# Weak keys in QC-MDPC-based cryptosystems via the extended Euclidean algorithm

Alessio Meneghetti[1,2†] and Federica Zanetti[1,3*†]

[1]Department of Mathematics, University of Trento, Trento, Italy.
[2]Department of Mathematics, University of Bari Aldo Moro, Bari, Italy.
[3]Center for Cybersecurity, Fondazione Bruno Kessler, Trento, Italy.

*Corresponding author(s). E-mail(s): fzanetti@fbk.eu;
Contributing authors: alessio.meneghetti@uniba.it;
†These authors contributed equally to this work.

**Abstract**

In this work we analyze a problem strongly linked to rational reconstruction, which forms the foundation of some post-quantum Quasi-Cyclic Moderate-Density Parity-Check and Quasi-Cyclic Low-Density Parity-Check code-based schemes such as LEDAkem and BIKE. Given a polynomial in a cyclic ring as input, our aim is to recover two polynomials with specific properties, whose ratio is the input one. The starting point of this work is the paper of Bardet, Dragoi, Luque, and Otmani, which describes some approaches based on the Extended Euclidean Algorithm, that solves this problem in some specific cases.
In comparison to previous work, we define an additional setting in which the problem can be solved. We also provide an alternative approach to estimate the probability of success, by taking into account a requirement that was not considered in the original paper, thus getting a more precise estimation. Finally, we present a key-recovery attack on BIKE, evaluate its computational cost, and compare it with that of the most efficient known attacks. Although this last step is performed specifically on BIKE, the methodology can be extended to other schemes as well.

**Keywords:** Code-based cryptography, Cryptanalysis, Extended Euclidean Algorithm, BIKE

**MSC Classification:** 11A05 , 11T71 , 68W40 , 94A60 , 94B60

# 1 Introduction

In the 1960s, Gallager [1] introduced the family of Low-Density Parity-Check (LDPC) codes, that have a simple property: they admit a sparse parity-check matrix. This means that the Hamming weight of each of its rows is approximately $\log(n)$, where $n$ is the length of the code. Similar to LDPC codes, Moderate-Density Parity-Check (MDPC) codes were first introduced in the context of communication [2] as codes admitting a parity-check matrix whose rows have Hamming weight approximately $\sqrt{n}$.

In public-key cryptosystems, LDPC and MDPC codes are often adopted due to their efficient decoding procedures [1, 3]. In order to obtain smaller keys, these codes are usually combined with a Quasi-Cyclic (QC) structure, namely, they admit quasi-circulant sparse parity-check matrices. Since there exists an isomorphism between the set of circulant matrices of order $r$ over the finite field with $q$ elements $\mathbb{F}_q$ and the ring of polynomials $\mathbb{F}_q[x]/(x^r - 1)$, a QC-LDPC (resp. QC-MDPC) code can be represented by a sequence of low-density (resp. moderate-density) polynomials. In the most common case, which is also the one we consider, this sequence consists of exactly a pair of polynomials. Some of the schemes proposed for the NIST standardization process were based on these types of codes [4, 5].

While these schemes rely on codes with a defined algebraic structure, the most effective known attacks on QC-LDPC and QC-MDPC code-based schemes generally do not focus on this structure, but instead treat the codes as random. Information Set Decoding (ISD) is a family of algorithms that represents the most prominent non-structural attack method used against code-based cryptosystems. Starting from Prange's original ISD algorithm [6], proposed in 1962, numerous variants have since been introduced [7–10], among which one of the most effective is the algorithm by May and Ozerov [11].

In this work, we focus instead on a structural attack that can be run against schemes based on QC-LDPC or QC-MDPC codes, with particular attention to BIKE [4]. The approach underlying this structural attack was first presented in 2016 by Bardet et al. [12]. The authors proposed a method based on the Extended Euclidean Algorithm (EEA) which, by exploiting the relationship between the private and public keys, can recover the private key if it satisfies certain properties. Starting from this approach, they firstly identify some weak keys retrievable in this way and they secondly propose to extend the collection of weak keys using some group actions. In this way they define three sets of weak private keys that can be reconstructed with three methods all based on the EEA. They provide a formula that describes the probability that a private key inside each set is chosen.

Our contribution is the identification of a new set of weak private keys, different from those defined in [12], whose elements can be recovered with the lowest computational-cost strategy. This improves the overall efficiency of the key-recovery attack. Moreover, when defining the weak key sets, we introduce an additional condition not mentioned in the original paper, which is, however, necessary for a successful key recovery. Due to this additional study, we obtain a more precise cost estimate of the attacks in [12]. Furthermore we present a new approach for computing the cardinality of the weak key sets, different from the one presented in [12].

## 2 A variation of the rational reconstruction problem

Throughout this paper, vectors are meant as row vectors unless otherwise specified, i.e. if $v$ is a vector then $v^\top$ is a column vector.
Let $r$ be a positive prime integer. We denote with $\mathbb{Z}_r$ the quotient ring $\mathbb{Z}/r\mathbb{Z}$, with $\mathbb{F}_2$ the finite field with 2 elements, and with $\mathcal{R}$ the cyclic polynomial ring $\mathbb{F}_2[x]/(x^r-1)$. For a polynomial $f(x) = a_0 + a_1 x + \dots a_{r-1}x^{r-1} \in \mathcal{R}$, we denote with $|f|$ the Hamming weight of the vector $(a_0, a_1, \dots, a_{r-1}) \in \mathbb{F}_2^r$ composed of the coefficients of $f(x)$.

Throughout this work, we consider search problems and evaluate the computational cost of algorithms and problems under the worst-case complexity model. In this context, an algorithm is said to be efficient if it runs in polynomial time in the input size.
The problem that we are interested in is a variation of the rational reconstruction problem, formally stated in Problem 1.

**Problem 1** *Let $h_0$ and $h_1$ be two polynomials in $\mathcal{R}$ such that $|h_0| = w_0$ and $|h_1| = w_1$ hold for two odd integers $w_0$ and $w_1$ with $1 < w_0 + w_1 < r$. Let $h = h_1 h_0^{-1} \in \mathcal{R}$.*
*On input $h$ determine $h_0$ and $h_1$.*

We remark that Problem 1 might have multiple valid solutions, i.e. there might also exist $\bar{h}_0 \neq h_0$, $\bar{h}_1 \neq h_1 \in \mathcal{R}$ with $\bar{h}_0, \bar{h}_1 \neq 0$ such that $|\bar{h}_0| = w_0$, $|\bar{h}_1| = w_1$ with $h = \bar{h}_1 \bar{h}_0^{-1} \in \mathcal{R}$. Whenever we are given an instance of Problem 1 where $h$ is known to have a solution $(h_0, h_1)$ we are thus interested in finding any solution $(\bar{h}_0, \bar{h}_1)$ regardless of it being equal to $(h_0, h_1)$.

In some specific cases Problem 1 can be solved using a method based on the Extended Euclidean Algorithm (described in Algorithm 1). On input $(x^r - 1, h)$, at each step the EEA will generate a tuple $(t_i, r_i)$ such that

$$t_i h \equiv r_i \mod (x^r - 1) .$$

When the polynomials $h_0$ and $h_1$ satisfy certain properties, Problem 1 can be efficiently solved by simply checking whether the partial output $(t_i, r_i)$ is a valid solution for Problem 1. The main difficult task is to define and estimate precisely the number of polynomials $h_0, h_1$ that can be recovered through the EEA.

In this work we focus on the variation of Problem 1 with $w_0 = w_1 = d$, and we identify new sets for which the latter can be efficiently solved. The following useful and well-known theorem, whose variants are often of central importance in decoding algorithms, see e.g. [13], represents the foundation of the approach.

**Theorem 1** *Let $a$ and $b$ be two nonzero polynomials in $\mathbb{F}_2[x]$ such that $\deg(a) \geq \deg(b)$. Let $t$ and $r$ be two nonzero polynomials in $\mathbb{F}_2[x]$ such that:*

1. $\gcd(r, t) = 1$;
2. $\deg(t) + \deg(r) < \deg(a)$;
3. $tb \equiv r \mod a$.

**Algorithm 1:** Extended Euclidean Algorithm (EEA) over $\mathbb{F}_2[x]$

**Input:** Two nonzero polynomials $a, b$ in $\mathbb{F}_2[x]$ such that $\deg(a) \geq \deg(b)$
**Output:** The $\gcd(a, b)$

$(r_0, r_1) \leftarrow (a, b)$, $(s_0, t_0) \leftarrow (1, 0)$, $(s_1, t_1) \leftarrow (0, 1)$;
$i \leftarrow 1$;
**while** $r_i \neq 0$ **do**
  $q_i \leftarrow r_{i-1}$ div $r_i$; `// div = integer division`
  $r_{i+1} \leftarrow r_{i-1} - q_i r_i$;
  $s_{i+1} \leftarrow s_{i-1} - q_i s_i$;
  $t_{i+1} \leftarrow t_{i-1} - q_i t_i$;
  $i \leftarrow i + 1$;
$\nu \leftarrow i - 1$;
**return** $r_\nu$

*Then, by computing the Extended Euclidean Algorithm, there exists an index $i \in \{0, \ldots, \nu\}$ such that the intermediate tuple $(s_i, t_i, r_i)$ satisfies $t = t_i$ and $r = r_i$.*

**Definition 1** A *weak polynomial pair* is a pair of polynomials $(h_0, h_1)$ such that Problem 1 can be solved efficiently via EEA-based methods when given $h := h_1 h_0^{-1}$ as input.

Let $h_0$ and $h_1$ be two polynomials in $\mathcal{R}$ such that $|h_0| = |h_1| = d$, with $d$ an odd integer, and let $h = h_0^{-1} h_1$. Note that, in this setting, $h_0$ and $h_1$ are both invertible in $\mathcal{R}$, and $h \in \mathcal{R}$ is well-defined. If we consider the polynomials in $\mathbb{F}_2[x]$, then there exists a polynomial $q \in \mathbb{F}_2[x]$ such that

$$q(x^r - 1) + h_0 h = h_1.$$

Our goal is to solve Problem 1 on input $h$ obtained as above, namely, to determine either the polynomials $h_0$ and $h_1$ or two alternative polynomials $\bar{h}_0$ and $\bar{h}_1$ in $\mathcal{R}$, such that $|\bar{h}_0| = |\bar{h}_1| = d$ and $h = \bar{h}_0^{-1} \bar{h}_1$ in $\mathcal{R}$. To achieve our purpose we consider two cases in which the polynomials $h_0$ and $h_1$ have specific properties that can be exploited to successfully apply Theorem 1.

***Case 1:***

There exist an $i \in \mathbb{Z}_r$ and two polynomials $\tilde{h}_0$ and $\tilde{h}_1$, that are coprime in $\mathbb{F}_2[x]$ and satisfy $\deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r$, such that

$$(h_0, h_1) = (x^i \tilde{h}_0 \mod (x^r - 1),\ x^i \tilde{h}_1 \mod (x^r - 1)).$$

In this scenario, the shift $x^i$ applied to $\tilde{h}_0$ and $\tilde{h}_1$ disappears when $h$ is computed, indeed,

$$h \equiv h_0^{-1} h_1 \mod (x^r - 1) \equiv (x^i \tilde{h}_0)^{-1} x^i \tilde{h}_1 \mod (x^r - 1) \equiv \tilde{h}_0^{-1} \tilde{h}_1 \mod (x^r - 1).$$

Consequently, a solution can be found by applying the Modified Extended Euclidean Algorithm (MEEA) presented in Algorithm 2 on input $x^r - 1$, $h$ and $d$.
Let us call $K_1$ the set of pairs of polynomials with these properties, i.e.

$$K_1 := \{(x^i \tilde{h}_0 \mod (x^r - 1),\ x^i \tilde{h}_1 \mod (x^r - 1)) \in \mathbb{F}_2[x]^2 \mid i \in \mathbb{Z}_r, \\ \gcd(\tilde{h}_0, \tilde{h}_1) = 1,\ \deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r,\ |\tilde{h}_0| = |\tilde{h}_1| = d\}. \quad (1)$$

In [12], the authors did not identify this set, but they only define the following base set

$$W_1 := \{(h_0, h_1) \in \mathcal{R}^2 \mid |h_i| = w_i,\ w_i \text{ odd, and } \deg(h_0) + \deg(h_1) < r\} \quad (2)$$

and provide a precise formula that describes its cardinality (Equation (12) in [12]) and the probability that a solution $(h_0, h_1)$ of a random instance $h$ lies inside this set.

Note that $W_1$ is approximately the subset of $K_1$ corresponding to the sole case $i = 0$, so they did not consider that an equal shift applied to both polynomials disappears when $h$ is computed. Hence, we define a broader set of weak keys that can be retrieved using Algorithm 2. Moreover, in $K_1$ there is the coprimality constraint that is not considered in $W_1$.

