \frac{|\mathrm{K}_{m,\epsilon}(w)|}{|\mathrm{K}(w)|} \text{ and } q_{m,\epsilon} := \frac{|\mathrm{E}_{m,\epsilon}(t/2, t/2)|}{|\mathrm{E}(t)|} \tag{37}$$

to be the probability that a random secret key (or a random error) satisfies the (m, ϵ)-gathering property.

Step 1 (*Ciphertexts Construction*). Initialize the set of ciphertexts to be $\mathrm{C} = \{\}$. Randomly generate $1/(\mathrm{DFR}_{m,\epsilon} \cdot q_{m,\epsilon})$ seeds $m \in \{0, 1\}^{256}$, then compute the errors $(e_0, e_1) = \mathtt{H}(m)$ according to the error generation step of BIKE. If an error (e_0, e_1) satisfies $w_H(e_0) = w_H(e_1) = t/2$ and the (m, ϵ)-gathering property, then add the corresponding ciphertext to C. On average, there will be $1/\mathrm{DFR}_{m,\epsilon}$ ciphertexts in C.

Step 2 (*Query and Collect Decryption Failures*). For a target T, query T's decryption oracle to decrypt the ciphertexts in C. If a decryption failure occurs,

[3] To be more specific, the complexity of the key recovery step is the product of $p_{\mathtt{true}}^{-1}$ and $r \cdot T_{\mathtt{ISD}}$. For m around 4000, $p_{\mathtt{true}}$ increases faster than $T_{\mathtt{ISD}}$, leading to a drop in the complexity of the key recovery step. But as m becomes very large, the complexity of the key recovery step will eventually approaches to 2^{128}.

then go to the key recovery step. Else, choose another target T', and repeat the query step.

Step 3 (*Recover the Secret Key*). For a target T that has a decryption failure, there is a probability $p_{\texttt{true}}$ that T's secret key (h_0, h_1) satisfies the (m, ϵ)-gathering property. In this case, (h_0, h_1) can be efficiently recovered by using the modified ISD algorithm in Section 5.2. On the other hand, if (h_0, h_1) does not satisfy the (m, ϵ)-gathering property, it is unlikely the modified ISD algorithm can efficiently recover (h_0, h_1). Thus the attacker simply try to recover (h_0, h_1) by using the modified ISD algorithm. If (h_0, h_1) is recovered successfully, the attack terminates. Otherwise, the attacker chooses another target and repeats the query and key recovery steps.

Analysis of the Probability $p_{\texttt{true}}$. Let 'FAIL' denote the event that a decryption failure occurs for the secret key $\bar{h} \coloneqq (h_0, h_1)$, and denote $\mathrm{X} \coloneqq \mathrm{K}(w) \times \mathrm{E}_{m,\epsilon}(t/2, t/2)$. Then it can be deduced that

$$\begin{aligned} & p_{\texttt{true}} = \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\bar{h} \in \mathrm{K}_{m,\epsilon}(w) \mid \mathrm{FAIL}] \\ = \; & \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\mathrm{FAIL} \mid \bar{h} \in \mathrm{K}_{m,\epsilon}(w)] \cdot \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\bar{h} \in \mathrm{K}_{m,\epsilon}(w)] / \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\mathrm{FAIL}] \\ = \; & \mathrm{DFR}_{m,\epsilon} \cdot p_{m,\epsilon} / \mathrm{DFR}_{\mathrm{avg}}^{e \sim (m,\epsilon)} \end{aligned}$$

by Bayes' theorem, where $\mathrm{DFR}_{\mathrm{avg}}^{e \sim (m,\epsilon)} = \Pr_{(\bar{h},\bar{e}) \xleftarrow{\$} \mathrm{X}}[\mathrm{FAIL}]$ is the average decryption failure rate for error drawn from $\mathrm{E}_{m,\epsilon}(t/2, t/2)$ and random key.

6.2 Complexity Analysis

The total complexity consists of three parts: the complexity of preprocessing (Step 1), the complexity of identifying decryption failures (Step 2) and the complexity of the key recovery step (Step 3).

The complexity of preprocessing. The complexity of preprocessing is determined by the total number of errors computed in Step 1, which is equal to

$$1/(\mathrm{DFR}_{m,\epsilon} \cdot q_{m,\epsilon}) \quad . \tag{38}$$

The complexity of identifying decryption failures. On average, $1/p_{m,\epsilon}$ targets are required to obtain a secret key satisfying the (m, ϵ)-gathering property, which should result in a successful key recovery in Step 3. Thus Step 2 is called $1/p_{m,\epsilon}$ times. On the other hand, the number of decryption queries for a single target is $|\mathrm{C}| = 1/\mathrm{DFR}_{m,\epsilon}$. Thus the complexity of identifying decryption failures is

$$1/p_{m,\epsilon} \cdot 1/\mathrm{DFR}_{m,\epsilon} \quad . \tag{39}$$

The complexity of the key recovery step. For the key recovery step, the probability that a secret key satisfying the (m, ϵ)-gathering property is $p_{\texttt{true}}$. Thus the complexity of the key recovery step is

$$1/p_{\texttt{true}} \cdot T_{\texttt{ISD}} \quad , \tag{40}$$

where $p_{\mathtt{true}} = \mathrm{DFR}_{m,\epsilon} \cdot p_{m,\epsilon} / \mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ and $T_{\mathtt{ISD}}$ is the time complexity of a single call of the ISD algorithm as in Corollary 5.

