Weight of Polynomial Products Mod $(X^n + 1)$ –Application to the HQC Cryptosystem–

Laila El Aimani

University of Cadi Ayyad - ENSA Safi
Route Sidi Bouzid BP 63, 46000 - Safi, Morocco
laila.elaimani@gmail.com

Abstract. We consider the problem of computing the variance and tail probability bounds for the weight of the product $z = x \cdot y$ of two random polynomials $x, y \in \mathbb{F}_2[X]/(X^n + 1)$, where the weight of a binary polynomial is defined as the number of its nonzero coefficients.
We investigate two probabilistic models for x and y: the *uniform slice* model, where the weights are fixed to w_x and w_y, and the *binomial* model, where the coefficients are independent Bernoulli variables with parameters p_x and p_y.
Our analysis is directly motivated by the need for accurate security analysis of the error vector of the HQC code-based encryption scheme. Prior work has progressed from heuristic arguments backed by extensive simulations to exact computations of the probability mass function (PMF) of the weight of the error vector; however, these computations were largely restricted to the uniform slice setting.
We adopt a different approach, leveraging the full covariance structure of the product vector z (i.e., the pairwise covariances of its coefficients) to derive tight, rigorously proven bounds on the tail probabilities of its weight for both the uniform and binomial models. These results confirm HQC's security guarantees and, moreover, reveal additional symmetry properties of the HQC error vector that are desirable for cryptographic design.

Keywords: Hamming weight · product of polynomials · covariance analysis · tail probability · decryption failure · HQC

1 Introduction

We consider the following problem. Given a positive integer n and two polynomials x, y in the ring $\mathbb{F}_2[X]/(X^n + 1)$, we are interested in the Hamming weight (the number of nonzero coefficients) of their product. More specifically, we want to compute the variance and tail probability of the weight of $x \cdot y \bmod (X^n + 1)$ in the following two models:

1. **The Uniform Slice Model.** The polynomials x and y have fixed weights w_x and w_y, respectively.
2. **The Binomial Model.** x and y follow a binomial distribution: their coefficients are independent and identically distributed (iid) random variables, each equal to 1 with probability p_x (resp. p_y).

1.1 Motivation

The problem finds a direct application in the HQC (Hamming Quasi-Cyclic) [7] code-based encryption scheme, which was selected by the National Institute of Standards and Technology (NIST) as a backup to the winner of the competition for the first post-quantum cryptography standard (ML-KEM, also known as Kyber [3]).

In a nutshell, a ciphertext in the HQC scheme consists of a binary codeword (encoding a plaintext) to which is added a random binary error vector. Decryption eliminates most of the errors using the decryption key to obtain a noisy version of the plaintext, and then recovers

the noiseless plaintext using an efficient decoding algorithm. Since the error vector is randomly generated, HQC has a nonzero decryption failure rate: decryption occasionally fails with some probability. To recover the plaintext efficiently, the weight of the error vector must not exceed the error-correction capacity of the underlying code. This requires a careful study of the error vector e' in HQC, which is given by

$$e' = x \cdot r_2 + y \cdot r_1 + e,$$

where x, y, r_1, r_2, e are independent uniformly random polynomials in $\mathbb{F}_2[X]/(X^n+1)$ with fixed weights w, w, w_1, w_2, w_e, respectively.

Clearly, the security analysis of HQC involves studying the expectation, variance and tail probability of the weight of polynomial products modulo $X^n + 1$. While the expectation of the weight is already well understood and has been derived in prior works (see, e.g., [10,7]), the main challenge lies in the variance and the underlying covariance structure, which are the focus of this work.

The following elementary lemmas give the expectation of $\mathsf{weight}(x \cdot y)$ under the two models; these formulae will be used later in our analysis.

Lemma 1 ([7]-Uniform Slice Expectation-). *Let $x, y \in \mathbb{F}_2[X]/(X^n + 1)$ be independent uniformly random polynomials with respective fixed weights w_x and w_y. Let $z = x \cdot y$. Then, for all i:*

$$\Pr[z_i = 1] = \sum_{\substack{s=0 \\ s \text{ odd}}}^{\min(w_x, w_y)} \frac{\binom{w_x}{s}\binom{n-w_x}{w_y-s}}{\binom{n}{w_y}}$$

Lemma 2 ([10]-Binomial Model Expectation-). *Let $x, y \in \mathbb{F}_2[X]/(X^n + 1)$ have independent coefficients with $\Pr(x_i = 1) = p_x$ and $\Pr(y_i = 1) = p_y$ for all i. Let further $z = x \cdot y$. Then, for all i:*

$$\Pr[z_i = 1] = \frac{1}{2}\left(1 - (1 - 2p_x p_y)^n\right).$$

By linearity of expectation, $\mathbb{E}(\mathsf{weight}(x \cdot y))$ is simply $n \cdot \Pr[z_i = 1]$.

1.2 Related Work

Let x, y be two independent uniformly random polynomials in $\mathbb{F}_2[X]/(X^n + 1)$ with respective weights w_x and w_y, and let $z = x \cdot y \bmod (X^n + 1)$.

The study of the weight distribution of z was initiated in the context of HQC [10]. To make the analysis tractable, the authors introduced two simplifying assumptions:

1. the coefficients of x and y are pairwise independent (binomial model); and
2. the coefficients of z are pairwise independent.

Under these assumptions, they derived heuristic estimates for the variance and tail probability of $\mathsf{weight}(z)$, which they corroborated with empirical simulations.

The first assumption was later dropped by the HQC team in their updated analysis [7]. Indeed, imposing pairwise independence in key generation and encryption introduces significant computational overhead and requires modifying the security proof, often at the cost of tightness.

The second assumption—the pairwise independence of the coefficients of z—has also come under scrutiny. Bombar et al. [6] provided evidence that this model fails in certain parameter regimes, indicating that the dependency structure of z is more subtle than previously assumed.

On the rigorous side, Kawachi [8] established the first formally proven bounds for the variance and tail probability, specifically within the binomial model. Nevertheless, these bounds are

substantially looser than the original heuristic estimates and, more critically, do not generalize directly to the uniform weight distribution or to the specific structure of the HQC error vector. In particular, a bound on the variance of $\mathsf{weight}(x \cdot y)$ does not automatically translate into a bound on the variance of the weight of the HQC error vector, thereby restricting the practical relevance of that approach. Kawachi later improved upon these bounds in [9]. However, this refined result remains confined to the binomial model, does not carry over to the HQC error vector analysis, and is valid only when x and y share the same Bernoulli parameter.

In a preliminary version of this work [1], we introduced exact and fully implementable computations of the covariance coefficients of the product $x \cdot y$ for both the binomial and uniform slice models. These coefficients yield the variance of $\mathsf{weight}(x \cdot y)$ and, more importantly, enable the computation of the covariance coefficients of the HQC error vector itself; this, in turn, gives the variance of the error vector's weight, thereby providing the first rigorously proven tight bound on the decryption failure rate.