**Algorithm 2:** Modified Extended Euclidean Algorithm (MEEA) over $\mathbb{F}_2[x]$

**Input:** Two nonzero polynomials $a, b$ in $\mathbb{F}_2[x]$ such that $\deg(a) \geq \deg(b)$ and an integer $d > 0$
**Output:** Two polynomials $(r_i, t_i)$ with $|r_i| = |t_i| = d$, such that $r_i \equiv t_i b \mod a$, or `failure`

```
(r_0, r_1) ← (a, b), (s_0, t_0) ← (1, 0), (s_1, t_1) ← (0, 1);
i ← 1;
while r_i ≠ 0 do
    if |r_i| = |t_i| = d then
        return (r_i, t_i);
    q_i ← r_{i-1} div r_i;
    r_{i+1} ← r_{i-1} − q_i r_i;
    s_{i+1} ← s_{i-1} − q_i s_i;
    t_{i+1} ← t_{i-1} − q_i t_i;
    i ← i + 1;
return failure
```

***Case 2:***

There exist two integers $i, j \in \mathbb{Z}_r$ and two polynomials $\tilde{h}_0$, $\tilde{h}_1$ coprime in $\mathbb{F}_2[x]$ with $\deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r$, such that

$$(h_0, h_1) = (x^i \tilde{h}_0 \mod (x^r - 1),\ x^j \tilde{h}_1 \mod (x^r - 1)).$$

In this case the shifts of the two polynomials are not directly visible in $h$, but their difference remains. More precisely

$$h \equiv h_0^{-1} h_1 \mod (x^r - 1) \equiv (x^i \tilde{h}_0)^{-1} x^j \tilde{h}_1 \mod (x^r - 1) \equiv x^{j-i} \tilde{h}_0^{-1} \tilde{h}_1 \mod (x^r - 1)$$

$$\implies x^{i-j} h \equiv \tilde{h}_0^{-1} \tilde{h}_1 \mod (x^r - 1).$$

Consequently, $\tilde{h}_0$ and $\tilde{h}_1$ can be found by applying the MEEA on each shift of $h$. Once an output different from `failure` is given, what we get are two polynomials $\tilde{h}_0$ and $\tilde{h}_1$ of weight $d$ such that, for a certain $k$,

$$\tilde{h}_1 \equiv \tilde{h}_0 x^k h \mod (x^r - 1).$$

At this point, by defining $\bar{h}_0 := \tilde{h}_0 x^k \mod (x^r - 1)$ and $\bar{h}_1 := \tilde{h}_1 \mod (x^r - 1)$, we obtain

$$h \equiv \bar{h}_0^{-1} \bar{h}_1 \mod (x^r - 1), \qquad \text{with } |\bar{h}_1| = |\bar{h}_2| = d.$$

The pairs that fall into this case are thus given by the following set

$$K_2 := \{(x^i \tilde{h}_0 \mod (x^r - 1),\ x^j \tilde{h}_1 \mod (x^r - 1)) \in \mathbb{F}_2[x]^2 \mid i, j \in \mathbb{Z}_r, \\ \gcd(\tilde{h}_0, \tilde{h}_1) = 1,\ \deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r,\ |\tilde{h}_0| = |\tilde{h}_1| = d\}. \quad (3)$$

It is easy to note that $K_1 \subset K_2$. However, unlike the set $K_1$, whose elements can be found by using Algorithm 2 on input $(x^r - 1, h, d)$, the elements of $K_2$ require the application of the same algorithm several times on different shifts of $h$. The latter is presented in detail in Algorithm 3.

In [12], the authors identify the following set

$$W_2 := \{(x^i h_0, x^j h_1) \in \mathcal{R}^2 \mid i, j \in \mathbb{Z}_r,\ |h_i| = w_i,\ w_i \text{ odd},\ \deg(h_0) + \deg(h_1) < r\}, \quad (4)$$

and provide a precise formula (Equation (19) in [12]) for the probability that a solution to a random instance $h$ lies inside $W_2$. We can note that $W_2$ is a generalization of $K_2$, in which the coprimality condition is omitted. Thus, our contribution in this case is to incorporate this condition as a necessary requirement. Note that, without the coprimality assumption, the algorithm fails to determine $h_0$ and $h_1$, and this fact leads to an overestimation of the probability (Equation (19) of [12]) of selecting instances that are solvable and consequently to an underestimation of the cost of the attack.

**Algorithm 3:** MEEA over $\mathbb{F}_2[x]$ for different shifts

**Input:** Two nonzero polynomials $a, b$ in $\mathbb{F}_2[x]$ such that $\deg(a) \geq \deg(b)$ and an integer $d > 0$

**Output:** Two polynomials $(r_i, t_i)$ and an integer $k \in \{0, \ldots, r-1\}$ such that $|r_i| = |t_i| = d$ and $r_i \equiv t_i b x^k \mod a$, or `failure`

```
for i ← 0 to r − 1 do
    b_shifted ← x^i b mod a;
    res ← Algorithm 2(a, b_shifted, d);
    if res ≠ failure then
        return (res, i);
return failure
```

To summarize, we have just described two sets for which Problem 1 can be solved. If $(h_0, h_1) \in K_1$, then the solution can be found by running Algorithm 2 on input $(x^r - 1, h, d)$. Otherwise, if $(h_0, h_1) \in K_2$, Algorithm 3 has to be run on the same input $(x^r - 1, h, d)$.

# 3 Complexity analysis

We recall that all complexity estimates are given with respect to the worst-case complexity model.
Both Algorithm 2 and 3 rely on the EEA. Let $a$ and $b$ be the two polynomials in $\mathbb{F}_2[x]$ given as input to the EEA, with $\deg(a) > \deg(b)$. At every step of the algorithm the most expensive operation computed is the division between the two polynomials $r_i$ and $r_{i+1}$. Since $\deg(r_{i+1}) < \deg(r_i)$, the time complexity of each step can be estimated as $\mathcal{O}\big(\deg(r_i)(\deg(r_i) - \deg(r_{i+1}))\big)$. Indeed the division algorithm performs $\deg(r_i) - \deg(r_{i+1})$ iterations and each iteration involves approximately $\deg(r_i)$ arithmetic operations.
Thus, the overall complexity $\mathcal{C}_{\text{EEA}}$ of the algorithm is obtained by multiplying the complexity per step by the total number of steps. More formally

$$
\begin{aligned}
\mathcal{C}_{\text{EEA}} &= \mathcal{O}\left(\sum_{i=0}^{\nu-1} \deg(r_i)(\deg(r_i) - \deg(r_{i+1}))\right) \\
&\approx \mathcal{O}\left(\deg(r_0)\left(\sum_{i=0}^{\nu-1} \deg(r_i) - \deg(r_{i+1})\right)\right) \\
&= \mathcal{O}\big(\deg(r_0)[(\deg(r_0) - \deg(r_1)) + (\deg(r_1) - \deg(r_2)) + \ldots \\
&\qquad\qquad + (\deg(r_{\nu-1}) - \deg(r_\nu))]\big) \\
&= \mathcal{O}\big(\deg(r_0)[\deg(r_0) - \deg(r_\nu)]\big) \\
&\approx \mathcal{O}\big((\deg(r_0))^2\big) = \mathcal{O}\big((\deg(a))^2\big).
\end{aligned}
$$

We remark that the difference between the EEA and Algorithm 2 is the adoption of a new stopping condition. In particular Algorithm 2 is run with input a polynomial $a$ of degree $r$ and a polynomial $b$ of degree smaller than $r$. Therefore, the complexity is

$$\mathcal{C}_{\text{Algorithm 2}} = \mathcal{O}\big((\deg(a))^2\big) = \mathcal{O}(r^2).$$

In the same way, Algorithm 3 requires in our scenario at most $r$ iterations of Algorithm 2. Consequently, its complexity is

$$\mathcal{C}_{\text{Algorithm 3}} = \mathcal{O}\big(r(\deg(a))^2\big) = \mathcal{O}(r^3).$$

# 4 Probability of selecting weak polynomials

Let $d$ be an odd integer such that $1 < d < \frac{r}{2}$, then we can define the set $\mathcal{P}_d$ as

$$\mathcal{P}_d := \{(h_0, h_1) \in \mathcal{R} \mid |h_0| = |h_1| = d\}.$$

Considering that the pair of polynomials $(h_0, h_1)$ is drawn uniformly at random from $\mathcal{P}_d$, in this section we study the probability of selecting a pair for which Problem 1 can be solved via the two described approaches.

Our strategy involves, as first step, calculating the cardinalities of the sets $K_1$ and $K_2$, defined respectively in Equation (1) and Equation (3). Secondly, we compute the ratio between these cardinalities and the total number of possible pairs of polynomials for Problem 1.

In order to calculate the cardinality of $K_1$ and $K_2$, we proceed as follows. First, we relax the coprimality condition used to define these sets and we define in this way two modified sets: $\mathcal{W}_1$ and $\mathcal{W}_2$.

$$\mathcal{W}_1 := \{(x^i\tilde{h}_0,\ x^i\tilde{h}_1) \in \mathcal{R}^2 \mid \deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r,\ |\tilde{h}_0| = |\tilde{h}_1| = d,\ i \in \mathbb{Z}_r\}, \quad (5)$$

$$\mathcal{W}_2 := \{(x^i\tilde{h}_0,\ x^j\tilde{h}_1) \in \mathcal{R}^2 \mid \deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r,\ |\tilde{h}_0| = |\tilde{h}_1| = d,\ i, j \in \mathbb{Z}_r\}. \quad (6)$$

Then, we compute the cardinality of the modified sets by using combinatorial methods. In a second phase, in order to estimate the cardinality of the original sets, we perform a statistical analysis to determine how many pairs within the modified sets satisfy the relaxed coprimality condition.

The probability of selecting a pair of polynomials $(h_0, h_1)$, for which we can solve Problem 1 through the first and second approach is, respectively,

$$\mathbb{P}_1 := \frac{|K_1|}{|\mathcal{P}_d|} \qquad \text{and} \qquad \mathbb{P}_2 := \frac{|K_2|}{|\mathcal{P}_d|}, \quad (7)$$

where $|\mathcal{P}_d| = \binom{r}{d}^2$.

*Remark 1* Note that the set $\mathcal{W}_2$ is simply $W_2$ restricted to the case $w_0 = w_1 = d$. On the other hand, $\mathcal{W}_1$ is not a restriction of $W_1$, since $\mathcal{W}_1$ also takes into account shifts.

Before entering the details of the analysis, let us state an important theorem, presented in [14], that will be needed later and a definition.

**Theorem 2** *Let $n, k, l \in \mathbb{N}^+$. The number of ways to split $n$ balls into $l$ bins such that the maximum number of balls inside each bin is at most $k$, is equal to*

$$R(n,l,k) := \sum_{t=0}^{l} (-1)^t \binom{l}{t} \binom{n - t(k+1) + l - 1}{l-1}.$$

A register is a finite sequence of bits. From now on, whenever we talk about a register without specifying the length, we mean a length-$r$ sequence of bits.

**Definition 2** Consider a register. A *run of ones* is a sequence of consecutive ones both preceded and followed by a zero, while a *run of zeros* is a sequence of consecutive zeros preceded and followed by a one.

Let us start from case 2, followed by case 1.

## 4.1 Weak keys for Attack 2

### 4.1.1 Cardinality of the set $\mathcal{W}_2$

We first introduce some lemmas that will be used in the proofs of the main results.

**Lemma 1** *Let $r$ and $d$ be two positive integers with $d < r$. The number of ways in which $d$ ones and $r - d$ zeros can be distributed into a circular register of length $r$ with a fixed starting position, given by $\binom{r}{d}$, can be rewritten as*

$$r \sum_{v=1}^{d} \sum_{\substack{(j_1,\ldots,j_v)\in\mathbb{N}^v \mid \forall i \\ j_i>0,\ j_1+\cdots+j_v=r-d}} \binom{d-1}{v-1} \frac{1}{v},$$

*or equivalently*

$$r \sum_{v=1}^{d} \sum_{\substack{(w_1,\ldots,w_v)\in\mathbb{N}^v \mid \\ w_1+\cdots+w_v=r-d-v}} \binom{d-1}{v-1} \frac{1}{v},$$

*where $v$ represents the number of runs of ones, and $(j_1, \ldots, j_v)$ represents the lengths of the zero runs.*

*Proof* Consider a circular register of length $r$, with a fixed starting position, and suppose to have $d$ ones and $r - d$ zeros to place inside this register. For now we place a one in the first

position of the register and a zero in the last. In a second phase we will consider all the shifts of every configuration.