Besides, note that the space complexity of Step 1 is $(\mathrm{DFR}_{m,\epsilon})^{-1}$, thus the total space complexity of the key recovery attack is $(\mathrm{DFR}_{m,\epsilon})^{-1} + S_{\mathtt{ISD}}$, where $S_{\mathtt{ISD}}$ is the space complexity of a single call of the ISD algorithm. Therefore, the total complexity of the key recovery attack is as follows.

Theorem 3. *The key recovery attack in Section 6.1 can be mounted in time complexity*

$$C_{total} = (\mathrm{DFR}_{m,\epsilon} \cdot q_{m,\epsilon})^{-1} + (\mathrm{DFR}_{m,\epsilon} \cdot p_{m,\epsilon})^{-1} + p_{\mathtt{true}}^{-1} \cdot T_{ISD}, \tag{41}$$

and space complexity $S_{\mathtt{MATCH}} + \mathrm{DFR}_{m,\epsilon}^{-1}$ *where* $p_{\mathtt{true}} = \mathrm{DFR}_{m,\epsilon} \cdot p_{m,\epsilon} / \mathrm{DFR}_{avg}^{e\sim(m,\epsilon)}$. *The number of targets required is* $1/p_{m,\epsilon}$, *the number of queries required for a single target is* $1/(\mathrm{DFR}_{m,\epsilon})$.

Concrete Attack Complexity for BIKE. Next, we consider BIKE's parameter set targeting 128-bit security. In this attack, we initialize $m = 5100, \epsilon = 1$. Then from Table 3 and Table 5 it has $p_{m,\epsilon} = 2^{-76.69}, q_{m,\epsilon} = 2^{-75.58}$, $\mathrm{DFR}_{m,\epsilon} = 2^{-22.08}$, $T_{\mathrm{ISD}} = 2^{75.86}$. To determine the concrete complexity of the key recovery attack, an upper bound on $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ must be established. Our experiments suggest that $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ is very close to the average decryption failure rate $\mathrm{DFR}_{\mathrm{avg}}$ for small parameters. Additionally, through extrapolation method it can be deduced that $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)} < 2^{-80}$ for BIKE's parameter set targeting 128-bit security. We refer to Appendix E for further details. Therefore, it can be deduced from Theorem 3 that the total complexity is

$$C_{\mathrm{total}} = 2^{98.77}, \tag{42}$$

where the number of targets required is $1/p_{m,\epsilon} = 2^{76.69}$, and the number of queries required for a single target is $1/(\mathrm{DFR}_{m,\epsilon}) = 2^{22.08}$ (see Table 1 for the details).

7 Conclusion

We propose the gathering property for QC-MDPC and demonstrate its strong correlation with the decryption failure rate. By considering the secret keys satisfying the gathering property, we construct a new set of weak keys by using isomorphisms of the ring $\mathcal{R}$. For BIKE's parameter set targeting 128-bit security, we derive a lower bound on the average decryption failure rate $\mathrm{DFR}_{\mathrm{avg}} \geq 2^{-116.61}$.

We present two multi-target key recovery attacks against QC-MDPC based schemes with CCA security, as well as an analysis of their complexity for BIKE's parameter set targeting 128-bit security. The first attack prohibits ciphertexts reusing and has a complexity of $2^{116.61}$, while the second attack allows ciphertexts reusing and can attain a complexity of $2^{98.77}$.

There are many issues should be addressed in future work. For example, we exclusively consider $\epsilon = 0, 1$ due to the need for rigorous calculations of probability. A natural concern is whether $\epsilon \geq 2$ leads to stronger results on weak keys and better key recovery attacks. Besides, for the sake of simplicity, in our attack the filtered ciphertexts will satisfy $w_H(\mathbf{e_0}) = w_H(\mathbf{e_1}) = t/2$. It is expected that the gathering property performs better for unbalanced error weights, resulting in attacks with improved complexity.

Acknowledgments

We thank the anonymous reviewers from CRYPTO 2023 for the valuable comments. This work is supported by the National Key R&D Program of China (2020YFA0309705, 2018YFA0704701), Shandong Key Research and Development Program (2020ZLYS09), the Major Scientific and Technological Innovation Project of Shandong, China (2019JZZY010133), the Major Program of Guangdong Basic and Applied Research (2019B030302008), and Tsinghua University Dushi Program.

References

1. National institute of standards and technology: Post-quantum cryptography project. https://csrc.nist.gov/projects/post-quantum-cryptography (2016)
2. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Miller, C., Moody, D., Peralta, R., et al.: Status report on the third round of the nist post-quantum cryptography standardization process. US Department of Commerce, NIST (2022)
3. Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Gueron, S., Guneysu, T., Aguilar Melchor, C., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.P., Zémor, G., Vasseur, V., Ghosh, S., Richter-Brokmann, J.: BIKE. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions
4. Aragon, N., Gaborit, P.: A key recovery attack against lrpc using decryption failures. In: Coding and Cryptography, International Workshop, WCC. vol. 2019 (2019)
5. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology – EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 520–536. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4_31
6. Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the inherent intractability of certain coding problems (corresp.). IEEE Trans. Inf. Theory **24**(3), 384–386 (1978). https://doi.org/10.1109/TIT.1978.1055873, https://doi.org/10.1109/TIT.1978.1055873
7. Bindel, N., Schanck, J.M.: Decryption failure is more likely after success. In: Ding, J., Tillich, J.P. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020. pp. 206–225. Springer, Heidelberg, Germany, Paris, France (Apr 15–17, 2020). https://doi.org/10.1007/978-3-030-44223-1_12