Following this initial contribution, Bitzer et al. [4] revisited the heuristic framework of [7] for the uniform setting. Drawing on techniques from the analysis of moderate-density parity-check (MDPC) codes, they modeled the dependencies among the coefficients of z by exploiting the combinatorial constraint that, viewed as polynomials over $\mathbb{Z}$, the product $x \cdot y$ has weight $w_x w_y$. This led to refined empirical security estimates, which they supported with extensive simulations.

Most recently, Annechini et al. [2] obtained a closed-form expression for the full probability mass function (PMF) of $\mathsf{weight}(x \cdot y)$—and consequently of the weight of the HQC error vector—in the uniform case. This represents a significant quantitative advance, as it permits the computation of exact tail probabilities in addition to the variance.

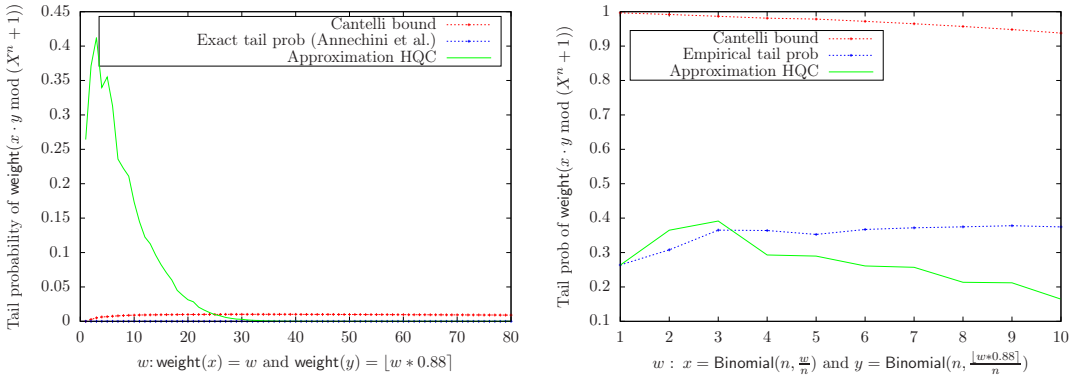
1.3 Our Contributions

To summarize the current state of the art, the methods have progressed from heuristic approximations to exact computation of the PMF of the weight in the uniform case, enabling the exact determination of both variance and tail probabilities. Our preliminary work [1] remains pioneering in providing the first exact expressions for the variance in both the uniform slice model and the binomial case, and consequently for the variance of the HQC error vector's weight. This yielded the first rigorously proven tight tail probabilities for the above-mentioned quantities.

In this submission, we do not modify the core results; rather, we explicitly highlight a subtle property that follows directly from our covariance framework. This observation enriches the overall understanding of the HQC error vector and demonstrates the versatility of the covariance-based approach.

More precisely, this work makes the following contributions:

1. **Explicit formulae for the covariance coefficients of the product of polynomials.** We present exact expressions for the covariance coefficients of the product $z = x \cdot y$ (viewed as an n-bit vector) for random polynomials x and y in $\mathbb{F}_2[X]/(X^n + 1)$ under two distinct distributions: the uniform slice model and the binomial model. These expressions allow us to derive the exact variance and bounds on the tail probabilities of the weight of the product using Cantelli's inequality.
   Our analysis shows that the weight of z is sharply concentrated around its mean in the uniform case. In the binomial model, strong concentration holds only for large Bernoulli parameters.
2. **Application to the HQC scheme.** We extend our covariance analysis to the HQC scheme by deriving exact formulae for the covariance coefficients of the error vector under both the



(a) Uniform slice model (b) Binomial model

Fig. 1: Upper bound on the tail probability $\Pr[\mathsf{weight}(x \cdot y) \geq 1.1\,\mathbb{E}(\mathsf{weight}(x \cdot y))]$ for $n = 17769$ and $w \in [0, 80]$

uniform slice and binomial models. This enables a security reassessment of HQC, structured as follows:

- First, we derive exact formulae for the variance of the weight of the HQC error vector. Then, using Cantelli's inequality, we verify the security guarantees established in [10,7] for the uniform model; however, we find that these guarantees hold for the binomial instantiation only when the Bernoulli parameter is sufficiently large.
- Second, we observe that choosing n prime in the HQC scheme implies that all covariance coefficients of the error vector are equal. This symmetry property is desirable in both coding-theoretic and cryptographic applications.

2 Overview of Our Techniques

2.1 Tools and Notations

Generalities. Let $\mathbb{Z}$ denote the set of integers and $\mathbb{N}$ the set of natural numbers. The absolute value of an integer i is denoted by $|i|$. For real numbers a and b, we write $[a, b]$ and (a, b) for the closed and open intervals with endpoints a and b, respectively. For nonnegative integers n and k, the binomial coefficient $\binom{n}{k}$ is defined as $\frac{n!}{k!(n-k)!}$ if $k \leq n$, and 0 otherwise.

Let $a, b, n \in \mathbb{Z}$ with $n > 0$, and let $\mathbb{Z}_n$ denote the ring of integers modulo n. The modulo notation is used in two related ways, distinguishable from context. First, $b = a \bmod n$ denotes the binary operation yielding the unique remainder b satisfying $0 \leq b < n$. Second, the congruence relation $a \equiv b \pmod{n}$ holds if and only if n divides $a - b$. Note that if $a \equiv b \pmod{n}$ and $0 \leq b \leq n - 1$, then $b = a \bmod n$.

We work in the ring $\mathbb{F}_2[X]/(X^n + 1)$ of binary polynomials modulo $X^n + 1$, where addition and multiplication are performed modulo $X^n + 1$. Throughout, a polynomial in this ring is identified with its coefficient vector in $\{0, 1\}^n$. For such a vector $x = (x_0, \ldots, x_{n-1})$, we denote by $\mathsf{weight}(x)$ its Hamming weight, i.e., the number of nonzero coordinates.

For sets A and B, $A \cap B$ denotes their intersection, $A \setminus B$ their set difference, and $|A|$ and $|B|$ their respective cardinalities. We analyze the complexity of our algorithms using the standard Big-O notation, which describes worst-case asymptotic behavior.

Probabilities. The probability mass function (PMF) of a random variable is denoted by $\mathrm{PMF}(\cdot)$. For a random variable X, $\mathbb{E}(X)$ and $\mathrm{Var}(X)$ denote its expectation and variance, respectively.