Let us focus on the possible number of runs of ones. Since there are a total of $d$ ones, the number of runs is between 1 and $d$. Note that, the starting point of each run is characterized by a one preceded by a zero. Let $v$ in $\{1,\dots,d\}$ be the number of runs of ones. Note that, among these $v$ ones preceded by a zero, we also count the one in the first position as being preceded by the zero in the last position (since we work with a circular register). Thus, we are left with $d-v$ ones to distribute among the runs. The number of ways to do this can be easily determined using the stars and bars method, and it is

$$\binom{(d-v)+(v-1)}{v-1} = \binom{d-1}{v-1}.$$

To avoid counting the same configuration multiple times, when considering the shift, we divide the just computed value by $v$. So, the remaining $d-v$ ones could be distributed in

$$\binom{d-1}{v-1}\frac{1}{v}$$

ways.

At this point, we can focus on how to distribute the zeros. For a fixed number $v$ of runs of ones, there are also $v$ runs of zeros. We are interested not only in the number of ways to distribute the $r-d-v$ remaining zeros, but we also want to know the length of each run. Let the tuple $(j_1,\dots,j_v)$ represent the lengths of the $v$ runs of zeros, then the following sum describes the number of ways in which the $r-d-v$ zeros could be distributed among the runs.

$$\sum_{\substack{(j_1,\dots,j_v)\in\mathbb{N}^v \mid \ \forall i\\ j_i>0,\ j_1+\cdots+j_v=r-d}} 1 = \sum_{\substack{(w_1,\dots,w_v)\in\mathbb{N}^v \mid \\ w_1+\cdots+w_v=r-d-v}} 1.$$

Summarizing, with the constraint that in the first position of the register there is a one, in the last position there is a zero, and without counting the shifts, the number of ways in which $d$ ones and $r-d$ zeros could be distributed over a register of length $r$ is

$$\sum_{v=1}^{d} \sum_{\substack{(j_1,\dots,j_v)\in\mathbb{N}^v \mid \ \forall i\\ j_i>0,\ j_1+\cdots+j_v=r-d}} \binom{d-1}{v-1}\frac{1}{v}$$

or, equivalently,

$$\sum_{v=1}^{d} \sum_{\substack{(w_1,\dots,w_v)\in\mathbb{N}^v \mid \\ w_1+\cdots+w_v=r-d-v}} \binom{d-1}{v-1}\frac{1}{v}.$$

At this stage, we can consider the $r$ possible shifts, thereby obtaining the formula stated in the lemma. □

**Lemma 2** *Let $r, d$ and $m$ be positive integers such that $m, d < r$ and consider the following set*

$$H := \{(j_1,\dots,j_v)\in\mathbb{N}^v \mid \ \sum_{i=1}^{v} j_i = r-d,\ 1\le j_i\le m,\ \exists k : j_k = m\}.$$

*Then,*

$$|H| = R(r-d-v, v, m-1) - R(r-d-v, v, m-2).$$

*Proof* First of all, note that the cardinality of $H$ is equal to the cardinality of the set $\tilde{H}$, defined as

$$\tilde{H} := \{(j_1, \dots, j_v) \in \mathbb{N}^v |\ \sum_{i=1}^{v} j_i = r - d - v,\ 0 \leq j_i \leq m-1,\ \exists k : j_k = m-1\}.$$

$\tilde{H}$ represents the ways in which $r-d-v$ balls can be stored into $v$ bins with capacity $m-1$, with at least one bin that contains $m-1$ balls. Its cardinality can be seen as the number of ways in which $r-d-v$ balls can be stored into $v$ bins with capacity $m-1$, minus the number of ways in which $r-d-v$ balls can be stored into $v$ bins with capacity $m-2$. From Theorem 2, it can be computed by the formula

$$R(r-d-v, v, m-1) - R(r-d-v, v, m-2).$$

□

**Lemma 3** *Let $r, d$ and $m$ be positive integers such that $m, d < r$ and consider the following set*

$$T = \Bigg\{(q_1, \dots, q_k) \in \mathbb{N}^k |\ \sum_{i=1}^{k} q_i = r - d,\ q_i > 0\ \forall i,\ \max\{q_i |\ i \in \{1, \dots, k\}\} \geq r - (m+1)\Bigg\}.$$

*Then*

$$|T| = \binom{r-d-1}{k-1} - R(r-d-k, k, r-m-3).$$

*Proof* First, note that the cardinality of $T$ is equal to the cardinality of the following set

$$\bar{T} = \left\{(q_1, \dots, q_k) \in \mathbb{N}^k |\ \sum_{i=1}^{k} q_i = r-d-k,\ \max\{q_i |\ i \in \{1, \dots, k\}\} \geq r - (m+1) - 1\right\}.$$

At this point the set $\bar{T}$ can be written as the difference between two other sets, more formally as

$$\left\{(q_1, \dots, q_k) \in \mathbb{N}^k |\ \sum_{i=1}^{k} q_i = r-d-k\right\} \setminus \left\{(q_1, \dots, q_k) \in \mathbb{N}^k |\ \sum_{i=1}^{k} q_i = r-d-k,\ 0 \leq q_i \leq r-(m+1)-2\ \forall i\right\}.$$

Hence, the searched cardinality can be thought of as the number of ways of distributing $r-d-k$ balls inside $k$ bins minus the number of ways of distributing $r-d-k$ balls inside $k$ bins with capacity $r-m-3$. Consequently, by Theorem 2 together with the stars and bars method, we get that

$$|T| = \binom{r-d-1}{k-1} - R(r-d-k, k, r-m-3).$$

□

To study the cardinality of $K_2$, let us first consider $\mathcal{W}_2$, defined in Equation (6), namely, the superset of $K_2$ where the requirement of coprimality is removed.

**Theorem 3** *Let*

$$\mathcal{W}_2 := \{(x^i\tilde{h}_0,\ x^j\tilde{h}_1) \in \mathcal{R}^2 \mid \deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r,\ |\tilde{h}_0| = |\tilde{h}_1| = d,\ i,j \in \mathbb{Z}_r\}.$$

*Then,*

$$|\mathcal{W}_2| = r^2 \sum_{v=1}^{d} \sum_{m=\lceil \frac{r-d}{v} \rceil}^{r-d-(v-1)} (R(r-d-v,v,m-1) - R(r-d-v,v,m-2)) \binom{d-1}{v-1} \frac{1}{v} \cdot \left( \sum_{k=1}^{d} \left( \binom{r-d-1}{k-1} - R(r-d-k,k,r-m-3) \right) \binom{d-1}{k-1} \frac{1}{k} \right).$$

*Proof* To compute $|\mathcal{W}_2|$, firstly, we focus on the polynomial $x^i\tilde{h}_0$, that can be seen as the distribution of $d$ ones and $r-d$ zeros over a register of length $r$. According to Lemma 1, it can be constructed in

$$r \sum_{v=1}^{d} \sum_{\substack{(j_1,\ldots,j_v)\in\mathbb{N}^v \mid \forall i \\ j_i>0,\ j_1+\cdots+j_v=r-d}} \binom{d-1}{v-1} \frac{1}{v}$$

different ways. Now, for every polynomial $x^i\tilde{h}_0$ we have to count the number of polynomials $x^j\tilde{h}_1$ such that $(x^i\tilde{h}_0, x^j\tilde{h}_1) \in \mathcal{W}_2$.

Let $v$ be the number of runs of ones and zeros of $x^i\tilde{h}_0$ and let $(j_1,\ldots,j_v)$ be the tuple that describes the lengths of the zero runs. We want to count the number of polynomials that can be written as $x^j\tilde{h}_1 \in \mathcal{R}$, with $\deg(\tilde{h}_1) \leq \max\{j_i \mid i \in \{1,\ldots,v\}\}$.
Let $m$ be the $\max\{j_i \mid i \in \{1,\ldots,v\}\}$. The requirement that $\tilde{h}_1$ has to have degree at most $m$, can be translated into the request that there has to be one run of zeros with at least $r-(m+1)$ elements.
Hence, using Lemma 1, with the latter constraint, we get that the number of ways in which $x^j\tilde{h}_1$ can be chosen is

$$r \sum_{k=1}^{d} \sum_{\substack{(q_1,\ldots,q_k)\in\mathbb{N}^k \mid \forall i \\ q_i>0,\ q_1+\cdots+q_k=r-d, \\ \max\{q_i\}\geq r-(m+1)}} \binom{d-1}{k-1} \frac{1}{k},$$

where $k$ describes the number of runs of ones and $(q_1,\ldots,q_k)$ describes the lengths of the runs of zeros.

Consequently,

$$|\mathcal{W}_2| = r \sum_{v=1}^{d} \sum_{\substack{(j_1,\ldots,j_v)\in\mathbb{N}^v \mid \forall i \\ j_i>0,\ j_1+\cdots+j_v=r-d}} \binom{d-1}{v-1} \frac{1}{v} \left( r \sum_{k=1}^{d} \sum_{\substack{(q_1,\ldots,q_k)\in\mathbb{N}^k \mid \forall i \\ q_i>0,\ q_1+\cdots+q_k=r-d, \\ \max\{q_i\}\geq r-(m+1)}} \binom{d-1}{k-1} \frac{1}{k} \right).$$

Although we have derived a formula, this is not yet sufficient. Let us now refine it in order to obtain a more computable version, as in the theorem statement.

Note that, in the summation over the tuples $(j_1, \ldots, j_v)$, the only relevant information to retain is the number of tuples that have the same maximum value $m$ and the value $m$ itself. Observe that $m$ can range between $\lceil \frac{r-d}{v} \rceil$ and $r-d-(v-1)$.

Due to Lemma 2, the cardinality of $\mathcal{W}_2$ can be rewritten as

$$|\mathcal{W}_2| = r \sum_{v=1}^{d} \sum_{m=\lceil \frac{r-d}{v} \rceil}^{r-d-(v-1)} (R(r-d-v, v, m-1) - R(r-d-v, v, m-2)) \binom{d-1}{v-1} \frac{1}{v} \cdot r \sum_{k=1}^{d} \sum_{\substack{(q_1,\ldots,q_k) \in \mathbb{N}^k \mid \forall i \\ q_i > 0,\ q_1+\cdots+q_k = r-d, \\ \max\{q_i\} \geq r-(m+1)}} \binom{d-1}{k-1} \frac{1}{k}.$$

Due to Lemma 3, we can enhance the latter equation in the following way

$$|\mathcal{W}_2| = r^2 \sum_{v=1}^{d} \sum_{m=\lceil \frac{r-d}{v} \rceil}^{r-d-(v-1)} (R(r-d-v, v, m-1) - R(r-d-v, v, m-2)) \binom{d-1}{v-1} \frac{1}{v} \cdot \left( \sum_{k=1}^{d} \left( \binom{r-d-1}{k-1} - R(r-d-k, k, r-m-3) \right) \binom{d-1}{k-1} \frac{1}{k} \right).$$

□

*Remark 2* Equation (19) in [12], for $w_1 = w_2 = d$, describes the probability of selecting a pair of polynomials within $\mathcal{W}_2$, i.e., it represents the quantity $\frac{|\mathcal{W}_2|}{|\mathcal{P}_d|}$. The two equations appear different because they were derived using different methodologies.

### 4.1.2 Cardinality of the set $K_2$

For case 2, determining the cardinality of the weak key set means estimating the size of $K_2$. This estimation is carried out through a statistical analysis aimed at determining the portion of elements in $\mathcal{W}_2$ that also belong to $K_2$.

We want to know the probability that a pair of polynomials in $\mathcal{W}_2$ is also inside $K_2$, that is, the probability that it can be written as $(x^i \tilde{h}_0, x^j \tilde{h}_1)$ with $\tilde{h}_0$ and $\tilde{h}_1$ coprime in $\mathbb{F}_2[x]$.

Note that, since the pairs of polynomials in $\mathcal{W}_2$ live in the ring $\mathcal{R}$, there is not a unique representation of them. However, we can state that a pair is weak if it admits a representation of the just described type.

The main challenge of this statistical analysis is to determine how to randomly select a pair of polynomials from $\mathcal{W}_2$. Note that, a straightforward approach would be to randomly generate a pair of polynomials in $\mathcal{P}_d$ and retain it only if it belongs

to $\mathcal{W}_2$. However, this method could not possibly work, because the probability of a randomly generated pair falling within $\mathcal{W}_2$ becomes too small for large values of $r$, making it unfeasible to generate enough pairs to conduct a meaningful analysis.
On the other hand, generating all elements inside $\mathcal{W}_2$ and then selecting one at random is infeasible, as the set is too large and the memory requirements to carry out the random generation in this way would be prohibitive.
In light of these thoughts, we will proceed by designing a method to construct the elements of $\mathcal{W}_2$ as if they were randomly sampled.