8. Blömer, J., May, A.: New partial key exposure attacks on RSA. In: Boneh, D. (ed.) Advances in Cryptology – CRYPTO 2003. Lecture Notes in Computer Science, vol. 2729, pp. 27–43. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2003). https://doi.org/10.1007/978-3-540-45146-4_2
9. Chaulet, J.: Étude de cryptosystèmes à clé publique basés sur les codes MDPC quasi-cycliques. Ph.D. thesis, Paris 6 (2017)
10. Chaulet, J., Sendrier, N.: Worst case QC-MDPC decoder for mceliece cryptosystem. In: IEEE International Symposium on Information Theory, ISIT 2016, Barcelona, Spain, July 10-15, 2016. pp. 1366–1370. IEEE (2016). https://doi.org/10.1109/ISIT.2016.7541522, https://doi.org/10.1109/ISIT.2016.7541522
11. Chou, T.: QcBits: Constant-time small-key code-based cryptography. In: Gierlichs, B., Poschmann, A.Y. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2016. Lecture Notes in Computer Science, vol. 9813, pp. 280–300. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–19, 2016). https://doi.org/10.1007/978-3-662-53140-2_14
12. Coppersmith, D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. Journal of Cryptology **10**(4), 233–260 (Sep 1997). https://doi.org/10.1007/s001459900030
13. D'Anvers, J., Batsleer, S.: Multitarget decryption failure attacks and their application to saber and kyber. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) Public-Key Cryptography - PKC 2022 - 25th IACR International Conference on Practice and Theory of Public-Key Cryptography, Virtual Event, March 8-11, 2022, Proceedings, Part I. Lecture Notes in Computer Science, vol. 13177, pp. 3–33. Springer (2022). https://doi.org/10.1007/978-3-030-97121-2_1, https://doi.org/10.1007/978-3-030-97121-2_1
14. D'Anvers, J.P., Guo, Q., Johansson, T., Nilsson, A., Vercauteren, F., Verbauwhede, I.: Decryption failure attacks on IND-CCA secure lattice-based schemes. In: Lin, D., Sako, K. (eds.) PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II. Lecture Notes in Computer Science, vol. 11443, pp. 565–598. Springer, Heidelberg, Germany, Beijing, China (Apr 14–17, 2019). https://doi.org/10.1007/978-3-030-17259-6_19
15. D'Anvers, J.P., Rossi, M., Virdia, F.: (One) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology – EUROCRYPT 2020, Part III. Lecture Notes in Computer Science, vol. 12107, pp. 3–33. Springer, Heidelberg, Germany, Zagreb, Croatia (May 10–14, 2020). https://doi.org/10.1007/978-3-030-45727-3_1
16. D'Anvers, J.P., Vercauteren, F., Verbauwhede, I.: The impact of error dependencies on ring/mod-LWE/LWR based schemes. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 103–115. Springer, Heidelberg, Germany, Chongqing, China (May 8–10, 2019). https://doi.org/10.1007/978-3-030-25510-7_6
17. den Boer, B., Bosselaers, A.: An attack on the last two rounds of MD4. In: Feigenbaum, J. (ed.) Advances in Cryptology – CRYPTO'91. Lecture Notes in Computer Science, vol. 576, pp. 194–203. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 11–15, 1992). https://doi.org/10.1007/3-540-46766-1_14
18. Drucker, N., Gueron, S., Kostic, D.: On constant-time qc-mdpc decoding with negligible failure rate. Cryptology ePrint Archive (2019)
19. Drucker, N., Gueron, S., Kostic, D.: QC-MDPC decoders with several shades of gray. In: Ding, J., Tillich, J.P. (eds.) Post-Quantum Cryptography - 11th Inter-

national Conference, PQCrypto 2020. pp. 35–50. Springer, Heidelberg, Germany, Paris, France (Apr 15–17, 2020). https://doi.org/10.1007/978-3-030-44223-1_3