For two binary random variables X_1 and X_2, their covariance is defined as

$$\mathrm{Cov}(X_1, X_2) = \Pr[X_1 = 1, X_2 = 1] - \Pr[X_1 = 1]\Pr[X_2 = 1].$$

If $X = X_1 + \cdots + X_n$ is a sum of binary random variables, then

$$\mathrm{Var}(X) = \sum_{1 \le k,l \le n} \mathrm{Cov}(X_k, X_l).$$

For a positive integer n and a real $p \in [0, 1]$, we write $X \sim \mathsf{Bernoulli}(p)$ to indicate that X is a binary random variable with $\Pr[X = 1] = p$ and $\Pr[X = 0] = 1 - p$. We write $(X_1, \ldots, X_n) \sim \mathsf{Binomial}(n, p)$ to indicate that the X_i's are independent and identically distributed with $X_i \sim \mathsf{Bernoulli}(p)$.

Tail probabilities quantify the deviation of a random variable from its expectation. We use **Cantelli's inequality**, which states that for any $\lambda > 0$,

$$\Pr[X - \mathbb{E}(X) > \lambda] < \frac{\mathrm{Var}(X)}{\lambda^2 + \mathrm{Var}(X)}.$$

Graphs. A graph G is defined by its vertex set $V(G)$ and its edge set $E(G)$. An edge connecting vertices v_i and v_j is denoted by (v_i, v_j). All graphs considered in this work are undirected.

A cycle graph C is an undirected graph consisting of a single cycle. If $V(C) = \{v_1, \ldots, v_n\}$, we write C as $v_1 - v_2 - \cdots - v_n - v_1$. A linear chain L is an undirected graph whose vertices form a single sequence without cycles; if $V(L) = \{v_1, \ldots, v_n\}$, we write L as $v_1 - v_2 - \cdots - v_n$.

Dynamic Programming. A central tool in our work is dynamic programming. This method decomposes a complex problem into simpler overlapping subproblems, solves each exactly once, and stores the solution for future reference. Our dynamic programming solutions are expressed as recurrence relations that define each term from previously computed values, complemented by well-specified base conditions. We implement these recurrences bottom-up, starting from the base cases and iteratively populating a state table until we obtain the final solution to the original problem.

2.2 Overview of Our Approach

Let Z be the weight of the product $x \cdot y$ in $\mathbb{F}_2[X]/(X^n+1)$, where x and y are random polynomials. Our goal is to compute $\mathrm{Var}(Z)$. To this end, we write Z as a sum of n binary random variables:

$$Z = Z_0 + Z_1 + \cdots + Z_{n-1},$$

where each Z_i indicates whether the i-th coefficient of the product is 1. The variance of Z is then given by

$$\mathrm{Var}(Z) = \sum_{0 \le k,l \le n-1} \mathrm{Cov}(Z_k, Z_l).$$

The diagonal terms $\mathrm{Cov}(Z_k, Z_k) = \mathbb{E}(Z_k)(1 - \mathbb{E}(Z_k))$ are straightforward, since $\mathbb{E}(Z_k)$ is provided by Lemmas 1 and 2. The main difficulty lies in computing $\mathrm{Cov}(Z_k, Z_l)$ for $k \neq l$, which requires the joint probability $\Pr[Z_k = 1, Z_l = 1]$.

To address this, we introduce an auxiliary random variable S that conditions on the structure of Z_k and Z_l, effectively reducing their correlation. The joint probability $\Pr[Z_k = 1, Z_l = 1]$ is then completely determined by the PMF of S. Computing this PMF is a central contribution of this work and spans two sections. We give a graph-theoretic interpretation of the problem,

reducing the computation of the PMF to counting edges with specified properties in a particular graph. We then show how to perform this counting when the graph is a cycle, using auxiliary functions that we compute via dynamic programming. Finally, using elementary number theory, we prove that the relevant graph is always either a single cycle or a disjoint union of cycles. This disjointness allows us to treat each component independently, which ultimately yields a complete characterization of the PMF of the auxiliary variable S.

3 Variance

Let x, y be random elements from $\mathbb{F}_2[X]/(X^n + 1)$. Let further $z = (z_0, \ldots, z_{n-1})$ denote the product $x \cdot y \bmod (X^n + 1)$. In this section, we compute the variance of $Z = \mathsf{weight}(z)$ across the following models:

1. **The Uniform Slice Model:** x and y are uniform random n-bit vectors with fixed weights w_x and w_y, respectively.
2. **The Binomial Model:** x and y are n-bit vectors, with iid Bernoulli(p_x) and Bernoulli(p_y) coordinates, respectively.

We have $Z = \sum_{k=0}^{n-1} Z_k$, where $Z_k = z_k$ (equivalently, $Z_k = \mathbf{1}_{z_k=1}$), and hence

$$Z_k = \bigoplus_{0 \le i < n} x_i y_{(k-i) \bmod n}.$$

Note that the Z_k's are not independent; thus we cannot simply sum their individual variances to compute $\mathrm{Var}(Z)$. Instead, we use the standard definition

$$\mathrm{Var}(Z) = \sum_{k,l} \mathrm{Cov}(Z_k, Z_l),$$

where

$$\mathrm{Cov}(Z_k, Z_l) = \Pr[Z_k = 1, Z_l = 1] - \Pr[Z_k = 1] \Pr[Z_l = 1].$$

For $k = l$, we have $\mathrm{Cov}(Z_k, Z_k) = \mathbb{E}[Z_k](1 - \mathbb{E}[Z_k])$, where $\mathbb{E}[Z_k] = \Pr[z_k = 1]$ was computed in Lemmas 1 and 2. It remains to compute $\mathrm{Cov}(Z_k, Z_l)$, i.e., the joint probability $\Pr[Z_k = 1, Z_l = 1]$ for $k \neq l$. Without loss of generality, we assume $k < l$ in the rest of this section.

We draw upon the idea of [8] at the outset. That is, we aim to reduce the correlation between Z_k and Z_l by conditioning on x. In this case Z_k and Z_l reduce to $\bigoplus_{i \in S_k} Y_i$ and $\bigoplus_{i \in S_l} Y_i$, respectively, where S_k and S_l denote the sets of surviving indices following the assignment of a value to x. Let us give an example for the uniform slice case. Suppose $n = 5$, $(k, l) = (3, 4)$, and $w_x = 3$. Suppose further that we assign the value $(1, 0, 1, 1, 0)$ to $x = (x_0, x_1, x_2, x_3, x_4)$. Then, under this assignment, $Z_k = \bigoplus_{0 \le i < n} x_i y_{(k-i) \bmod n}$ becomes $Z_3 = y_0 \oplus y_1 \oplus y_3$ and $Z_l = \bigoplus_{0 \le i < n} x_i y_{(l-i) \bmod n}$ becomes $Z_4 = y_1 \oplus y_2 \oplus y_4$. In other terms, the surviving indices for the y_i's involved in Z_k and Z_l are $\{0, 1, 3\}$ and $\{1, 2, 4\}$, respectively. More generally, we define the set of surviving indices for $k \in [0, n-1]$ as follows

$$S_k = \{(k - i) \bmod n : x_i = 1\}.$$

One can easily observe that the size of S_k equals $\mathsf{weight}(x)$ regardless of k.