**Definition 3** Let $\sim_{r_2}$ be an equivalence relation on $\mathcal{W}_2$, such that two pairs of polynomials in $\mathcal{W}_2$ are equivalent if one can be obtained from the other by applying shifts to both the first and second polynomial, i.e.

$$(h_0, h_1) \sim_{r_2} (h_0^{'}, h_1^{'}) \iff \exists\ i, j \in \mathbb{Z}_r \ s.t.\ (x^i h_0, x^j h_1) = (h_0^{'}, h_1^{'}).$$

Let $\tilde{\mathcal{W}}_2$ be a set of representatives for these equivalence classes such that $h_0(0) = h_1(0) = 1$. Note that a set of representatives of this type exists.

Hence, by conducting a statistical test to approximate the probability that a pair of polynomials in $\tilde{\mathcal{W}}_2$ is coprime in $\mathbb{F}_2[x]$, we can establish a lower bound on the probability that a pair of polynomials in $\mathcal{W}_2$ is a weak pair.

First, we need to discuss how to select elements from $\tilde{\mathcal{W}}_2$ according to the uniform distribution. This requires studying certain properties of polynomials uniformly sampled from $\tilde{\mathcal{W}}_2$. We remark that, indeed, not all features within $\tilde{\mathcal{W}}_2$ are distributed uniformly: for example, the length of the longest sequence of zeros in one polynomial of a uniformly picked pair may appear more frequently within certain ranges than in others. Our idea is to leverage these properties in order to define an efficient procedure that outputs a polynomial uniformly selected from $\tilde{\mathcal{W}}_2$.

*Procedure 1* Let us introduce some distributions that are given explicitly in Theorem 4. Let $\mathcal{D}_1$ represent the probability distribution of the length of the longest zero run in $h_0$, and $\mathcal{D}_2^m$ the probability distribution to have exactly $n$ longest runs of zeros of length $m$, for a fixed $m$. $\mathcal{D}_3^m$ represents the probability distribution of the length of the longest zero run in $h_1$, once the longest zero run of $h_0$ is set to $m$.

1. **Picking the longest zero run in $\mathbf{h_0}$**: we select a specific value $m$ based on $\mathcal{D}_1$;
2. **Selecting the number of runs of length m**: we determine a certain value $n$ of runs of length $m$ within $h_0$, by relying on the probability distribution $\mathcal{D}_2^m$;
3. **Selecting the remaining zero runs**: now, we need to determine the lengths of the remaining zero runs in $h_0$.
   - We randomly pick a subset of size $d - n - 1$ from the set $\{2, \ldots, r - n(m+1)\}$. Think of this subset as markers separating the zero runs;

- Each marker must be at most $m$ positions away from the previous and next one. The first separator needs to be at most $m$ positions away from the position 1 and the last separator needs to be at most $m$ positions away from the position $r - n(m+1) + 1$. If any of these checks fail, we simply discard the randomly generated subset and try again.

In this way we create $d - n$ zero runs with lengths between 0 and $m - 1$ (note that a run of length 0 is not a real run);

4. **Creating the polynomial $\mathbf{h_0}$**: we start by randomly distributing $n - 1$ longest zero runs among the $d - n$ zero runs identified in the previous step (one longest run is kept aside, to maintain it at the end). To do so, we follow these steps:
   - we generate a sequence $L$ of length $d - 1$;
   - we randomly select $n - 1$ positions from $\{1, \ldots, d-1\}$ and set these positions in the sequence to the value $m$. The remaining entries are filled with the lengths of the other runs;

   Finally, we create $h_0$: a polynomial of degree $r - m - 1$ that has constant term equal to 1, such that its zero runs precisely match the entries of the sequence $L$;
5. **Picking the longest run in $\mathbf{h_1}$**: we pick a value $k \in \{r - (m+1), \ldots, r - d\}$ representing the longest zero run in $h_1$, by using $\mathcal{D}_3^m$;
6. **Counting the runs of length $\mathbf{k}$**: we select the number $s$ of runs of length $k$ within $h_1$, using $\mathcal{D}_2^k$;
7. **Creating the polynomial $\mathbf{h_1}$**: finally, we create the polynomial $h_1$ using the same approach utilized for $h_0$, but with $k$ and $s$ instead of $m$ and $n$, respectively.

In preparation for the proof of correctness, we present some preliminary lemmas.

**Lemma 4** *Let $\tilde{\mathcal{W}}_2$ be a set of representative described by Definition 3, then*

$$|\tilde{\mathcal{W}}_2| = \sum_{v=1}^{d} \sum_{m=\lceil \frac{r-d}{v} \rceil}^{r-d-(v-1)} (R(r-d-v, v, m-1) - R(r-d-v, v, m-2)) \binom{d-1}{v-1} \frac{1}{v} \cdot \left( \sum_{k=1}^{d} \left( \binom{r-d-1}{k-1} - R(r-d-k, k, r-m-3) \right) \binom{d-1}{k-1} \frac{1}{k} \right).$$

*Proof* Directly derived from Theorem 3. □

**Lemma 5** *For fixed $m$ and $n$, the number of pairs of polynomials $(h_0, h_1)$ in $\tilde{\mathcal{W}}_2$ such that $h_0$ has exactly $n$ longest zero runs of length $m$ is the product of the quantity*

$$R(r-d-mn, d-n, m-1) \binom{d}{n} \frac{1}{d}$$

*by the number of possible polynomials $h_1$.*

*Proof* Once that $n$ is fixed, we are left with at most $d-n$ zero runs. For simplicity, we say that we have exactly $d-n$ runs left, each taking values between 0 and $m-1$, and collectively summing up to $r-d-mn$. Note that, in this case, a run of length 0 is not a real run. The manner in which these remaining runs can be selected mirrors the process of distributing $r-d-mn$ balls into $d-n$ bins, each with a capacity of $m-1$. At this point, the result must be multiplied by the number of ways in which the $n$ runs of maximum length can be positioned among the other runs. This is given by the binomial coefficient $\binom{d}{n}$. The final detail to address is to avoid circular repetitions. Consequently, the result has to be divided by $d$. Once $m$ is fixed, the number of possible polynomials $h_1$ is well defined and constant. □

At this point we can state the following theorem:

**Theorem 4** *The output of Procedure 1 is a pair of polynomials uniformly drawn from* $\tilde{\mathcal{W}}_2$.

*Proof* The two polynomials $h_0$ and $h_1$ will be represented by two registers of length $r$ with the first position set to one and the longest sequence of zeros placed at the end.

Let us consider $h_0$. The probability distribution $\mathcal{D}_1$ of its longest sequence of zeros having a specific length $m$ within the range $[\lceil \frac{r-d}{d} \rceil, \ldots, r-d]$ is not uniform. To characterize $\mathcal{D}_1$, we can tally, for each value of $m$, the number of pairs in $\tilde{\mathcal{W}}_2$ where $h_0$ has a longest zero run of length $m$, and, then, normalize it. To do this, we can rely on Lemma 4 that describes the cardinality of the set, where the variable $m$ represents the longest sequence of zeros in the first polynomial.

At this point, for every possible value of $m$, we also need to know the probability for $h_0$ to have exactly $n$ runs of zeros of length $m$. This will help us in assigning equal probability to each equivalence class of polynomials. These probability distributions, that depend on the fixed value of $m$ are called $\mathcal{D}_2^m$. Let us fix $m$. To find $\mathcal{D}_2^m$ we have to count, for every value of $n$ in $\{1, \ldots, d\}$, the pairs of polynomials $(h_0, h_1)$ in $\tilde{\mathcal{W}}_2$ such that $h_0$ has exactly $n$ longest zero runs of length $m$. For a fixed $m$, once the number of pairs in $\tilde{\mathcal{W}}_2$ with $n$ runs of maximum length $m$ in the first polynomial has been determined, following Lemma 5, we simply normalize these values. This yields the desired probability distributions $\mathcal{D}_2^m$.

Now, let us turn our attention to the second polynomial of the pair: $h_1$. Once $h_0$ is fixed, the options for $h_1$ become more restricted. To generate the polynomial $h_1$ such that the pair $(h_0, h_1)$ appears as if it were randomly drawn from $\tilde{\mathcal{W}}_2$, additional probability distributions must be determined.

First, for each possible longest zero run length (denoted by $m$) in $h_0$, we need to characterize the probability distribution for the longest zero run in $h_1$. These probability distributions will be denoted with $\mathcal{D}_3^m$. The value of the longest zero run in $h_1$, that we call $k$, can range from $r-(m+1)$ all the way up to $r-d$. To determine the number of polynomials $h_1$ with a specific $k$ value, we can follow the same logic used for calculating the cardinality of $\mathcal{W}_2$ in Theorem 3. Essentially, for each possible $k$ between $r-(m+1)$ and $r-d$, the number of polynomials $h_1$ with a longest zero run of length $k$ is given by

$$\sum_{v=1}^{d} (R(r-d-v, v, k-1) - R(r-d-v, v, k-2)) \binom{d-1}{v-1} \frac{1}{v}.$$

Once we have counted, for a fixed $m$, the number of polynomials $h_1$ for each possible $k$, we simply have to normalize these values in order to obtain $\mathcal{D}_3^m$. For a fixed $k$ value, we

might still expect to determine a separate probability distribution for the number of runs of maximum length $k$ within $h_1$. However, this distribution is actually identical to the one previously calculated for the number of runs of maximum length $m$ in $h_0$.

□

## 4.2 Weak keys for Attack 1

### 4.2.1 Cardinality of the set $\mathcal{W}_1$

Let us now focus on $\mathcal{W}_1$, described by Equation (5), that denotes the set of polynomial pairs corresponding to case 1, where the coprimality requirement is relaxed.

**Theorem 5** *Let*

$$\mathcal{W}_1 := \{(x^i\tilde{h}_0,\ x^i\tilde{h}_1) \in \mathcal{R}^2 \mid \deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r,\ |\tilde{h}_0| = |\tilde{h}_1| = d,\ i \in \mathbb{Z}_r\}.$$

*Then,*

1. *the cardinality of $\mathcal{W}_1$ is bounded from below by*

$$r\sum_{v=1}^{d}\sum_{m=\lceil\frac{r-d}{v}\rceil}^{r-d-(v-1)}(R(r-d-v,v,m-1)-R(r-d-v,v,m-2))\binom{d-1}{v-1}\frac{1}{v}\binom{m+1}{d};$$

2. *the cardinality of $\mathcal{W}_1$ is bounded from above by*

$$r\sum_{s=2d-2}^{r-1}\sum_{d_1=d-1}^{s-(d-1)}\left[\binom{d_1-1}{d-2}\binom{s-d_1}{d-1}+\binom{d_1-1}{d-1}\binom{s-d_1-1}{d-2}\right].$$

*Proof* 1. Let us start with the lower bound. First, we focus on the polynomial $x^i\tilde{h}_0$. Due to Lemma 1 it can be constructed in

$$r\sum_{v=1}^{d}\sum_{\substack{(j_1,\ldots,j_v)\in\mathbb{N}^v \mid\ \forall i\\ j_i>0,\ j_1+\cdots+j_v=r-d}}\binom{d-1}{v-1}\frac{1}{v}$$

ways, where $v$ represents the number of runs of ones and the tuple $(j_1,\ldots,j_v)$ represents the lengths of the runs of zeros.

Let us now focus on the polynomial $x^i\tilde{h}_1$. Notice that it must be chosen such that $\deg(\tilde{h}_0)+\deg(\tilde{h}_1) < r$, and consequently, the degree of $\tilde{h}_1$ must be at most $r-\deg(\tilde{h}_0)-1$. This is equivalent to requiring that all the ones are within $r-\deg(\tilde{h}_0)$ consecutive positions of the circular register. The degree of the polynomial $\tilde{h}_0$ is determined by the tuple $(j_1,\ldots,j_v)$. As a consequence, the degree of $\tilde{h}_1$ has to be at most $j_k$ as $k$ varies, depending on the position on the register where the fist coefficient of $\tilde{h}_1$ is placed (or equivalently depending on the shift

$x^i$). Let $m = \max\{j_i \mid \ i \in \{1, \ldots, v\}\}$.

For every polynomial $x^i\tilde{h}_0$ a lower bound to the number of polynomials $x^i\tilde{h}_1$ such that $(x^i\tilde{h}_0, x^i\tilde{h}_1) \in \mathcal{W}_1$ is

$$\binom{m+1}{d}.$$

The latter represents the number of polynomials of weight $d$ and degree at most $m$. Consequently, we can state that

$$|\mathcal{W}_1| \geq r \sum_{v=1}^{d} \sum_{\substack{(j_1,\ldots,j_v)\in\mathbb{N}^v \mid \ \forall i \\ j_i>0,\ j_1+\cdots+j_v=r-d}} \binom{d-1}{v-1} \frac{1}{v} \binom{m+1}{d}.$$

As in Section 4.1.1, we can reformulate this equation into the more easily computable form

$$|\mathcal{W}_1| \geq r \sum_{v=1}^{d} \sum_{m=\lceil \frac{r-d}{v} \rceil}^{r-d-(v-1)} (R(r-d-v, v, m-1) - R(r-d-v, v, m-2)) \binom{d-1}{v-1} \frac{1}{v} \binom{m+1}{d},$$

which is the claimed lower bound.