20. Drucker, N., Gueron, S., Kostic, D., Persichetti, E.: On the applicability of the fujisaki-okamoto transformation to the BIKE KEM. Int. J. Comput. Math. Comput. Syst. Theory **6**(4), 364–374 (2021). https://doi.org/10.1080/23799927.2021.1930176, https://doi.org/10.1080/23799927.2021.1930176
21. Dumer, I.: On minimum distance decoding of linear codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory. pp. 50–52 (1991)
22. Esser, A., May, A., Verbel, J.A., Wen, W.: Partial key exposure attacks on bike, rainbow and NTRU. In: Dodis, Y., Shrimpton, T. (eds.) Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part III. Lecture Notes in Computer Science, vol. 13509, pp. 346–375. Springer (2022). https://doi.org/10.1007/978-3-031-15982-4_12, https://doi.org/10.1007/978-3-031-15982-4_12
23. Esser, A., May, A., Zweydinger, F.: McEliece needs a break - solving McEliece-1284 and quasi-cyclic-2918 with modern ISD. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part III. Lecture Notes in Computer Science, vol. 13277, pp. 433–457. Springer, Heidelberg, Germany, Trondheim, Norway (May 30 – Jun 3, 2022). https://doi.org/10.1007/978-3-031-07082-2_16
24. Fabsic, T., Hromada, V., Stankovski, P., Zajac, P., Guo, Q., Johansson, T.: A reaction attack on the QC-LDPC McEliece cryptosystem. In: Lange, T., Takagi, T. (eds.) Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017. pp. 51–68. Springer, Heidelberg, Germany, Utrecht, The Netherlands (Jun 26–28, 2017). https://doi.org/10.1007/978-3-319-59879-6_4
25. Fujisaki, E., Okamoto, T.: How to enhance the security of public-key encryption at minimum cost. In: Imai, H., Zheng, Y. (eds.) PKC'99: 2nd International Workshop on Theory and Practice in Public Key Cryptography. Lecture Notes in Computer Science, vol. 1560, pp. 53–68. Springer, Heidelberg, Germany, Kamakura, Japan (Mar 1–3, 1999). https://doi.org/10.1007/3-540-49162-7_5
26. Gallager, R.: Low-density parity-check codes. IRE Transactions on information theory **8**(1), 21–28 (1962)
27. Gama, N., Nguyen, P.Q.: New chosen-ciphertext attacks on NTRU. In: Okamoto, T., Wang, X. (eds.) PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography. Lecture Notes in Computer Science, vol. 4450, pp. 89–106. Springer, Heidelberg, Germany, Beijing, China (Apr 16–20, 2007). https://doi.org/10.1007/978-3-540-71677-8_7
28. Guo, Q., Johansson, T.: A new decryption failure attack against HQC. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 353–382. Springer, Heidelberg, Germany, Daejeon, South Korea (Dec 7–11, 2020). https://doi.org/10.1007/978-3-030-64837-4_12
29. Guo, Q., Johansson, T., Stankovski, P.: A key recovery attack on MDPC with CCA security using decoding errors. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016, Part I. Lecture Notes in Computer Science, vol. 10031, pp. 789–815. Springer, Heidelberg, Germany, Hanoi, Vietnam (Dec 4–8, 2016). https://doi.org/10.1007/978-3-662-53887-6_29

30. Henecka, W., May, A., Meurer, A.: Correcting errors in RSA private keys. In: Rabin, T. (ed.) Advances in Cryptology – CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 351–369. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–19, 2010). https://doi.org/10.1007/978-3-642-14623-7_19
31. Heyse, S., von Maurich, I., Güneysu, T.: Smaller keys for code-based cryptography: QC-MDPC McEliece implementations on embedded devices. In: Bertoni, G., Coron, J.S. (eds.) Cryptographic Hardware and Embedded Systems – CHES 2013. Lecture Notes in Computer Science, vol. 8086, pp. 273–292. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–23, 2013). https://doi.org/10.1007/978-3-642-40349-1_16
32. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the Fujisaki-Okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) TCC 2017: 15th Theory of Cryptography Conference, Part I. Lecture Notes in Computer Science, vol. 10677, pp. 341–371. Springer, Heidelberg, Germany, Baltimore, MD, USA (Nov 12–15, 2017). https://doi.org/10.1007/978-3-319-70500-2_12
33. Horlemann, A.L., Puchinger, S., Renner, J., Schamberger, T., Wachter-Zeh, A.: Information-set decoding with hints. In: Code-Based Cryptography: 9th International Workshop, CBCrypto 2021 Munich, Germany, June 21–22, 2021 Revised Selected Papers. pp. 60–83. Springer (2022)
34. Howgrave-Graham, N., Nguyen, P.Q., Pointcheval, D., Proos, J., Silverman, J.H., Singer, A., Whyte, W.: The impact of decryption failures on the security of NTRU encryption. In: Boneh, D. (ed.) Advances in Cryptology – CRYPTO 2003. Lecture Notes in Computer Science, vol. 2729, pp. 226–246. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2003). https://doi.org/10.1007/978-3-540-45146-4_14
35. Jaulmes, É., Joux, A.: A chosen-ciphertext attack against NTRU. In: Bellare, M. (ed.) Advances in Cryptology – CRYPTO 2000. Lecture Notes in Computer Science, vol. 1880, pp. 20–35. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2000). https://doi.org/10.1007/3-540-44598-6_2
36. Kirshanova, E., May, A.: Decoding mceliece with a hint - secret goppa key parts reveal everything. In: Galdi, C., Jarecki, S. (eds.) Security and Cryptography for Networks - 13th International Conference, SCN 2022, Amalfi, Italy, September 12-14, 2022, Proceedings. Lecture Notes in Computer Science, vol. 13409, pp. 3–20. Springer (2022). https://doi.org/10.1007/978-3-031-14791-3_1, `https://doi.org/10.1007/978-3-031-14791-3_1`
37. von Maurich, I., Güneysu, T.: Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices. In: Mosca, M. (ed.) Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014. pp. 266–282. Springer, Heidelberg, Germany, Waterloo, Ontario, Canada (Oct 1–3, 2014). https://doi.org/10.1007/978-3-319-11659-4_16
38. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology – ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 107–124. Springer, Heidelberg, Germany, Seoul, South Korea (Dec 4–8, 2011). https://doi.org/10.1007/978-3-642-25385-0_6
39. McEliece, R.J.: A public-key cryptosystem based on algebraic. Coding Thv **4244**, 114–116 (1978)
40. Misoczki, R., Tillich, J., Sendrier, N., Barreto, P.S.L.M.: Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In: Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013. pp. 2069–2073. IEEE

(2013). https://doi.org/10.1109/ISIT.2013.6620590, https://doi.org/10.1109/ISIT.2013.6620590