Back to the previous example, once x has been assigned the value $(1, 0, 1, 1, 0)$, the conditional probability $\Pr[Z_3 = 1, Z_4 = 1 \mid x = (1, 0, 1, 1, 0)]$ becomes

$$\begin{aligned}\Pr[Z_3 = 1, Z_4 = 1 \mid x = (1, 0, 1, 1, 0)] &= \Pr[y_0 \oplus y_1 \oplus y_3 = 1, y_1 \oplus y_2 \oplus y_4 = 1] \\ &= \sum_{b \in \{0,1\}} \Pr[Y_1 = b, y_0 \oplus y_3 = 1 - b, y_2 \oplus y_4 = 1 - b].\end{aligned}$$

More generally, for a fixed value $x \in \{0,1\}^n$ with $\mathsf{weight}(x) = w_x$,

$$\Pr[Z_k = 1, Z_l = 1 \mid x] = \sum_{b \in \{0,1\}} \Pr\left[\bigoplus_{i \in S_k \cap S_l} y_i = b, \bigoplus_{i \in S_k \setminus S_l} y_i = 1-b, \bigoplus_{i \in S_l \setminus S_k} y_i = 1-b\right].$$

At this point, we diverge from the approach of [8] that computes an upper bound for $\Pr[\bigoplus_{i \in S_k \cap S_l} y_i = b, \bigoplus_{i \in S_k \setminus S_l} y_i = 1-b, \bigoplus_{i \in S_l \setminus S_k} y_i = 1-b]$ under the assumption of the *independence of the y_i's*, which consequently provides an upper bound on $\mathrm{Var}(Z)$.

In fact, we compute precisely $\Pr[Z_k = 1, Z_l = 1]$ through determining the joint PMF of the random variables $(|S_k \cap S_l|, |S_k \setminus S_l|, |S_l \setminus S_k|)$ across two models: (1) the uniform slice model, where x is an n-bit vector with fixed weight w_x, and (2) the binomial model where $x \sim \mathsf{Binomial}(n, p_x)$.

The Uniform Slice Model. In this model, the size of the sets S_k and S_l equals w_x. Thus, the size of the sets $S_k \setminus S_l$ and $S_l \setminus S_k$ is entirely determined by the size of $S_k \cap S_l$, namely, $|S_k \setminus S_l| = |S_l \setminus S_k| = w_x - |S_k \cap S_l|$. It suffices then to compute the PMF of the random variable $|S_k \cap S_l|$. The joint probability $\Pr[Z_k = 1, Z_l = 1]$ is given by:

$$\begin{aligned}
\Pr[Z_k = 1, Z_l = 1] &= \sum_{\substack{x \in \{0,1\}^n \\ \mathsf{weight}(x) = w_x}} \Pr[Z_k = 1, Z_l = 1 \mid x] \Pr[x] \\
&= \sum_{\substack{x \in \{0,1\}^n \\ \mathsf{weight}(x) = w_x}} \sum_{b \in \{0,1\}} \Pr[x] \Pr\left[\bigoplus_{i \in S_k \cap S_l} y_i = b, \bigoplus_{i \in S_k \setminus S_l} y_i = 1-b, \bigoplus_{i \in S_l \setminus S_k} y_i = 1-b\right] \\
&= \sum_{s=0}^{w_x} \sum_{b \in \{0,1\}} \Pr[|S_k \cap S_l| = s] \Pr\left[\bigoplus_{j=1}^{s} y_{i_j} = b, \bigoplus_{j=1}^{w_x - s} y_{k_j} = 1-b, \bigoplus_{j=1}^{w_x - s} y_{l_j} = 1-b\right].
\end{aligned}$$

The transition from the second to the third line is justified because grouping all x's with the same intersection size s yields the PMF $\Pr[|S_k \cap S_l| = s]$, which implicitly accounts for the factor $\Pr[x]$. Furthermore, the inner probability over the y_i's depends only on the cardinalities of the three disjoint sets, namely s, $w_x - s$, and $w_x - s$, since the y_i's are identically distributed. Thus, we may replace the specific sets with generic sums of s, $w_x - s$, and $w_x - s$ summands.

The Binomial Model. Unlike the uniform slice model, where the size of S_k and S_l is constant, it is now a random variable. Furthermore, we note that $|S_k| = |S_l|$ (both equal to the weight of x). Consequently, we need to compute the joint PMF of the random variables $(|S_k|, |S_k \cap S_l|)$. The joint probability $\Pr[Z_k = 1, Z_l = 1]$ is given by:

$$\begin{aligned}
\Pr[Z_k = 1, Z_l = 1] &= \sum_{x \in \{0,1\}^n} \Pr[Z_k = 1, Z_l = 1 \mid x] \Pr[x] \\
&= \sum_{x \in \{0,1\}^n} \sum_{b \in \{0,1\}} \Pr\left[\bigoplus_{i \in S_k \cap S_l} y_i = b, \bigoplus_{i \in S_k \setminus S_l} y_i = 1-b, \bigoplus_{i \in S_l \setminus S_k} y_i = 1-b\right] \\
&= \sum_{r=0}^{n} \sum_{s=0}^{r} \sum_{b \in \{0,1\}} \Pr[|S_k| = r, |S_k \cap S_l| = s] \cdot \Pr\left[\bigoplus_{j=1}^{s} y_{i_j} = b, \bigoplus_{j=1}^{r-s} y_{k_j} = 1-b, \bigoplus_{j=1}^{r-s} y_{l_j} = 1-b\right].
\end{aligned}$$

The computation of the PMF of $|S_k \cap S_l|$ or the joint PMF of $(|S_k|, |S_k \cap S_l|)$ is sufficiently involved to warrant a separate treatment, which we provide in Section 4 and Appendix 6. For

the remainder of this section, we focus on computing, for both the uniform slice and binomial cases, the following probability

$$\Pr\left[\bigoplus_{j=1}^{n_1} y_{i_j} = b_1, \bigoplus_{j=1}^{n_2} y_{k_j} = b_2, \bigoplus_{j=1}^{n_3} y_{l_j} = b_3\right], \tag{1}$$

where

- n_1, n_2, n_3 are nonnegative integers, and $b_1, b_2, b_3 \in \{0, 1\}$,
- the sets $\{y_{i_j} : j \in [1, n_1]\}$, $\{y_{k_j} : j \in [1, n_2]\}$, and $\{y_{l_j} : j \in [1, n_3]\}$ are pairwise disjoint and are subsets of $\{y_i : i \in [0, n-1]\}$.

This probability will directly enable the computation of the variance $\mathrm{Var}(\mathsf{weight}(x \cdot y))$.