2. Let us turn our attention to the upper bound. $\mathcal{W}_1$ can be written as

$$\mathcal{W}_1 = \bigcup_{s=2d-2}^{r-1} \mathcal{W}_1^s,$$

where

$$\mathcal{W}_1^s := \{(x^i\tilde{h}_0, x^i\tilde{h}_1) \in \mathcal{R}^2 \mid \ \deg(\tilde{h}_0) + \deg(\tilde{h}_1) = s, \ |\tilde{h}_0| = |\tilde{h}_1| = d, \ \ \tilde{h}_0(0) = 1 \vee \tilde{h}_1(0) = 1, \ i \in \mathbb{Z}_r\}.$$

Note that the condition $\tilde{h}_0(0) = 1 \vee \tilde{h}_1(0) = 1$ is not restrictive, since every pair in $\mathcal{W}_1$ can be written in the form $(x^i\tilde{h}_0, x^i\tilde{h}_1) \in \mathcal{R}^2$ where $\deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r$ and at least one of the polynomials $\tilde{h}_0, \tilde{h}_1$ has constant term equal to 1.

Since the sets $\mathcal{W}_1^s$ are not disjoint,

$$|\mathcal{W}_1| \leq \sum_{s=2d-2}^{r-1} |\mathcal{W}_1^s|.$$

In turn, each set $\mathcal{W}_1^s$ can be seen as the union of the sets $\mathcal{W}_1^{s,i}$, with $i \in \mathbb{Z}_r$, defined as

$$\mathcal{W}_1^{s,i} := \{(x^i\tilde{h}_0, x^i\tilde{h}_1) \in \mathcal{R}^2 \mid \deg(\tilde{h}_0) + \deg(\tilde{h}_1) = s, \\ |\tilde{h}_0| = |\tilde{h}_1| = d,\ \tilde{h}_0(0) = 1 \vee \tilde{h}_1(0) = 1\}.$$

Note that $\mathcal{W}_1^{s,i}$ represents the subset of $\mathcal{W}_1^s$ where the shift $i$ is fixed. Consequently,

$$\mathcal{W}_1 = \bigcup_{s=2d-2}^{r-1} \mathcal{W}_1^s = \bigcup_{s=2d-2}^{r-1} \bigcup_{i=0}^{r-1} \mathcal{W}_1^{s,i}.$$

Since $\forall i_1, i_2 \in \mathbb{Z}_r$ it holds that $\left|\mathcal{W}_1^{s,i_1}\right| = \left|\mathcal{W}_1^{s,i_2}\right|$, then

$$|\mathcal{W}_1| \leq \sum_{s=2d-2}^{r-1} |\mathcal{W}_1^s| \leq \sum_{s=2d-2}^{r-1} \sum_{i=0}^{r-1} \left|\mathcal{W}_1^{s,i}\right| \leq r \sum_{s=2d-2}^{r-1} \left|\mathcal{W}_1^{s,0}\right|.$$

To determine the cardinality of $\mathcal{W}_1^{s,0}$, we begin by noting that the degree of the first polynomial can range from $d-1$ to $s-(d-1)$. Thus, we sum over a variable $d_1$ spanning over these values.

Considering that at least one of the two polynomials must have a constant term set to 1, we identify two cases:

(a) The first polynomial has a constant term equal to 1;
(b) The first polynomial has a constant term equal to 0, leading to the second polynomial having a constant term set to 1.

Consequently, let $d_1$ represent the degree of the first polynomial,

$$\left|\mathcal{W}_1^{s,0}\right| = \sum_{d_1=d-1}^{s-(d-1)} \left[\binom{d_1-1}{d-2}\binom{s-d_1}{d-1} + \binom{d_1-1}{d-1}\binom{s-d_1-1}{d-2}\right].$$

In conclusion, we can state that

$$|\mathcal{W}_1| \leq r \sum_{s=2d-2}^{r-1} \sum_{d_1=d-1}^{s-(d-1)} \left[\binom{d_1-1}{d-2}\binom{s-d_1}{d-1} + \binom{d_1-1}{d-1}\binom{s-d_1-1}{d-2}\right].$$

□

*Remark 3* The set $\mathcal{W}_1$ was not identified in [12]. However, if we consider Equation (12) in [12] and we set $w_1 = w_2 = d$ we obtain the cardinality of a subset of $\mathcal{W}_1$, namely the set $W_1$ (Equation 2), with $w_0 = w_1 = d$.

### 4.2.2 Cardinality of the set $K_1$

For case 1, to determine the number of weak keys, we need to estimate the cardinality of $K_1$. Our strategy is to determine the proportion of elements in $\mathcal{W}_1$ that also belong to $K_1$. Hence, we want to know the probability that a pair of polynomials inside $\mathcal{W}_1$ is also inside $K_1$.

In order to conduct this analysis we have to randomly sample some pairs in $\mathcal{W}_1$ and check if there exists a representation of the type $(x^i\tilde{h}_0, x^i\tilde{h}_1)$ such that $\deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r$ and $\gcd(\tilde{h}_0, \tilde{h}_1) = 1$ in $\mathbb{F}_2[x]$.
However, in this scenario, generating random polynomial pairs from $\mathcal{W}_1$ is even more challenging than the previous case, because we lack a precise formula for the cardinality of $\mathcal{W}_1$.
We have therefore decided to start from the upper and lower bounds established in Section 4.2.1. Since these bounds represent, respectively, the cardinality of a subset and of a multiset[1] of elements within $\mathcal{W}_1$, we conduct our statistical analysis on these sets instead of on $\mathcal{W}_1$ itself.

Let us start from the lower bound and consider the following subset of $\mathcal{W}_1$

$$\mathcal{W}_1^{\text{low}} := \{(x^i h_0, x^i h_1) \in \mathcal{R}^2 \mid |h_0| = |h_1| = d,\ r - \deg(h_0) > m, \\ \deg(h_0) + \deg(h_1) < r,\ i \in \mathbb{Z}_r\}, \quad (8)$$

where $m$ denotes the length of the longest zero run in $h_0$.

**Lemma 6** *Let $\mathcal{W}_1^{\text{low}}$ be the set defined in Equation (8), then*

$$\left|\mathcal{W}_1^{\text{low}}\right| = r \sum_{v=1}^{d} \sum_{m=\lceil \frac{r-d}{v} \rceil}^{r-d-(v-1)} (R(r-d-v, v, m-1) - \\ R(r-d-v, v, m-2)) \binom{d-1}{v-1} \frac{1}{v} \binom{m+1}{d}.$$

*Proof* Directly derived from Theorem 5 point 1.
Note that, in order to count the elements $(x^i h_0, x^i h_1)$ in $\mathcal{W}_1^{\text{low}}$, we may construct them by following the first point of the proof of Theorem 5. In particular, the construction of the first polynomial in the pair relies on Lemma 1. It is convenient to consider the representation $(x^i h_0, x^i h_1)$ with $h_0(0) = 1$, since this allows to follow step by step the proof of Lemma 1. □

*Remark 4* Note that $|\mathcal{W}_1^{\text{low}}|$ is exactly the lower bound of $|\mathcal{W}_1|$ stated in Theorem 5.

[1] A multiset is a generalization of a set, where repetition of elements is allowed.

**Definition 4** Let $\sim_{r_1}$ be an equivalence relation on $\mathcal{W}_1^{\text{low}}$, such that two pairs of polynomials are equivalent if one can be obtained from the other by applying an equal shift, i.e.

$$(h_0, h_1) \sim_{r_1} (h_0^{'}, h_1^{'}) \iff \exists\ i \in \mathbb{Z}_r \ s.t.\ (x^i h_0, x^i h_1) = (h_0^{'}, h_1^{'}).$$

Let $\tilde{\mathcal{W}}_1^{\text{low}}$ be a set of representatives for these equivalence classes, such that each representative is in the form $(h_0, h_1)$, with $h_0(0) = 1$.

**Corollary 1** *Let $\tilde{\mathcal{W}}_1^{\text{low}}$ be a set of representative described by Definition 4, then*

$$\left|\tilde{\mathcal{W}}_1^{\text{low}}\right| = \sum_{v=1}^{d} \sum_{m=\lceil \frac{r-d}{v} \rceil}^{r-d-(v-1)} (R(r-d-v, v, m-1) - R(r-d-v, v, m-2)) \binom{d-1}{v-1} \frac{1}{v} \binom{m+1}{d}.$$

*Proof* Note that, the cardinality of $\mathcal{W}_1^{\text{low}}$ in Lemma 6 is expressed as the product of $r$ by a second part. $\mathcal{W}_1^{\text{low}}$ consists of the disjoint union of a set of representative $\tilde{\mathcal{W}}_1^{\text{low}}$ with the set of all pairs of polynomials in $\tilde{\mathcal{W}}_1^{\text{low}}$ subjected to an equal shift. From this we get the statement of the corollary. □

We want to conduct an analysis on the probability that a pair of polynomials in $\mathcal{W}_1^{\text{low}}$ can be represented as $(x^i \tilde{h}_0, x^i \tilde{h}_1)$, with $\deg(\tilde{h}_0) + \deg(\tilde{h}_1) < r$ and $\gcd(\tilde{h}_0, \tilde{h}_1) = 1$ in $\mathbb{F}_2[x]$.

We will conduct our analysis on the set $\tilde{\mathcal{W}}_1^{\text{low}}$ defined in Corollary 1. However, since the polynomial pairs we are working with reside in the quotient ring $\mathcal{R}$, they may have multiple representations of the form $(x^i \tilde{h}_0, x^i \tilde{h}_1)$. Therefore, when analyzing $\tilde{\mathcal{W}}_1^{\text{low}}$, we account for only one representation of each pair. This approach allows us to derive a lower bound for the probability that a pair in $\mathcal{W}_1^{\text{low}}$ is weak.
Let us define a procedure that mimics a random draw from $\tilde{\mathcal{W}}_1^{\text{low}}$.

*Procedure 2* Let us introduce some distributions that are given explicitly in Theorem 6. Let $\mathcal{D}_4$ represent the probability distribution of the longest zero run in the polynomial $h_0$ and $\mathcal{D}_2^m$ represent the probability distribution for $h_0$ to have exactly $n$ longest zero runs of length $m$, for a fixed $m$.
Because the first four steps of this procedure are identical to those in Section 4.1.2, aside from the use of different distributions, we present only a summary here.

1. **Selecting the length of the longest zero run in $\mathbf{h_0}$**: we select a value $m$ accordingly to $\mathcal{D}_4$;
2. **Determining the number of zero runs of length m inside $\mathbf{h_0}$**: we select the value $n$ of runs of length $m$ based on $\mathcal{D}_2^m$;
3. **Selecting the remaining zero runs**:

- we randomly pick a subset of size $d-n-1$ from the set $\{2,\dots,r-n(m+1)\}$, that represents the markers separating the zero runs;
- we check that each element in the subset is at most $m$ positions away from the previous one, from the next one, from the first position, and from the position $r-n(m+1)+1$. If a check fails, we try again.

4. **Creating the polynomial $\mathbf{h_0}$**: firstly we randomly distribute $n-1$ longest zero runs among the $d-n$ zero runs identified in the previous step keeping one longest run at the end. Then we create $h_0$: a polynomial with degree $r-m-1$ whose runs precisely match with the ones distributed before;
5. **Creating the polynomial $\mathbf{h_1}$**: to create this second polynomial we just need to generate a length-$r$ sequence of zeros and randomly select $d$ positions among the first $m+1$, whose values are set to 1. This resulting sequence represents the coefficients of $h_1$.

**Theorem 6** *The output of Procedure 2 is a pair of polynomials uniformly drawn from* $\tilde{\mathcal{W}}_1^{\text{low}}$.

*Proof* Firstly, we focus on the polynomial $h_0$, which can be represented by a length-$r$ register with the first position set to 1 and the longest zero run placed at the end.
The probability distributions that we will use to generate a pair $(h_0,h_1)$, as if it were randomly drawn from $\tilde{\mathcal{W}}_1^{\text{low}}$, can be computed in the same way as in Section 4.1.2.
More specifically, the cardinality formula for the set $\tilde{\mathcal{W}}_1^{\text{low}}$, stated in Corollary 1, enables efficient enumeration of elements where the first polynomial has a longest zero run of length $m$, with $m$ ranging from $\lceil \frac{r-d}{d} \rceil$ to $r-d$. This enumeration, followed by normalization, computes the probability distribution of the longest zero run in the polynomial $h_0$, that we will call $\mathcal{D}_4$.