41. Misoczki, R., Tillich, J.P., Sendrier, N., Barreto, P.S.: Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In: 2013 IEEE international symposium on information theory. pp. 2069–2073. IEEE (2013)
42. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Prob. Contr. Inform. Theory **15**(2), 157–166 (1986)
43. Sendrier, N.: Decoding one out of many. In: Yang, B.Y. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011. pp. 51–67. Springer, Heidelberg, Germany, Tapei, Taiwan (Nov 29 – Dec 2 2011). https://doi.org/10.1007/978-3-642-25405-5_4
44. Sendrier, N., Vasseur, V.: On the decoding failure rate of QC-MDPC bit-flipping decoders. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019. pp. 404–416. Springer, Heidelberg, Germany, Chongqing, China (May 8–10, 2019). https://doi.org/10.1007/978-3-030-25510-7_22
45. Sendrier, N., Vasseur, V.: On the existence of weak keys for qc-mdpc decoding. Cryptology ePrint Archive (2020)
46. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. IEEE Computer Society Press, Santa Fe, NM, USA (Nov 20–22, 1994). https://doi.org/10.1109/SFCS.1994.365700
47. Tillich, J.: The decoding failure probability of MDPC codes. In: 2018 IEEE International Symposium on Information Theory, ISIT 2018, Vail, CO, USA, June 17-22, 2018. pp. 941–945. IEEE (2018). https://doi.org/10.1109/ISIT.2018.8437843, https://doi.org/10.1109/ISIT.2018.8437843
48. Vasseur, V.: Post-quantum cryptography: a study of the decoding of QC-MDPC codes. Ph.D. thesis, Université de Paris (2021)
49. Vasseur, V.: Qc-mdpc codes dfr and the ind-cca security of bike. HAL (2022)
50. Zhou, Y., van de Pol, J., Yu, Y., Standaert, F.X.: A third is all you need: Extended partial key exposure attack on crt-rsa with additive exponent blinding. In: Advances in Cryptology–ASIACRYPT 2022: 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5–9, 2022, Proceedings, Part IV. pp. 508–536. Springer (2023)

Supplementary Material

A Proof of Lemma 1

First, we are going to prove that one vector with gathering property (m, ϵ) is calculated no more than twice when $m < r/2$. Let $a_0, a_1, ..., a_{v-1}$ be indexes for the nonzero elements and suppose $0 = a_0 < a_1 < ... < a_{v-2} < m$ and $a_{v-2} < a_{v-1}$. We call a_{v-1}, the element out of gathering area, as out element.

If this vector is counted more than once, the new out element must be a_0 or a_{v-2}. Otherwise, a_0, a_{v-2}, a_{v-1} are in another gathering area whose length longer than $|a_{v-1} - a_{v-2}| + |a_0 - a_{v-1}| \geq r - m > m$ which causes an contradiction. If the new out element is a_0, the begin index of new gathering area is between $[1, a_1]$ thus the end index is between $(m+1, m+a_1)$. So we have $a_{v-1} < m + a_1$. Under this condition, if a_{v-2} is represented as a out element in another way, the end index of new gathering area is larger than a_{v-3}, thus the begin index of new gathering area is larger than $r - m + a_{v-3}$. So we have $a_{v-1} \geq r - m + a_{v-3}$. Then $m + a_1 > r - m + a_{v-3}$ which is in conflict with $2m < r$ and $a_1 < a_{v-3}$. For $\mathrm{E}_{m,\epsilon}(t/2, t/2)$, the proof is same by replacing v with $t/2$. As a result, we have proved that a vector is counted no more than twice.

Let $\mathrm{B}_b = \mathrm{K}_{m,\epsilon}^{(b)}(w) \cap (\bigcup_{i \neq b} \mathrm{K}_{m,\epsilon}^{(i)}(w))$ and $\mathrm{A}_b = \mathrm{K}_{m,\epsilon}^{(b)}(w) - \mathrm{B}_b$. Let p_a denotes to the DFR when $(h_0, h_1) \xleftarrow{\$} \mathrm{A}_b, (e_0, e_1) \xleftarrow{\$} \mathrm{E}_{m,\epsilon}(t/2, t/2)$ and p_b denotes to the DFR when $(h_0, h_1) \xleftarrow{\$} \mathrm{B}_b, (e_0, e_1) \xleftarrow{\$} \mathrm{E}_{m,\epsilon}(t/2, t/2)$.

$$\begin{aligned}
&\mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)\\(e_0,e_1)\xleftarrow{\$}\mathrm{E}_{m,\epsilon}(t/2,t/2)}} && (43)\\
&= \frac{1}{|\mathrm{K}_{m,\epsilon}(w)|}\left(\sum_{b=0}^{r} p_a|A_b| + \frac{1}{2}\sum_{b=0}^{r} p_b|B_b|\right) && (44)\\
&= \frac{rp_a|A_0| + \frac{r}{2}p_b|B_0|}{r|A_0| + \frac{r}{2}|B_0|} && (45)
\end{aligned}$$

On the other hand,

$$\mathrm{DFR}_{\substack{(h_0,h_1)\leftarrow\mathtt{F}(w,m,\epsilon)\\(e_0,e_1)\xleftarrow{\$}\mathrm{E}_{m,\epsilon}(t/2,t/2)}} = \frac{p_a|\mathrm{A}_0| + \frac{1}{2}p_b|\mathrm{B}_0|}{|\mathrm{A}_0| + \frac{1}{2}|\mathrm{B}_0|} = \mathrm{DFR}_{\substack{(h_0,h_1)\xleftarrow{\$}\mathrm{K}_{m,\epsilon}(w)\\(e_0,e_1)\xleftarrow{\$}\mathrm{E}_{m,\epsilon}(t/2,t/2)}} \tag{46}$$