3.1 Uniform Slice Model

Theorem 1
Let y be a uniformly random n-bit vector with fixed weight w_y. The probability defined in Equation (1) is given by:

$$\Pr\left[\bigoplus_{j=1}^{n_1} y_{i_j} = b_1, \bigoplus_{j=1}^{n_2} y_{k_j} = b_2, \bigoplus_{j=1}^{n_3} y_{l_j} = b_3\right] = \frac{1}{\binom{n}{w_y}} \sum_{\substack{p_1,p_2,p_3 \\ p_1+p_2+p_3 \le w_y}} \left(\prod_{i=1}^{3} \mathbf{1}_{p_i \equiv b_i \pmod 2}\right) \cdot \binom{n_1}{p_1}\binom{n_2}{p_2}\binom{n_3}{p_3}\binom{n-n_1-n_2-n_3}{w_y-p_1-p_2-p_3}.$$

Proof. First, note that the size of the probability space is $\binom{n}{w_y}$.

Let us now count the number of sequences $(y_0, \ldots, y_{n-1})$ that satisfy the conditions of the conjunction $\bigoplus_{j=1}^{n_1} y_{i_j} = b_1, \bigoplus_{j=1}^{n_2} y_{k_j} = b_2, \bigoplus_{j=1}^{n_3} y_{l_j} = b_3$.

Assume for simplicity that $b_1 = b_2 = b_3 = 0$. A sequence that satisfies the aforementioned conjunction is constrained by the following Hamming weight distribution: it must contain an even number p_1 of ones amongst n_1 positions, an even number p_2 of ones amongst n_2 positions, an even number p_3 of ones amongst n_3 positions, and finally $w_y - p_1 - p_2 - p_3$ positions must be set to one amongst the remaining $n - n_1 - n_2 - n_3$ positions. The number of such sequences amounts to $\sum_{\substack{\text{even } p_1,p_2,p_3 \\ p_1+p_2+p_3 \le w_y}} \binom{n_1}{p_1}\binom{n_2}{p_2}\binom{n_3}{p_3}\binom{n-n_1-n_2-n_3}{w_y-p_1-p_2-p_3}$.

For arbitrary b_1, b_2, b_3, the considered p_1, p_2, p_3 must have the same parity as b_1, b_2, b_3, respectively. This explains the indicator condition $\prod_{i=1}^{3} \mathbf{1}_{p_i \equiv b_i \pmod 2}$. □

Let $\mathsf{pmf}_U(n, w_x, k, l, s)$ denote the probability $\Pr[|S_k \cap S_l| = s]$, where the probability space is the set of n-bit vectors with weight w_x. Then, computing the variance is straightforward using:

$$\begin{aligned}\mathrm{Var}(Z) &= \sum_{k=0}^{n-1} \mathrm{Cov}(Z_k, Z_k) + 2 \sum_{0 \le k < l < n} \Pr[Z_k = 1, Z_l = 1] - 2 \sum_{0 \le k < l < n} \Pr[Z_k = 1] \Pr[Z_l = 1] \\ &= n \cdot \mathbb{E}[Z_k](1 - \mathbb{E}[Z_k]) + 2 \sum_{0 \le k < l < n} \Pr[Z_k = 1, Z_l = 1] - n(n-1)\mathbb{E}^2[Z_k].\end{aligned}$$

Algorithm 1: Computation of the variance of $\mathsf{weight}(x \cdot y)$ in the uniform slice model

Input: Nonnegative integers n, w_x, w_y satisfying $w_x, w_y \le n$.
Random n-bit vectors $(x_0, \ldots, x_{n-1})$ and $(y_0, \ldots, y_{n-1})$ of weights w_x and w_y, respectively.

Output: $\mathrm{Var}(\mathsf{weight}(x \cdot y))$

$\mathsf{tmp} \leftarrow \sum_{\substack{s=0 \\ s \text{ odd}}}^{\min(w_x, w_y)} \frac{\binom{w_x}{s}\binom{n-w_x}{w_y-s}}{\binom{n}{w_y}}$;

$\mathrm{Cov}_1 \leftarrow \mathsf{tmp}(1 - \mathsf{tmp})$;

$\mathrm{Cov}_2 \leftarrow \frac{1}{\binom{n}{w_y}} \sum_{l=1}^{n-1} \sum_{k=0}^{l-1} \sum_{s=0}^{w_x} \sum_{b=0}^{1} \mathsf{pmf}_U(n, w_x, k, l, s) \times \sum_{p_1+p_2+p_3 \le w_y} \mathbf{1}_{p_1 \equiv b \pmod 2} \mathbf{1}_{p_2 \equiv 1-b \pmod 2} \mathbf{1}_{p_3 \equiv 1-b \pmod 2} \binom{s}{p_1}\binom{w_x-s}{p_2}\binom{w_x-s}{p_3}\binom{n-2w_x+s}{w_y-p_1-p_2-p_3}$;

return $n\mathrm{Cov}_1 + 2\mathrm{Cov}_2 - n(n-1)\mathsf{tmp}^2$;

3.2 Binomial Model

Lemma 3. *Let $y = (y_0, \ldots, y_{n-1})$ be an n-bit vector with Bernoulli(p_y) coordinates. Consider further a bit $b \in \{0, 1\}$ and a nonnegative integer $N \le n$. Then*

$$\Pr\left[\bigoplus_{j=1}^{N} y_{i_j} = b\right] = \begin{cases} \dfrac{1 + (1-2p_y)^N}{2} & \text{if } b = 0, \\ \dfrac{1 - (1-2p_y)^N}{2} & \text{if } b = 1. \end{cases}$$

Proof. Suppose that $b = 0$. Then

$$\begin{aligned} \Pr\left[\bigoplus_{j=1}^{N} y_{i_j} = 0\right] &= \Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 0,\ y_{i_N} = 0\right] + \Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 1,\ y_{i_N} = 1\right] \\ &= (1-p_y)\Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 0 \mid y_{i_N} = 0\right] + p_y \Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 1 \mid y_{i_N} = 1\right] \\ &= (1-p_y)\Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 0\right] + p_y \Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 1\right] \\ &= (1-p_y)\Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 0\right] + p_y \left(1 - \Pr\left[\bigoplus_{j=1}^{N-1} y_{i_j} = 0\right]\right). \end{aligned}$$

If we denote $\Pr[\bigoplus_{j=1}^{N} y_{i_j} = 0] = F(N)$, then F is defined by the recurrence $F(N) = p_y + (1 - 2p_y)F(N-1)$ and the base case $F(0) = 1$. Solving this recurrence for $F(N)$ gives the closed-form expression $\Pr[\bigoplus_{j=1}^{N} y_{i_j} = 0] = (1 + (1-2p_y)^N)/2$.