Next, for each possible value of $m$, we also need to determine the probability distribution for $h_0$ having exactly $n$ zero runs of length $m$. These distributions, which differ depending on the fixed value of $m$, are exactly $\mathcal{D}_2^m$, presented in Procedure 1.

At this point we can focus on $h_1$, which must satisfy the condition of having degree at most $m$. □

Regarding the upper bound of Theorem 5, the situation is different. Unlike the cases just discussed, the bound we found is not only an upper bound for $|\mathcal{W}_1|$, but also for $|K_1|$, since $|K_1| \leq |\mathcal{W}_1|$.
Consequently, we can define a set that contains $K_1$ and has cardinality matching the upper bound, and thus conduct a statistical analysis to determine a lower bound on the probability that the elements of this superset are coprime. We remark that this approach does not hold the same relevance as the analyses conducted so far. However, it can still provide insight into the impact of the coprimality requirement on the count of weak pairs. For this reason, we have decided to perform this statistical analysis as well. Let us define the following set

$$\tilde{\mathcal{W}}_1^{\text{up}} := \{(h_0,h_1) \in \mathcal{R}^2 \mid |h_0| = |h_1| = d,\ h_0(0) = 1 \vee h_1(0) = 1, \\ \deg(h_0) + \deg(h_1) < r\}. \quad (9)$$

Let $\mathcal{W}_1^{\text{up}}$ denote the multiset formed by taking the just defined set and by adding all elements in $\tilde{\mathcal{W}}_1^{\text{up}}$ to which an equal shift to both polynomials is applied. Note that $\mathcal{W}_1^{\text{up}}$ is a multiset, and not a simple set, since it contains some pairs more than once. What is essential, however, is that $\mathcal{W}_1^{\text{up}}$ contains all elements of $\mathcal{W}_1$.
The probability that a pair in $\mathcal{W}_1^{\text{up}}$ is weak is not equal to the probability that a pair in $\tilde{\mathcal{W}}_1^{\text{up}}$ is coprime. We can however say that the second probability is a lower bound for the first one.

**Lemma 7** *Let $\tilde{\mathcal{W}}_1^{\text{up}}$ be the set defined in Equation (9). Then,*

$$\left|\tilde{\mathcal{W}}_1^{\text{up}}\right| = \sum_{s=2d-2}^{r-1} \sum_{d_1=d-1}^{s-(d-1)} \left[ \binom{d_1-1}{d-2}\binom{s-d_1}{d-1} + \binom{d_1-1}{d-1}\binom{s-d_1-1}{d-2} \right],$$

*where the variable $s$ represents the sum of the degrees of the two polynomials, and the variable $d_1$ represents the degree of the first polynomial.*

*Proof* Directly derived from Theorem 5 point 2. □

We will now focus on the set $\tilde{\mathcal{W}}_1^{\text{up}}$. We want to conduct a statistical analysis to estimate the probability that a pair of polynomials inside this set is coprime in $\mathbb{F}_2[x]$. To achieve this, we need a method for randomly selecting elements from the set.
Also in this case, in order to efficiently construct elements of $\tilde{\mathcal{W}}_1^{\text{up}}$ as if they were randomly picked, we need to focus on some properties of the two polynomials and find their probability distributions.

The procedure we will use for constructing elements of $\tilde{\mathcal{W}}_1^{\text{up}}$ as if they were randomly selected is the following.

*Procedure 3* Let $\mathcal{D}_5$ represent the probability distribution of the degree of $h_0$, given in Theorem 7.

1. **Selecting the degree of $\mathbf{h_0}$**: we pick a value $d_1$, for the degree of $h_0$, following the probability distribution $\mathcal{D}_5$;
2. **Generating the polynomial $\mathbf{h_0}$**: we randomly generate a polynomial of degree $d_1$ and of weight $d$. To do this we build a sequence of $d_1 + 1$ zeros, that represents the coefficients of the polynomial. Then, we set the coefficient in position $d_1+1$ to 1, in this way the resulting polynomial will be of degree $d_1$. We then select randomly other $d-1$ positions among the remaining ones and we set these coefficients to 1;
3. **Generating the polynomial $\mathbf{h_1}$**: we randomly generate a polynomial of degree at most $r-1-d_1$ and weight $d$, following a similar method used in the previous point. We build a sequence of $r-d_1$ zeros that represents the coefficients and we randomly select $d$ positions that are turned to 1;

4. **Checking if the condition $\mathbf{h_0(0) = 1 \vee h_1(0) = 1}$ is satisfied**: we check if $h_0(0) = 1 \vee h_1(0) = 1$, otherwise we discard $h_0$ and $h_1$ and restart from the second step.

**Theorem 7** *The output of Procedure 3 is a pair of polynomials uniformly drawn from* $\tilde{\mathcal{W}}_1^{\text{up}}$.

*Proof* Let us start by focusing on the first polynomial of the pair. Analyzing the formula that describes $|\tilde{\mathcal{W}}_1^{\text{up}}|$, given in Lemma 7, the probability distribution $\mathcal{D}_5$ of the degree of $h_0$ can be easily found.
At this point the second polynomial must be of degree at most $r - 1 - \deg(h_0)$ and at least one among $h_0$ and $h_1$ must have constant term equal to 1. □

# 5 Key-recovery attack for BIKE

## 5.1 Experimental setup

Now that we have constructed a method to efficiently generate random polynomial pairs in $\tilde{\mathcal{W}}_2$, $\tilde{\mathcal{W}}_1^{\text{low}}$ and $\tilde{\mathcal{W}}_1^{\text{up}}$, we can proceed with the statistical analysis. To do this we need to fix the parameters $r$ and $d$.

Since Problem 1 is closely related to the security of certain post-quantum code-based schemes, solving it may, in some cases, lead to breaking the corresponding cryptosystems. To highlight this connection and carry out a statistical analysis with real-world parameters we will apply our approach to a known scheme.

BIKE [4] is a Key-Encapsulation Mechanism (KEM) based on the Niederreiter framework [15] that uses QC-MDPC codes. We briefly present here the parameters of BIKE, the key generation algorithm, and, thus, the link between the public and private keys. For a complete description of BIKE we refer to [4].
BIKE uses QC-MDPC codes of length $n = 2r$, dimension $r$, and designed minimum distance $w$. In particular,

- $r$ is a positive prime integer with the property that 2 is primitive modulo $r$, that determines the structure of the polynomial ring $\mathcal{R}$;
- $w \approx \sqrt{n}$ is an even positive integer used to define the row weight of the QC-MD parity-check matrix of the code, and it has to be such that $d := w/2$ is odd.
  This parameter determines the private key space $\mathcal{H}_w := \{(h_0, h_1) \in \mathcal{R}^2 \mid |h_0| = |h_1| = d\}$.

Other than $r$ and $w$, the parameters in BIKE include $t$ and $l$, which however do not have an impact on the key generation and thus on our analysis. The parameters suggested in BIKE to achieve the three NIST security categories are shown in Table 1.

For the scope of this work, we will focus only on the main points of the key generation algorithm.

| Security level | Parameters | | | |
|---|---|---|---|---|
| | $r$ | $w$ | $t$ | $l$ |
| I | 12 323 | 142 | 134 | 256 |
| III | 24 659 | 206 | 199 | 256 |
| V | 40 973 | 274 | 264 | 256 |

**Table 1**: Suggested BIKE parameters

The private key consists of a pair of polynomials $(h_0, h_1)$, randomly selected from the private key space $\mathcal{H}_w$. Hence, $h_0, h_1 \in \mathcal{R}$, each with weight exactly $d$, and they represent a QC-MDPC matrix. The public key $h$, computed as $h := h_1 h_0^{-1} \in \mathcal{R}$ represents an equivalent QC parity-check matrix.

Note that recovering the two polynomials that constitute the private key, starting from the public key, means solving Problem 1.

Consequently, the presented approaches constitute some key-recovery attacks for BIKE, and the sets $K_1$ and $K_2$ represent the weak keys retrievable with these two attacks. From now on, we will refer to the strategy where we attempt to find the private key by running the MEEA on $(x^r - 1, h, d)$ as Attack 1, while Attack 2 consists of running the MEEA on each shift of $h$. In order to estimate the cost of these two attacks we need to know the probability of selecting a weak key. This probability is the ratio of the cardinality of the sets $K_1$ (or $K_2$), and the cardinality of the weak key space $\mathcal{H}_w$ (i.e. $\mathcal{P}_d$). In order to estimate $|K_1|$ and $|K_2|$, we proceed first by computing $|\mathcal{W}_1|$ and $|\mathcal{W}_2|$. Then, using MAGMA [16] to generate the polynomials according to Section 4.1.2 and Section 4.2.2, we perform the statistical analysis.

## 5.2 Experimental results

We estimate the cardinality of $K_1$ and $K_2$ for the parameters BIKE suggested for the security levels I and III.

To reliably estimate the probability that a pair inside $\tilde{\mathcal{W}}_2$, $\tilde{\mathcal{W}}_1^{low}$ or $\tilde{\mathcal{W}}_1^{\text{up}}$ is co-prime we will calculate the corresponding confidence intervals (CI), that is a range of values that is likely to contain the true value of an unknown population parameter.

Let us now consider a random sample $(X_1, X_2, \ldots, X_n)$ drawn from a population following a Bernoulli distribution with parameter $p$. In this context, each $X_i$ is a binary random variable that takes the value 1 with probability $p$, and 0 with probability $1-p$. Our goal is to determine the confidence interval for the parameter $p$, denoted by $\text{CI}_p(1-\alpha)$. This confidence interval will give us a range of plausible values for $p$ with a specified confidence level of $1-\alpha$ (or equivalently with a significance level of $\alpha$).

**Definition 5** Let $(X_1, X_2, \ldots, X_n)$ be a random sample from a given distribution. The *Sample Mean Estimator*, denoted by $\bar{X}_n$, is defined as

$$\bar{X}_n := \frac{1}{n} \sum_{i=1}^{n} X_i.$$

In our case $\bar{X}_n$ represents the average value of coprimality across the $n$ randomly chosen polynomial pairs.

Our aim is to find the value of $a$ that satisfies the equation

$$\mathbb{P}(|\bar{X}_n - p| \leq a) = 1 - \alpha,$$

where $p$ denotes the true value of the target population parameter.
In other words, we are seeking the value of $a$ that ensures, with a probability of $1 - \alpha$, that the difference between the sample mean $\bar{X}_n$ and the true probability $p$ falls within the range $[-a, a]$.
The equation can be further transformed into

$$\mathbb{P}\left( -\frac{a}{\sqrt{\frac{p(1-p)}{n}}} \leq \frac{\bar{X}_n - p}{\sqrt{\frac{p(1-p)}{n}}} \leq \frac{a}{\sqrt{\frac{p(1-p)}{n}}} \right) = 1 - \alpha.$$

Since $X_1, \ldots, X_n$ are independent identically distributed Bernoulli random variables of parameter $p$, then $\sum_{i=1}^{n} X_i$, as well as $\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$, is a binomial random variable with parameters $n$ and $p$. Considering that $\mathrm{Var}(\frac{1}{n} \sum_{i=1}^{n} X_i) = \frac{p(1-p)}{n}$, due to the *Central Limit Theorem*[2]

$$\frac{\bar{X}_n - p}{\sqrt{\frac{p(1-p)}{n}}} \sim \mathcal{N}(0, 1).$$

This connection between the binomial distribution and the standard normal distribution allows us to express the values $-\frac{a}{\sqrt{\frac{p(1-p)}{n}}}$ and $\frac{a}{\sqrt{\frac{p(1-p)}{n}}}$ as the quantiles $z_{\frac{\alpha}{2}}$ and $z_{1-\frac{\alpha}{2}}$, respectively.
Note that, since we are working with a two tailed CI, we need to split the significance level $\alpha$ in half to find the relevant $z$-values from the standard normal distribution table.
We can write

$$\mathrm{CI}_p(1-\alpha) = \left[ \bar{X}_n + z_{\frac{\alpha}{2}} \sqrt{\frac{p(1-p)}{n}}; \bar{X}_n + z_{1-\frac{\alpha}{2}} \sqrt{\frac{p(1-p)}{n}} \right].$$

[2] For a detailed explanation of the Central Limit Theorem refer e.g. to [17] or [18].

Since we are estimating $p$ itself, we can replace $\mathrm{Var}(X_i) = p(1-p)$ with its unbiased estimator.