We can also prove

$$\mathrm{DFR}_{\substack{(h_0,h_1)\leftarrow\mathtt{F}(w,m,\epsilon)\\(e_0,e_1)\xleftarrow{\$}\mathrm{E}_{m,\epsilon}(t/2,t/2)}} = \mathrm{DFR}_{\substack{(h_0,h_1)\leftarrow\mathtt{F}(w,m,\epsilon)\\(e_0,e_1)\leftarrow\mathtt{F}(t,m,\epsilon)}} \tag{47}$$

for the same reason. The proof is completed using (46) and (47).

B Proof of Lemma 2

Suppose that $m < r/2$ and $\epsilon = 1$, we aim to prove that

$$0.95 \le \frac{|\mathrm{K}_{m,\epsilon}(w)|}{\tau(w/2, m, \epsilon)} \le 1 \text{ and } 0.95 \le \frac{|\mathrm{E}_{m,\epsilon}(t/2, t/2)|}{\tau(t/2, m, \epsilon)} \le 1 \ , \tag{48}$$

where

$$\tau(x, m, \epsilon) \coloneqq r \sum_{d=x-\epsilon}^{m} \binom{d-2}{x-2-\epsilon}\binom{r-d}{\epsilon}\binom{r}{x} \ . \tag{49}$$

Let $a_0, a_1, ..., a_{v-1}$ be indexes for the nonzero elements and suppose $a_0 < a_1 < ... < a_{v-2} < m$ and $a_{v-2} < a_{v-1}$. We call a_{v-1} as the out element when the indexes are expressed like this. Let $a_0 = 0$ and we enumerate all a_1, a_{v-3}, a_{v-2}. If a vector is counted more than once, either a_0 or a_{v-2} is out of the gathering area in certain expressions according to the proof in Appendix A. If a_0 is out, the new gathering area contains a_1, a_{v-1}. So $a_{v-2} < a_{v-1} < a_1 + m$. If a_{v-1} is out, the new gathering area contains a_{v-1}, a_{v-3}. So $r - (m - a_{v-3}) < a_{v-1} < r$. As a result, a_{v-1} has $(m - a_{v-3} - 1) + (a_1 + m - a_{v-2} - 1)$ available values. Let $\mathrm{B}_b = \mathrm{K}^{(b)}_{m,\epsilon}(w) \cap (\bigcup_{i \ne b} \mathrm{K}_i)$ and $\mathrm{A}_b = \mathrm{K}^{(b)}_{m,\epsilon}(w) - \mathrm{B}_b$.

$$|\mathrm{B}_b| = \sum_{0<a_1, a_{v-3}, a_{v-2}<m} (2m - a_{v-3} + a_1 - a_{v-2} - 2)\binom{a_{v-3} - a_1 - 1}{v-5} \tag{50}$$

As for those in $\mathrm{E}_{m,\epsilon}(t/2, t/2)$, we have similar equation by replacing v with $t/2$. We found $\frac{|\mathrm{B}_b|}{|\mathrm{A}_b|}$ is lower than 5% when $m \le 5500$. When calculating the complexity in theorem 2, the approximate error is lower than

$$|\log_2(0.95)| < 0.08$$

bit. So we can use the approximate equation when counting vectors with gathering property (m, ϵ).

C Proof of Inequality (21)

Denote $v = w/2$, $\tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w) = \mathrm{K}^{\phi_i}_{m,\epsilon}(w) \cap \mathrm{K}^{\texttt{overlap}}_{m,\epsilon}(w)$ and $\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w) = \mathrm{K}^{\phi_i}_{m,\epsilon}(w) - \tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)$. Observe that

$$\frac{|\mathrm{K}^{\texttt{overlap}}_{m,\epsilon}(w)|}{|\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)|} = 1 - \frac{\sum_i |\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)|}{|\mathrm{K}^{\texttt{union}}_{m,\epsilon}(w)|} \tag{51}$$

$$\le 1 - \frac{\sum_i |\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)|}{\sum_i |\mathrm{K}^{\phi_i}_{m,\epsilon}(w)|} \tag{52}$$

$$= 1 - \frac{|\bar{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)|}{|\mathrm{K}^{\phi_i}_{m,\epsilon}(w)|}, \tag{53}$$

thus the left half of (21) holds.

Next we give an upper bound on $|\tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)|$. Our idea is to give upper bound on $|\mathrm{K}^{\phi_i}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_j}_{m,\epsilon}(w)|$, and then use the fact that $|\tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)| \leq (\frac{r-1}{2} - 1) \cdot |\mathrm{K}^{\phi_i}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_j}_{m,\epsilon}(w)|$. By symmetry, we only consider $|\mathrm{K}^{\phi_1}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_i}_{m,\epsilon}(w)|$ for $1 < i < r/2$. The following claim will be used in the estimation.

Claim. Suppose $r = 12323$ and $2900 \leq m \leq 4000$. Then for any affine map $f(x) = ax + b$ defined over $\mathbb{Z}_r$ such that $a, b \in \mathbb{Z}$ and $1 < a < r/2$, it has

$$|f([0, m)) \cap [0, m)| \leq m/2 \ . \tag{54}$$

This claim has been verified by enumerating a, b for $r = 12323$ and all m's involved in our experiment/deduction. For general m and r, it is still unknown whether the claim holds.