For $b = 1$, we apply the same approach to define $\Pr[\bigoplus_{j=1}^{N} y_{i_j} = 1] = G(N)$. This function satisfies the same recurrence $G(N) = p_y + (1 - 2p_y)G(N-1)$ as F, but with the base case $G(0) = 0$. The solution to this recurrence yields the desired expression. □

Corollary 1. *Let $y = (y_0, \ldots, y_{n-1})$ be an n-bit vector with Bernoulli(p_y) coordinates. The probability defined in Equation (1) is given by:*

$$\Pr\left[\bigoplus_{j=1}^{n_1} y_{i_j} = b_1, \bigoplus_{j=1}^{n_2} y_{k_j} = b_2, \bigoplus_{j=1}^{n_3} y_{l_j} = b_3\right] = \prod_{i=1}^{3} \frac{1 + (-1)^{b_i}(1-2p_y)^{n_i}}{2}.$$

Proof. Follows directly from Lemma 3 and the independence of the y_i's. □

The algorithm for computing the variance $\mathrm{Var}(\mathsf{weight}(x \cdot y))$ is straightforward, given the probability $\mathsf{pmf}_B(n, p_x, k, l, r, s) = \Pr[|S_k| = r, |S_k \cap S_l| = s]$, defined over the space of n-bit vectors with iid Bernoulli(p_x) coordinates.

Algorithm 2: Computation of the variance of $\mathsf{weight}(x \cdot y)$ in the binomial model

Input: Positive integer n and real numbers $p_x, p_y \in [0, 1]$.
Random n-bit vectors $(x_0, \ldots, x_{n-1}) \sim \mathsf{Binomial}(n, p_x)$ and $(y_0, \ldots, y_{n-1}) \sim \mathsf{Binomial}(n, p_y)$.

Output: $\mathrm{Var}(\mathsf{weight}(x \cdot y))$

$\mathsf{tmp} \leftarrow \frac{1-(1-2p_xp_y)^n}{2}$;
$\mathrm{Cov}_1 \leftarrow \mathsf{tmp}(1 - \mathsf{tmp})$;
$\mathrm{Cov}_2 \leftarrow \frac{1}{8} \sum_{l=1}^{n-1} \sum_{k=0}^{l-1} \sum_{r=0}^{n} \sum_{s=0}^{r} \sum_{b=0}^{1} \mathsf{pmf}_B(n, p_x, k, l, r, s) \times$;
$\left[1 + (-1)^b(1-2p_y)^s\right] \left[1 + (-1)^{1-b}(1-2p_y)^{r-s}\right]^2$;
return $n\mathrm{Cov}_1 + 2\mathrm{Cov}_2 - n(n-1)\mathsf{tmp}^2$;

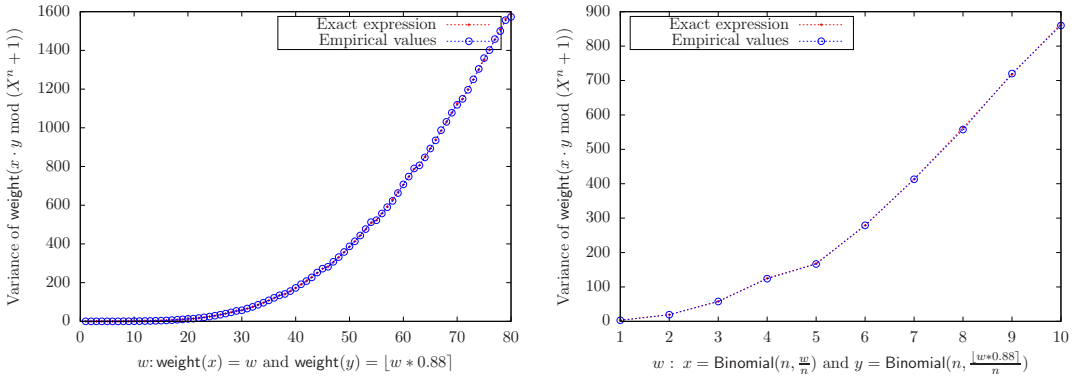
3.3 Validation and Comparative Analysis

For our experiments, we consider the following distributions:

- **Uniform slice**: Both polynomials are sampled uniformly from the set of polynomials with degrees less than $n = 17669$ and respective weights w_x and w_y less than 80 satisfying $w_y = \lfloor w_x \cdot 0.88 \rfloor$; this reflects the real parameters of the HQC scheme in the uniform setting.
- **Binomial**: Both polynomials have degrees less than $n = 503$ and coefficients drawn independently from a Bernoulli distribution with respective parameters $\frac{w_x}{n}$ and $\frac{w_y}{n}$, satisfying $w_y = \lfloor w_x \cdot 0.88 \rfloor$ and $w_x \leq 10$. We resorted to smaller parameters here (while retaining the same ratios) because of the cubic growth of the PMF tables (in n) in that setting (see Remark 4), which renders full-scale runs infeasible on our computer.

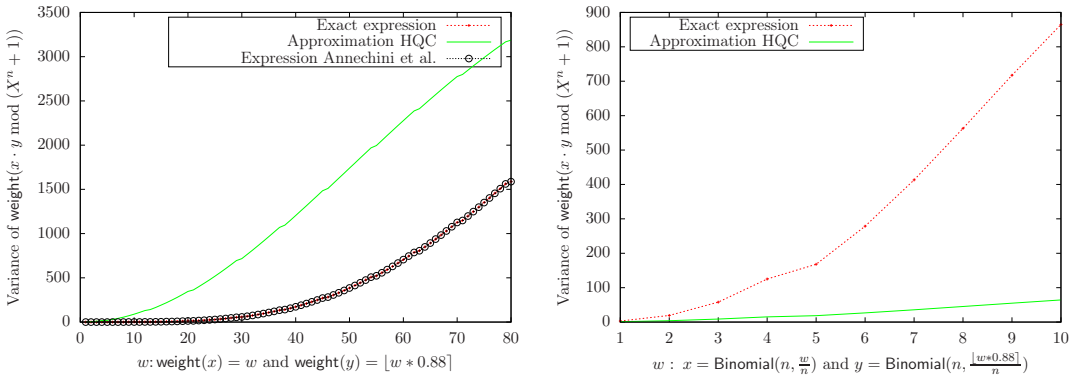
Experimental Validation. We experimentally validate the derived variance expressions through 100,000 trials for the above distributions. In each trial, we compute the product modulo $X^n + 1$ and record the weight Z and its square Z^2. The empirical variance is $\langle Z^2 \rangle - \langle Z \rangle^2$, where $\langle \cdot \rangle$ denotes the sample mean over all trials. As illustrated in Figure 2, the perfect match between the empirical and theoretical variance values graphically corroborates our derived formulae.

Comparison with Existing Approximations. We compare our analytical variance expressions against the HQC approximation [7,10] for both the uniform and binomial settings, using the parameter sets described above. Recall that the HQC approximation assumes the coefficients of the product $x \cdot y$ are Bernoulli distributed. We observe that this approximation overestimates the variance in the uniform setting but underestimates it in the binomial setting. We further validate our uniform-setting variance expression against the PMF-based computation from [2]; the two results coincide exactly, confirming the correctness of our derivation.