**Definition 6** Let $(X_1, X_2, \ldots, X_n)$ be a random sample from a given distribution. The *Sample Variance Estimator* $S_n^2$ is defined as

$$S_n^2 := \frac{\sum_{i=1}^{n}(X_i - \bar{X}_n)^2}{n-1}.$$

This procedure yields the following formulation of the searched confidence interval

$$\mathrm{CI}_p(1-\alpha) = \left[\bar{X}_n + z_{\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}}; \bar{X}_n + z_{1-\frac{\alpha}{2}}\frac{S_n}{\sqrt{n}}\right].$$

***Security level I***

In this section we will use the parameters for the first security level presented in Table 1. For the analysis conducted with these parameters we collect a sample of size $n = 10^5$. This large sample size contributes to a narrower CI, leading to a more precise estimation of the coprimality probability. We set a confidence level of 99% $(1 - \alpha = 0.99)$.

Let us start with the randomly chosen polynomial pairs of $\tilde{\mathcal{W}}_2$, obtained by using Procedure 1. Out of these 100 000 pairs, 89 083 are coprime. Consequently

$$\bar{X}_n = \frac{89\ 083}{100\ 000} = 0.89083,$$

$$S_n^2 = \frac{89\ 083(1-0.89083)^2 + 10\ 917(0-0.89083)^2}{99\ 999} \approx 0.09725,$$

$$\begin{aligned}\mathrm{CI}_p(0.99) &\approx \left[0.89083 - 2.576\sqrt{\frac{0.09725}{100\ 000}};\ 0.89083 + 2.576\sqrt{\frac{0.09725}{100\ 000}}\right] \\ &\approx [0.888;\ 0.894]\,.\end{aligned}$$

According to this analysis we can say that, with a 99% level of confidence, the pairs $(h_0, h_1)$ in $\tilde{\mathcal{W}}_2$ such that $h_0$ and $h_1$ are coprime in $\mathbb{F}_2[x]$ are between 88.8% and 89.4%. It is worth remembering that this probability is a lower bound to the probability that a pair in $\mathcal{W}_2$ is a weak key.

Let us now consider $\tilde{\mathcal{W}}_1^{\mathrm{low}}$ and $\tilde{\mathcal{W}}_1^{\mathrm{up}}$. First of all let us focus on the lower bound. Out of 100 000 pairs randomly selected from $\tilde{\mathcal{W}}_1^{\mathrm{low}}$ following Procedure 2, 88 857 are

coprime in $\mathbb{F}_2[x]$, consequently, the sample mean and the sample variance are

$$\bar{X}_n = \frac{88\ 857}{100\ 000} = 0.88857,$$

$$S_n^2 = \frac{88\ 857(1-0.88857)^2 + 11\ 143(0-0.88857)^2}{99\ 999} \approx 0.09901.$$

This yields the following confidence interval

$$\begin{aligned}\mathrm{CI}_p(0.99) &\approx \left[0.88857 - 2.576\sqrt{\frac{0.09901}{100\ 000}};\ 0.88857 + 2.576\sqrt{\frac{0.09901}{100\ 000}}\right] \\ &\approx [0.886;\ 0.892].\end{aligned}$$

Hence, we can say with a 99% level of confidence that, between 88.6% and 89.2% of pairs $(h_0, h_1)$ in the set $\tilde{\mathcal{W}}_1^{\mathrm{low}}$ are coprime in $\mathbb{F}_2[x]$.
As already mentioned, this value is a lower bound for the probability that an element in $\mathcal{W}_1^{\mathrm{low}}$ is a weak key.

Let us now focus on the upper bound case. Out of 100 000 pairs inside $\tilde{\mathcal{W}}_1^{\mathrm{up}}$ sampled following Procedure 3, 88 807 are coprime. This translates into the following sample mean and sample variance,

$$\bar{X}_n = \frac{88\ 807}{100\ 000} = 0.88807,$$

$$S_n^2 = \frac{88\ 807(1-0.88807)^2 + 11\ 193(0-0.88807)^2}{99\ 999} \approx 0.09940,$$

and into the following confidence interval

$$\begin{aligned}\mathrm{CI}_p(0.99) &\approx \left[0.88807 - 2.576\sqrt{\frac{0.09940}{100\ 000}};\ 0.88807 + 2.576\sqrt{\frac{0.09940}{100\ 000}}\right] \\ &\approx [0.885;\ 0.891].\end{aligned}$$

Hence, with a high probability, within the set $\tilde{\mathcal{W}}_1^{\mathrm{up}}$, between 88.5% and 89.1% of pairs are coprime in $\mathbb{F}_2[x]$. Consequently, it is likely that more that 88% of the pairs in $\mathcal{W}_1^{\mathrm{up}}$ are weak keys (referring to the proportion of pairs within the entire collection, including duplicates). Given such a high percentage, attempting to establish an upper bound for the probability that an element in $\mathcal{W}_1^{\mathrm{up}}$ can be represented as $(x^i h_0, x^i h_1)$ such that $\deg(h_0) + \deg(h_1) < r$ and $\gcd(h_0, h_1) = 1$ in $\mathbb{F}_2[x]$ would probably not lead to a relevant improvement over the following bound:

$$|K_1| \leq |\mathcal{W}_1^{\mathrm{up}}|\,.$$

In conclusion, with BIKE's first set of parameters $r = 12\ 323$ and $w = 142$, a conservative estimation is

$$\frac{1}{2}\left|\mathcal{W}_2\right| \leq \left|K_2\right| \qquad\qquad \frac{1}{2}\left|\mathcal{W}_1^{\text{low}}\right| \leq \left|K_1\right| \leq \left|\mathcal{W}_1^{\text{up}}\right|. \tag{10}$$

***Security level III***

In this section we set the parameters for security level III, as presented in Table 1. For the analysis conducted with these parameters, the size of the selected sample is $n = 10^4$, and the confidence level is set at 99%.

First, let us start with the set $\tilde{\mathcal{W}}_2$. Out of the 10 000 pairs randomly selected by using Procedure 1, 8 904 are coprime in $\mathbb{F}_2[x]$. Hence, the sample mean and the sample variance estimators are respectively given by

$$\bar{X}_n = \frac{8\ 904}{10\ 000} = 0.8904,$$

$$S_n^2 = \frac{8\ 904(1-0.8904)^2 + 1\ 096(0-0.8904)^2}{9\ 999} \approx 0.0976,$$

yielding the following confidence interval

$$\begin{aligned}\text{CI}_p(0.99) &\approx \left[0.8904 - 2.576\sqrt{\frac{0.0976}{10\ 000}};\ 0.8904 + 2.576\sqrt{\frac{0.0976}{10\ 000}}\right] \\ &\approx [0.882;\ 0.899]\,.\end{aligned}$$

Let us now consider $\tilde{\mathcal{W}}_1^{\text{low}}$ and $\tilde{\mathcal{W}}_1^{\text{up}}$. Firstly, let us focus on the lower bound case. Out of 10 000 pairs, sampled from $\tilde{\mathcal{W}}_1^{\text{low}}$ by following Procedure 2, 8 840 are coprime, consequently

$$\bar{X}_n = \frac{8\ 840}{10\ 000} = 0.8840,$$

$$S_n^2 = \frac{8\ 840(1-0.8840)^2 + 1\ 160(0-0.8840)^2}{9\ 999} \approx 0.1026,$$

$$\begin{aligned}\text{CI}_p(0.99) &\approx \left[0.8840 - 2.576\sqrt{\frac{0.1026}{10\ 000}};\ 0.8840 + 2.576\sqrt{\frac{0.1026}{10\ 000}}\right] \\ &\approx [0.875;\ 0.893]\,.\end{aligned}$$

For the upper bound case we get that out of 10 000 pairs, randomly selected from $\tilde{\mathcal{W}}_1^{\text{up}}$ by using Procedure 3, 8 867 are coprime. Hence,

$$\bar{X}_n = \frac{8\ 867}{10\ 000} = 0.8867,$$

$$S_n^2 = \frac{8\ 867(1-0.8867)^2 + 1\ 133(0-0.8867)^2}{9\ 999} \approx 0.1004.$$

$$\begin{aligned}\mathrm{CI}_p(0.99) &\approx \left[0.8867 - 2.576\sqrt{\frac{0.1004}{10\ 000}};\ 0.8867 + 2.576\sqrt{\frac{0.1004}{10\ 000}}\right] \\ &\approx [0.878;\ 0.895]\,.\end{aligned}$$

Consequently, also in this case attempting to establish an upper bound for the probability that an element in $\mathcal{W}_1^{\mathrm{up}}$ can be represented as $(x^i h_0, x^i h_1)$ such that $\deg(h_0) + \deg(h_1) < r$ and $\gcd(h_0, h_1) = 1$ in $\mathbb{F}_2[x]$ would probably not lead to a relevant improvement.

In conclusion with BIKE's third level parameters $r = 24\ 659$ and $w = 206$, a conservative estimation is

$$\frac{1}{2}|\mathcal{W}_2| \leq |K_2| \qquad\qquad \frac{1}{2}\left|\mathcal{W}_1^{\mathrm{low}}\right| \leq |K_1| \leq |\mathcal{W}_1^{\mathrm{up}}|\,. \tag{11}$$

*Remark 5* In Equations 10 and 11 we use the factor $\frac{1}{2}$ as it corresponds to one bit in the attacks' cost estimate (see the next section). Tighter factors could be used, but they would not lead to a significant improvement.

## 5.3 Attacks' cost

At this point, we can estimate the cost of the proposed attacks, bearing in mind that $\mathcal{H}_w$ represents the private key space.

- For the security level I we have that

$$|\mathcal{H}_w| = |\mathcal{P}_d| \approx 2^{1251.86};$$

- For the security level III we have that

$$|\mathcal{H}_w| = |\mathcal{P}_d| \approx 2^{1915.32}.$$

### 5.3.1 Attack 1

Let us now analyze the cost of the first attack that involves running Algorithm 2 on input $(x^r - 1, h, d)$.
The probability $\mathbb{P}_1$ of selecting a weak key is defined as the ratio between the number of private keys that can be broken through this attack and the total number of private keys, described in Equation (7). To be precise, this is a lower bound since we can not exclude the possibility that there exist keys breakable in this way outside $K_1$, but we will consider it as an estimation.

***Security level I***

Given the reasoning, experimental results, and discussion provided above, and considering the parameter set for the first security level, we obtain that

$$2^{1119.33} \approx \frac{1}{2}|\mathcal{W}_1^{\text{low}}| \leq |K_1| \leq |\mathcal{W}_1^{\text{up}}| \approx 2^{1121.33}.$$

Hence,

$$\frac{1}{2^{132.53}} \approx \frac{|\mathcal{W}_1^{\text{low}}|}{2\,|\mathcal{H}_w|} \leq \mathbb{P}_1 \leq \frac{|\mathcal{W}_1^{\text{up}}|}{|\mathcal{H}_w|} \approx \frac{1}{2^{130.53}}.$$

Let us recall that, as stated in Section 3, the cost of Algorithm 2 in our scenario is

$$\mathcal{C}_{\text{Algorithm 2}} = \mathcal{O}(r^2) \approx 2^{27.18}.$$

In conclusion, the cost of the attack, defined as the ratio between the cost of Algorithm 2 and the probability that the attack is successful, i.e.

$$\mathcal{C}_{\text{Attack 1}} := \frac{\mathcal{C}_{\text{Algorithm 2}}}{\mathbb{P}_1},$$

can be estimated by the following bounds.

$$2^{158} \approx \frac{2^{27.18}}{\frac{1}{2^{130.53}}} \leq \mathcal{C}_{\text{Attack 1}} \leq \frac{2^{27.18}}{\frac{1}{2^{132.53}}} \approx 2^{160}.$$

***Security level III***

Fixing the parameters to those advised for the third security level, we get that

$$2^{1719.57} \approx \frac{1}{2}|\mathcal{W}_1^{\text{low}}| \leq |K_1| \leq |\mathcal{W}_1^{\text{up}}| \approx 2^{1721.56}.$$

Consequently,

$$\frac{1}{2^{195.75}} \approx \frac{|\mathcal{W}_1^{\text{low}}|}{2\,|\mathcal{H}_w|} \leq \mathbb{P}_1 \leq \frac{|\mathcal{W}_1^{\text{up}}|}{|\mathcal{H}_w|} \approx \frac{1}{2^{193.76}}.$$

Since the cost of Algorithm 2 is

$$\mathcal{C}_{\text{Algorithm 2}} = \mathcal{O}(r^2) \approx 2^{29.18},$$

we obtain

$$2^{223} \approx \frac{2^{29.18}}{\frac{1}{2^{193.76}}} \leq \mathcal{C}_{\text{Attack 1}} \leq \frac{2^{29.18}}{\frac{1}{2^{195.75}}} \approx 2^{225}.$$

### 5.3.2 Attack 2

Let us now evaluate the cost of the second attack, which requires executing Algorithm 3 on input $(x^r - 1, h, d)$.