Note that $\mathrm{K}^{\phi_1}_{m,\epsilon}(w) = \mathrm{K}_{m,\epsilon}(w) = \cup_{0 \leq b < r} \mathrm{K}^{(b)}_{m,\epsilon}(w)$, where $\mathrm{K}^{(b)}_{m,\epsilon}(w)$ is defined in Section 3.1. Then $|\mathrm{K}^{\phi_1}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_i}_{m,\epsilon}(w)| \leq \sum_{0 \leq b < r} |\mathrm{K}^{(b)}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_i}_{m,\epsilon}(w)|$. Suppose $(\mathbf{h_0}, \mathbf{h_1}) \in \mathrm{K}^{(b)}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_i}_{m,\epsilon}(w)$. Our counting starts by choosing the position $j \in [b, b + m)$ such that

- the j-th position of h_0 is 1,
- the positions $[i^{-1}j, i^{-1}j + m)$ of $\phi_i^{-1}(h_0)$ contains the most possible 1's, (at least $(v - \epsilon)$ 1's).

Then there are at least $(v - 2\epsilon)$ 1's of h_0 in $[b, b + m) \cap \big(i \cdot [i^{-1}j, i^{-1}j + m)\big)$, and by the above claim it has $|[b, b + m) \cap \big(i \cdot [i^{-1}j, i^{-1}j + m)\big)| \leq m/2$. There are two cases for j. First, if $j = b$, the other $(v - 2\epsilon - 1)$ positions must be in $[b, b + m) \cap \big(i \cdot [i^{-1}j, i^{-1}j + m)\big)$, which has at most $\binom{m/2}{v-2\epsilon-1}$ choices. Second, if $j \neq b$, then there are at most $(m - 1)$ choices for j, and at most $\binom{m/2}{v-2\epsilon-2}$ choices for the other $(v - 2\epsilon - 2)$ positions. For the remaining 2ϵ 1's of h_0, there are at most $\binom{r}{2\epsilon}$ choices. It follows that

$$\begin{aligned}&|\mathrm{K}^{(b)}_{m,\epsilon}(w) \cap \mathrm{K}^{\phi_i}_{m,\epsilon}(w)| \\ \leq&\rho(w, m, \epsilon) \coloneqq \binom{v - \epsilon}{v - 2\epsilon}\left[\binom{m/2}{v - 2\epsilon - 1} + (m - 1)\binom{m/2}{v - 2\epsilon - 2}\right]\binom{r}{2\epsilon}\binom{r}{v}.\end{aligned}$$

Then is follows directly that

$$|\tilde{\mathrm{K}}^{\phi_i}_{m,\epsilon}(w)| \leq (\frac{r - 1}{2} - 1)r \cdot \rho(w, m, \epsilon), \tag{55}$$

On the other hand, by Lemma 2,

$$|\mathrm{K}_{m,\epsilon}(w)| \geq (1 + \xi) \cdot \tau(w/2, m, \epsilon). \tag{56}$$

Thus δ can be set to be

$$\delta = \frac{r(r - 3)}{2(1 + \xi)} \cdot \frac{\rho(w, m, \epsilon)}{\tau(w/2, m, \epsilon)}. \tag{57}$$

For $\epsilon = 1, m = 4000$, it can be calculated that $\delta \leq 2^{-35}$. For $m \in [2900, 4000)$, δ is even smaller, which is negligible with respect to our simulated decryption failure rates.

D Black Gray Flipping

Algorithm 3: Black Gray Flipping

```
Input: $\mathbf{H} \in \mathbb{F}_2^{r \times n}, \mathbf{s} \in \mathbb{F}_2^r$
Output: A vector $\mathbf{e}$ such that $\mathbf{He} = \mathbf{s}$
$\mathbf{e} \leftarrow 0^n$
for i=1,2,...,NbIter do
    $T$ =threshold($w_H(\mathbf{s} + \mathbf{He})$,i)
    $\mathbf{e}, black, gray \leftarrow BFIter(\mathbf{s} + \mathbf{He}, \mathbf{e}, T, \mathbf{H})$
    if $i = 1$ then
        $\mathbf{e} \leftarrow BFMaskedIter(\mathbf{s} + \mathbf{He}, \mathbf{e}, black, (v + 3)/2, \mathbf{H})$
        $\mathbf{e} \leftarrow BFMaskedIter(\mathbf{s} + \mathbf{He}, \mathbf{e}, gray, (v + 3)/2, \mathbf{H})$
    end if
end for
if $\mathbf{He} = \mathbf{s}$ then
    return $\mathbf{e}$
else
    return $\perp$
end if
procedure BFIter($\mathbf{s}$,$\mathbf{e}$,T,$\mathbf{H}$)
for j=0,1,...,n-1 do
    if $|\mathbf{s} \cap \mathbf{h_j}| \geq T$ then
        $\mathbf{e}_j \leftarrow \boldsymbol{e}_j + 1$
        $black_j \leftarrow 1$
    else if $|\mathbf{s} \cap \mathbf{h_j}| \geq T - \tau$ then
        $gray_j \leftarrow 1$
    end if
    return $\mathbf{e}, black, gray$
end for
procedure BFMaskedIter($\mathbf{s}$,$\mathbf{e}$,mask,T,$\mathbf{H}$)
for j=0,1,...,n-1 do
    if $|\mathbf{s} \cap \mathbf{h_j}| \geq T$ then
        $\mathbf{e}_j \leftarrow \mathbf{e}_j + mask_j$
    end if
end for
return $\mathbf{e}$
```

E Experiments under Small Parameters

According to the deduction in Section 6.1, it has

$$p_{\mathtt{true}} = \mathrm{DFR}_{m,\epsilon} \cdot p_{m,\epsilon} / \mathrm{DFR}_{\mathrm{avg}}^{e \sim (m,\epsilon)}. \tag{58}$$

Table 7. The parameters in the BGF algorithm.