(a) Uniform slice case (b) Binomial case

Fig. 2: Variance of the weight of the polynomial product modulo $X^n + 1$



(a) Comparison with the approximation from [7] (b) Comparison with the bound from [10]

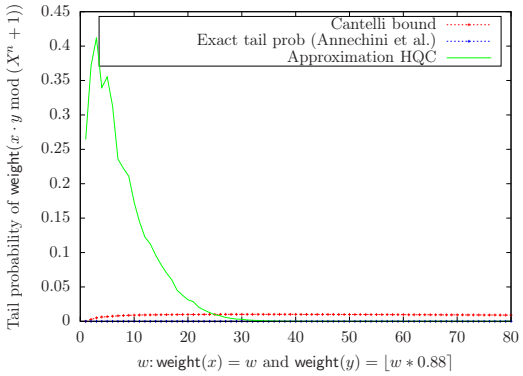
Fig. 3: Comparison of the variance with the HQC approximation [7,10]

Tail Probability of $\mathsf{weight}(x \cdot y)$. We compute an upper bound on the tail probability $\Pr[\mathsf{weight}(x \cdot y) > 1.1 \cdot \mathbb{E}[\mathsf{weight}(x \cdot y)]]$ for the parameter sets described above using Cantelli's bound with our derived variance. We further compare this bound against several alternatives. In the uniform setting, we compare against the tail probability obtained from the HQC approximation under the Bernoulli assumption and against the PMF-based tail probability from [2]. In the binomial setting, we compare against the bound derived from the HQC approximation, the bound derived from Kawachi's approximation [8], and the empirical tail probability obtained from 100,000 trials.

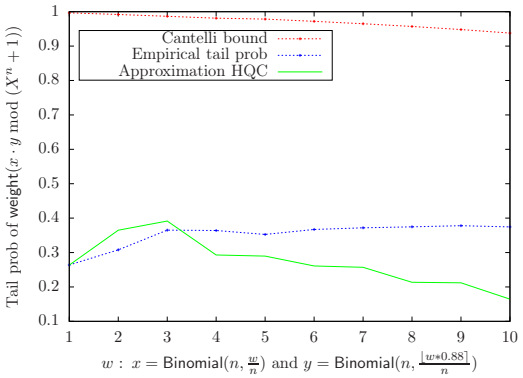
It is clear that Cantelli's bound provides a tight upper bound on the tail probability for the uniform setting, but not for the binomial one. The HQC approximation yields reasonable bounds for both settings. The approximation from [8] gives completely irrelevant bounds for our parameters. The author provides an improved bound in [9], but it cannot be applied to our setting because it assumes $x, y \sim \mathsf{Binomial}(n, p)$, which does not hold here.

4 Probability Law of $|S_k \cap S_l|$ in the Uniform Slice Model

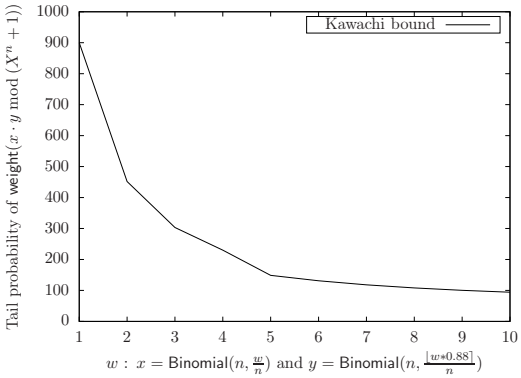
Let n be a positive integer. Let further w_x, k, l be integers in $[0, n]$ with $k < l < n$. In this section, we will determine the probability mass function of the random variable $|S_k \cap S_l|$ across the probability space of w_x-weighted n-bit vectors $x = (x_0, \ldots, x_{n-1})$, where $S_j = \{(j - i) \bmod n : x_i = 1\}$, $j = k, l$.



(a) Uniform setting



(b) Binomial setting



(c) Bound via Kawachi [8]'s approximation in the binomial setting

Fig. 4: Upper bound on the tail probability $\Pr[\mathsf{weight}(x \cdot y) > 1.1 \cdot \mathbb{E}[\mathsf{weight}(x \cdot y)]]$

$$
\begin{aligned}
e \in S_k \cap S_l &\iff \exists i,j \in \mathbb{Z}_n \colon x_i = x_j = 1 \text{ and } e = (k-i) \bmod n \text{ and } e = (l-j) \bmod n \\
&\iff \exists i,j \in \mathbb{Z}_n \colon x_i = x_j = 1 \text{ and } \{k-i = l-j \text{ or } k-i+n = l-j \text{ or } k-i = l-j+n\} \\
&\iff \exists i,j \in \mathbb{Z}_n \colon x_i = x_j = 1 \text{ and } \{k-i = l-j \text{ or } k-i+n = l-j\} \\
&\iff \exists i,j \in \mathbb{Z}_n \colon x_i = x_j = 1 \text{ and } \{j-i = l-k \text{ or } i-j = n+k-l\}
\end{aligned}
$$

The penultimate equivalence follows from $i, j \in \mathbb{Z}_n$ and $k < l$; thus $j - i$ cannot equal $n + l - k$.

We conclude that if $|S_k \cap S_l|$ equals a given value $s \in [0, w_x]$, then there exist exactly s pairs (i, j) among the following pairs:

$$\{(0, l-k), \ldots, (n+k-l-1, n-1)\} \cup \{(n+k-l, 0) \ldots (n-1, l-k-1)\}$$

that satisfy $x_i = x_j = 1$. In fact, any pair (i, j) with $0 \le i, j \le n-1$ that leads to a common element of S_k and S_l must satisfy $x_i = x_j = 1$ in addition to $j - i = l - k$ or $i - j = n + k - l$. We need to distinguish two cases: (1) the case '$l - k = n + k - l$', and (2) the case '$l - k \neq n + k - l$'.

4.1 The Case '$l - k = n + k - l$'

Theorem 2

Given a random n-bit vector $x \in \{0,1\}^n$ of weight w_x, and nonnegative integers k and l satisfying $k < l < n$ and $l - k = n + k - l$, the probability mass function of the random variable $|S_k \cap S_l|$

is given by

$$\Pr[|S_k \cap S_l| = s] = \begin{cases} 0 & \text{if } s \bmod 2 = 1, \\ \frac{1}{\binom{n}{w_x}} \sum_{w=0}^{w_x} \binom{l-k}{s/2} \binom{l-k-s/2}{w-s/2} \binom{l-k-w}{w_x-w-s/2} & \text{otherwise.} \end{cases} \quad (2)$$

Proof. Recall that computing the PMF of $|S_k \cap S_l|$ amounts to computing the probability that there exist exactly s pairs (i, j) among the possible pairs

$$\{(0, l-k), \ldots, (l-k-1, n-1)\} \cup \{(l-k, 0) \ldots (n-1, l-k-1)\}$$

with $x_i = x_j = 1$, where the probability space is the set of random n-bit vectors $x \in \{0,1\}^n$ with weight w_x. Thus, the size of the probability space is $\binom{n}{w_x}$.