It is worth noting that the cost of this second attack can not be lower than the first one presented in Section 5.3.1. This is because Algorithm 3 has a cost that is $r$ times higher, while the number of weak keys that can be exploited by this attack does not increase proportionally by a factor of $r$. Nevertheless, for the sake of completeness, we still show in detail its cost.
The probability $\mathbb{P}_2$ of selecting a weak key for this case, defined as the ratio between the number of private keys breakable with this attack and the cardinality of the private key space is described in Equation (7).
Note that, also in this case, this ratio is a lower bound estimation, but we will treat it as an approximation.

***Security level I***

By fixing the parameters to those advised for the first level of security we get

$$2^{1126.47} \approx \frac{1}{2} |\mathcal{W}_2| \leq |K_2| .$$

Hence, the probability of selecting a weak key is bounded by the value

$$\frac{1}{2^{125.39}} \approx \frac{|\mathcal{W}_2|}{2\,|\mathcal{H}_w|} \leq \mathbb{P}_2$$

and, as already seen, the cost of Algorithm 3 is estimated by

$$\mathcal{C}_{\text{Algorithm 3}} = \mathcal{O}(r^3) \approx 2^{40.77}.$$

In conclusion, as the cost of an attack is defined as the ratio between the resources needed to execute it and the probability of success, i.e.

$$\mathcal{C}_{\text{Attack 2}} := \frac{\mathcal{C}_{\text{Algorithm 3}}}{\mathbb{P}_2},$$

we obtain

$$\mathcal{C}_{\text{Attack 2}} \leq \frac{2^{40.77}}{\frac{1}{2^{125.39}}} \approx 2^{166}.$$

Despite this is an upper bound, we expect that the cost $\mathcal{C}_{\text{Attack 2}}$, as defined above, will not be too far from it.

***Security level III***

By fixing the parameters to those advised for the third level of security, we get that

$$2^{1727.25} \approx \frac{1}{2} |\mathcal{W}_2| \leq |K_2| .$$

Hence,

$$\frac{1}{2^{188.07}} \approx \frac{|\mathcal{W}_2|}{2\ |\mathcal{H}_w|} \leq \mathbb{P}_2.$$

The cost of Algorithm 3 is estimated by

$$\mathcal{C}_{\text{Algorithm 3}} = \mathcal{O}(r^3) \approx 2^{43.77}.$$

In conclusion an upper bound, that will not be too far from the real value, for the cost of this attack is

$$\mathcal{C}_{\text{Attack 2}} \leq \frac{2^{43.77}}{\frac{1}{2^{188.07}}} \approx 2^{232}.$$

## 5.4 Results

Table 2 summarizes the attack costs, in the specific setting where it is assumed that a weak key is used, and the probability of selecting a weak key.

| Security level | Attack 1 | | Attack 2 | |
|---|---|---|---|---|
| I | $2^{-132.53} \leq \mathbb{P} \leq 2^{-130.53}$ | $\mathcal{C} \approx 2^{27.18}$ | $2^{-125.39} \leq \mathbb{P}$ | $\mathcal{C} \approx 2^{40.77}$ |
| III | $2^{-195.75} \leq \mathbb{P} \leq 2^{-193.76}$ | $\mathcal{C} \approx 2^{29.18}$ | $2^{-188.07} \leq \mathbb{P}$ | $\mathcal{C} \approx 2^{43.77}$ |

**Table 2**: Comparison of the probability of selecting a weak key (denoted by $\mathbb{P}$) and the costs of running the attack on a single instance (denoted by $\mathcal{C}$) for BIKE.

*Remark 6* In [12], for Attack 1, the authors identify only the set $W_1$ (Equation 2), whose elements, within the BIKE setting, are selected with probabilities approximately equal to $2^{-138.67}$ and $2^{-202.44}$ for security levels I and III, respectively. By identifying the set $\mathcal{W}_1$ we instead obtain that the probability of selecting a weak key for Attack 1 is at least $2^{-132.53}$ and $2^{-195.75}$ for security levels I and III, respectively.

We conclude this section with a comparison table. This table presents the costs of the attacks described, alongside the original Prange's ISD algorithm and the currently most efficient attack on BIKE due to May and Ozerov [11], computed accordingly to *CryptographicEstimators*[3] [19], when the parameters for the first and third levels of security are used.

*Remark 7* Within the BIKE setting, the results obtained in [12] for Attack 1 translate into an estimated cost of $2^{165.85}$ and $2^{231.62}$ for security levels I and III, respectively. Our work shows that Attack 1 can be carried out with a cost lower than $2^{160}$ and $2^{225}$ for security levels I and III, respectively.

[3] https://github.com/Crypto-TII/CryptographicEstimators

| Security level | Attack 1 | Attack 2 | Prange's ISD | Best known attack |
|---|---|---|---|---|
| I | $2^{158} < \mathcal{C} < 2^{160}$ | $\mathcal{C} \approx 2^{166}$ | $\mathcal{C} \approx 2^{174}$ | $\mathcal{C} \approx 2^{146}$ |
| III | $2^{223} < \mathcal{C} < 2^{225}$ | $\mathcal{C} \approx 2^{232}$ | $\mathcal{C} \approx 2^{241}$ | $\mathcal{C} \approx 2^{210}$ |

**Table 3**: Comparison of attack costs (denoted by $\mathcal{C}$) against BIKE.

# 6 Conclusions

The framework introduced in [12] has started a novel line of algebraic cryptanalysis for QC-MDPC code-based cryptosystems, highlighting possible vulnerabilities of BIKE and similar schemes. In this work we have presented two extensions of the state of the art.

- The first one is the introduction of a new necessary hypothesis for successful attacks, namely, the weak keys have to be pairs of coprime polynomials. Due to this, we were able to provide more accurate estimations of the computational cost of the key-recovery in [12].
- The second one is the introduction of a variant of a known attack, corresponding to a new larger set of weak keys that leads to an improvement of the attack cost. Within the BIKE setting this corresponds to an improvement of at least 5 and 6 bits for security levels I and III, respectively.

Overall, the analysis of the computational costs shows how this framework is quite promising, as highlighted in particular by the comparison with the cost of Prange ISD. It is important to notice, however, that despite the advancements presented in this paper, the class of algebraic key-recovery attacks described does not currently break BIKE.

As extensions of this work, we plan to understand whether techniques such as enumeration, meet-in-the-middle search and representations can be borrowed from the ISD framework and applied to our algorithm, as well. Moreover, we plan to understand whether our approach can be used to produce codewords whose weight is moderately small (say, something in between the minimum distance and the Gilbert-Varshamov bound). Existence of such codewords would first provide a distinguisher for QC-MDPC codes. Moreover, from the knowledge of such codewords one may obtain the secret key, using techniques such as [20, 21].

# 7 Declarations

## 7.1 Availability of data and materials

Not applicable.

## 7.2 Competing interests

Not applicable.

### 7.3 Funding

This work was partially supported by Ministero delle Imprese e del Made in Italy (IPCEI Cloud DM 27 giugno 2022 – IPCEI-CL-0000007) and European Union (Next Generation EU). This publication was created with the co-financing of the European Union FSE-REACT-EU, PON Research and Innovation 2014-2020 DM1062/2021. This work was partially supported by the project PRIN 2022SC, title "Algebraic Methods in Cryptanalysis", Grant Ref. 2022RFAZCJ, CUP H53C24000830006.

### 7.4 Authors' contributions

All authors contributed equally to this work. This research was initiated at the University of Trento, Trento, Italy, and the core of this work was part of the second author's M.Sc. thesis.

### 7.5 Acknowledgements

The authors express their sincere gratitude to Paolo Santini for suggesting the idea of investigating this research direction, providing an initial spark for the study, and always being helpful with any questions.
They also thank the reviewers for their constructive comments, which significantly improved the manuscript.

### 7.6 Authors' information

The first author is a member of the INdAM Research Group GNSAGA and is currently affiliated with the University of Bari Aldo Moro, Bari, Italy. The second author is currently affiliated with both the University of Trento, Trento, Italy and the Bruno Kessler Foundation, Trento, Italy.

## References

[1] Gallager, R.G.: Low-Density Parity-Check Codes. Ph.d. thesis, Massachusetts Institute of Technology, Cambridge, MA (1963). https://dspace.mit.edu/handle/1721.1/11804

[2] Ouzan, S., Be'ery, Y.: Moderate-Density Parity-Check Codes. arXiv preprint arXiv:0911.3262 (2009)

[3] Drucker, N., Gueron, S., Kostic, D.: QC-MDPC Decoders with Several Shades of Gray. In: Ding, J., Tillich, J.-P. (eds.) Post-Quantum Cryptography, pp. 35–50. Springer, Cham (2020)

[4] Aragon, N., Barreto, P.L., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.-C., Gaborit, P., Ghosh, S., Gueron, S., Güneysu, T., Aguilar Melchor, C., Misoczki, R., Persichetti, E., Richter-Brockmann, J., Sendrier, N., Tillich, J.-P., Zémor, G., Vasseur, V.: BIKE: Bit Flipping Key Encapsulation. Round 4 Submission. https://bikesuite.org/ (2024)

[5] Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G., Santini, P.: LEDAcrypt: LowdEnsity parity-check coDe-bAsed cryptographic systems. https://www.ledacrypt.org/ (2020)

[6] Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory **8**(5), 5–9 (1962)

[7] Lee, P.J., Brickell, E.F.: An Observation on the Security of McEliece's Public-Key Cryptosystem. In: Barstow, D., Brauer, W., Brinch Hansen, P., Gries, D., Luckham, D., Moler, C., Pnueli, A., Seegmüller, G., Stoer, J., Wirth, N., Günther, C.G. (eds.) Advances in Cryptology — EUROCRYPT '88, pp. 275–280. Springer, Berlin, Heidelberg (1988)

[8] Leon, J.S.: A probabilistic algorithm for computing minimum weights of large error-correcting codes. IEEE Transactions on Information Theory **34**(5), 1354–1359 (1988)

[9] Stern, J.: A method for finding codewords of small weight. In: Cohen, G., Wolfmann, J. (eds.) Coding Theory and Applications, pp. 106–113. Springer, Berlin, Heidelberg (1989)

[10] 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, pp. 520–536. Springer, Berlin, Heidelberg (2012)

[11] May, A., Ozerov, I.: On Computing Nearest Neighbors with Applications to Decoding of Binary Linear Codes. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology – EUROCRYPT 2015, pp. 203–228. Springer, Berlin, Heidelberg (2015)

[12] Bardet, M., Dragoi, V., Luque, J.-G., Otmani, A.: Weak Keys for the Quasi-Cyclic MDPC Public Key Encryption Scheme. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) Progress in Cryptology – AFRICACRYPT 2016, pp. 346–367. Springer, Cham (2016)

[13] McEliece, R.J.: The Theory of Information and Coding. Encyclopedia of Mathematics and its Applications, vol. 3. Addison-Wesley, Reading, MA (1977)

[14] Bonde, V., Siktar, J.M.: On the Combinatorics of Placing Balls into Ordered Bins. Integers: Electronic Journal of Combinatorial Number Theory **21** (2021)

[15] Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems of Control and Information Theory **15**(2), 157–166 (1986)

[16] Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symbolic Comput. **24**(3-4), 235–265 (1997) https://doi.org/10.100

6/jsco.1996.0125 . Computational algebra and number theory (London, 1993)

[17] Hogg, R.V., Craig, A.T.: Introduction to Mathematical Statistics, 4th edn. Macmillan, New York (1978)

[18] Bickel, P.J., Doksum, K.A.: Mathematical Statistics: Basic Ideas and Selected Topics, Volume I, 2nd edn. Chapman and Hall/CRC, New York (2015). https://doi.org/10.1201/b18312

[19] Esser, A., Verbel, J., Zweydinger, F., Bellini, E.: SoK: CryptographicEstimators – a Software Library for Cryptographic Hardness Estimation. In: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, pp. 560–574. Association for Computing Machinery, New York, NY, USA (2024). https://doi.org/10.1145/3634737.3645007

[20] Pellegrini, A., Tognolini, G.: Breaking HWQCS: a code-based signature scheme from high weight QC-LDPC codes. In: Code-Based Cryptography Workshop, pp. 67–93 (2024). Springer

[21] Santini, P., Baldi, M., Chiaraluce, F.: Cryptanalysis of a One-Time Code-Based Digital Signature Scheme. In: 2019 IEEE International Symposium on Information Theory (ISIT), pp. 2594–2598 (2019). IEEE