Security	NbIter	τ	threshold(S,i)
Level 1	5	3	$max(\lfloor 0.0069722S + 13.530 \rfloor, 36)$
Level 3	5	3	$max(\lfloor 0.005265S + 15.2588 \rfloor, 52)$
Level 5	5	3	$max(\lfloor 0.00402312S + 17.8785 \rfloor, 69)$

As $p_{m,\epsilon}$ and $\mathrm{DFR}_{m,\epsilon}$ have already been determined through previous deductions or experiments, the estimation of p_{true} is effectively identical to estimating $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$. Our experimental design focuses on BIKE's parameter set targeting 128-bit security, but we opt to measure $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ directly by setting $10009 \leq r \leq 10179$. We conduct two sets of experiments. The first set measures both $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ and $\mathrm{DFR}_{\mathrm{avg}}$, which allows us to demonstrate that these values are close, indicating that $p_{\mathtt{true}}$ is not excessively small. The second set directly tests $p_{\mathtt{true}}$ under small parameters and observes the resulting trend.

Table 8. The $\mathrm{DFR}_{\mathrm{avg}}$ and $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ for $m = 6000$ and $\epsilon = 0$. The data are displayed both in fractions and exponents.

r	10009	10019	10029	10039
$\mathrm{DFR}_{\mathrm{avg}}$	$\frac{50}{300000}$ $(2^{-12.55})$	$\frac{80}{700000}$ $(2^{-13.10})$	$\frac{188}{2059468}$ $(2^{-13.42})$	$\frac{50}{800000}$ $(2^{-13.97})$
$\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$	$\frac{160}{1120506}$ $(2^{-12.77})$	$\frac{84}{700000}$ $(2^{-13.02})$	$\frac{183}{2363499}$ $(2^{-13.66})$	$\frac{61}{900000}$ $(2^{-13.85})$

r	10049	10059	10069
DFR_{avg}	$\frac{77}{1700000}$ $(2^{-14.43})$	$\frac{100}{3602726}$ $(2^{-15.14})$	$\frac{100}{4029708}$ $(2^{-15.30})$
$\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$	$\frac{74}{1600000}$ $(2^{-14.40})$	$\frac{100}{3356234}$ $(2^{-15.03})$	$\frac{100}{4100118}$ $(2^{-15.32})$

E.1 Experiments for $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$

For $r = 12323$, we have observed in Section 4 that weak keys exhibiting the gathering property result in a high decryption failure rate. However, when considering errors that have the gathering property but random keys, we discovered that the resulting decryption failure rate is too low to detect. Despite conducting numerous experiments in the range of $2000 \leq m \leq 3500$, we have yet to identify any failures. For instance, when $(m, \epsilon) = (3100, 1)$, we do not encounter any failure while conducting 2^{25} decryptions, which is significantly lower than the $\mathrm{DFR} = 2^{-18.38}$ we identified for weak keys. Even with $m = 2000$, we note that the decryption failure rate with keys satisfying the gathering property is approximately 0.1%, while the decryption failure rate with errors satisfying the gathering property remains below 2^{-25}.

On the other hand, for $10009 \leq r \leq 10069$ we estimate $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ and $\mathrm{DFR}_{\mathrm{avg}}$ and present the values in Table 8. Through the table we can infer that the discrepancy between $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ and $\mathrm{DFR}_{\mathrm{avg}}$ is very small. We believe that this fact is also true for $r = 12323$.

E.2 Experiments for $p_{\texttt{true}}$

For $10129 \le r \le 10179$ we simulated the value of $p_{\texttt{true}}$, and the results are listed in Table 9. As r increases, there is a clear downward trend for $\log_2(p_{\texttt{true}}^{-1})$, exhibiting strong linearity. This encouraged us to employ linear regression to compute the value of $p_{\texttt{true}}$ for larger r. Specifically, we were able to fit the data using the equation $y = -0.030x + 350.364$ with a related coefficient of 0.985. Hence, we estimated that $p_{\texttt{true}} \approx 2^{-18.59}$ when $r = 12323$.

Table 9. The $p_{m,\epsilon}$, $\mathrm{DFR}_{m,\epsilon}$ and $\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$ for $(m,\epsilon) = (5100, 1)$. The data is provided in its logarithmic form.

r	$\mathrm{DFR}_{m,\epsilon}$	$\mathrm{DFR}_{\mathrm{avg}}^{e\sim(m,\epsilon)}$	$p_{m,\epsilon}$	$p_{\texttt{true}}^{-1}$
10129	-7.87	-18.11	-57.35	47.11
10139	-8.05	-18.59	-57.44	46.90
10149	-8.31	-19.46	-57.54	46.40
10159	-8.48	-20.04	-57.64	46.07
10169	-8.66	-20.39	-57.73	46.01
10179	-8.81	-21.03	-57.83	45.61