First note that if (i, j) is a possible pair, i.e., a pair leading to an element in $S_k \cap S_l$, then $x_i = x_j = 1$ and $|i - j| = l - k = n + k - l$. It follows that $|j - i| = l - k = n + k - l$, leading to (j, i) being also a possible pair. Therefore, if s is odd, then the probability must be zero.

Suppose now that s is even. We need to specify the weight composition of $x = (x_0, \ldots, x_{n-1})$ that gives exactly $s/2$ pairs (i, j) satisfying: (i) $i \in [0, l-k-1]$, (ii) $j = i + l - k$, and (iii) $x_i = x_j = 1$. In fact, this will give a total of s pairs (I, J) with $I, J \in [0, n-1]$, $|I - J| = l - k$, and $x_I = x_J = 1$.

Suppose that the weight of the subvector $(x_0, \ldots, x_{l-k-1})$ is w. Then we set $s/2$ coordinates in $[0, l-k-1]$ to 1, then choose $w - s/2$ among the remaining $l - k - s/2$ coordinates and set them to 1.

It remains now to describe the weight composition of the second subvector $(x_{l-k}, \ldots, x_{n-1})$. Let $i_1, \ldots, i_w$ be the indices of the coordinates set to 1 in the subvector $(x_0, \ldots, x_{l-k-1})$. Recall that the indices j in the pairs (i, j) are constrained by the rule $j = l - k + i$. Thus, the coordinates in $(x_{l-k}, \ldots, x_{n-1})$ whose indices are $i_1 + l - k, \ldots, i_w + l - k$ will be assigned as follows: $s/2$ coordinates (corresponding to the first chosen $s/2$ indices) are set to 1 and the remaining $w - s/2$ coordinates are set to 0. In this way, we get exactly $s/2$ pairs (i, j) satisfying the above three requirements. It remains now to complete setting the coordinates of the subvector $(x_{l-k}, \ldots, x_{n-1})$ so as to obtain the partial weight $w_x - w$. To do so, we need to discard w indices from $[l-k, n-1]$, namely the $i_1 + l - k, \ldots, i_w + l - k$ indices, then choose $w_x - w - s/2$ indices and set the corresponding coordinates to 1. □

4.2 The Case '$l - k \neq n + k - l$'

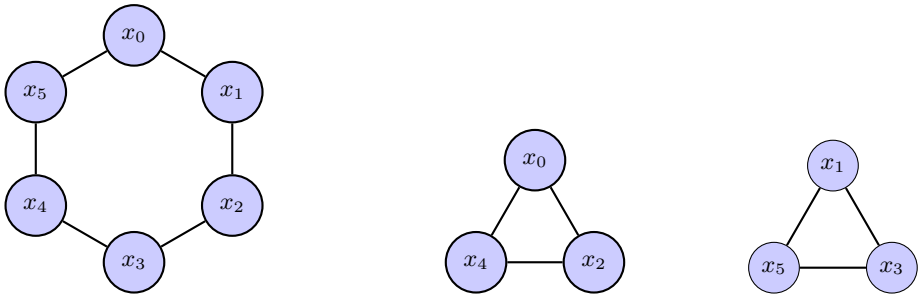
In this case, we give a graphic representation to our problem.

Definition 1 (Graph G). *Let $(x_0, \ldots, x_{n-1}) \in \{0,1\}^n$ be a random n-bit vector with weight w_x. We define G to be the undirected graph on the vertices $\{x_0, \ldots, x_{n-1}\}$ where adjacency is given by the rule: vertex x_i is adjacent to vertex x_j if and only if $(i-j) \bmod n \in \{l-k, n+k-l\}$, which is equivalent to $(i - j) \equiv l - k \pmod{n}$ or $(i - j) \equiv k - l \pmod{n}$.*

Note that according to our definition of adjacency, x_i is adjacent to x_j if and only if x_j is adjacent to x_i. Thus, this definition is consistent with the usual requirement for an undirected graph.

We can then reduce the problem of computing the probability that $|S_k \cap S_l|$ takes a given value s to computing the probability that the graph G contains exactly s edges incident to two vertices both assigned the value 1.

At this stage, we state and prove the following result. The shape of the graph G depends on the value of $\gcd(l - k, n)$. More precisely, if $\gcd(n, l - k) = 1$, then G is a cycle graph, and if $\gcd(l - k, n) = d$ with $d > 1$, then G is a disjoint union of d cycle subgraphs. Figures 5a and 5b exemplify this result for $n = 6$.



(a) $(l,k) = (5,0)$ $[\gcd(l-k,n) = 1]$ (b) $(l,k) = (5,1)$ $[\gcd(l-k,n) = 2]$

Fig. 5: **Graph G for $n = 6$ and different values of (l,k)**

Theorem 3
If $\gcd(l-k,n) = 1$*, then G is a cycle graph.*

Proof. We consider the cycle graph H on the vertices $x_{(i(l-k)) \bmod n}$ for $i = 0,\dots,n-1$, where vertex $x_{(i(l-k)) \bmod n}$ is adjacent to vertex $x_{((i+1)(l-k)) \bmod n}$ for $i = 0,\dots,n-2$, and vertex $x_{((n-1)(l-k)) \bmod n}$ is adjacent to vertex x_0.

Since $\gcd(l-k,n) = 1$, $l-k$ is a generator of $(\mathbb{Z}_n,+)$. It follows that the values $(i(l-k)) \bmod n$ span the additive group $\mathbb{Z}_n$. Thus $V(H) = V(G)$.

Moreover, an edge $(x_{(i(l-k)) \bmod n}, x_{((i+1)(l-k)) \bmod n})$, for $i = 0,\dots,n-2$, satisfies $((i+1)(l-k)-i(l-k)) \bmod n = l-k$. Also, the edge $(x_{((n-1)(l-k)) \bmod n}, x_0)$ satisfies $((n-1)(l-k)) \bmod n = n+k-l$. Thus $E(H) \subseteq E(G)$.
Now let (x_i,x_j) be an edge in G. Then $(i-j) \bmod n \in \{l-k, n+k-l\}$. Since $l-k$ is a generator of $\mathbb{Z}_n$, there exists $s \in \mathbb{Z}_n$ such that $i \equiv s(l-k) \pmod n$. It follows that $s(l-k) \equiv l-k+j \pmod n$ or $s(l-k) \equiv k-l+j \pmod n$. Thus $j \equiv (s-1)(l-k) \pmod n$ or $j \equiv (s+1)(l-k) \pmod n$, which implies $(x_i,x_j) \in E(H)$.
We conclude that $E(G) = E(H)$, and thus $G = H$. □

Theorem 4
If $\gcd(l-k,n) = d$*, then G is a disjoint union of cycle graphs, each of length n/d.*

Proof. Let H_j, for $0 \le j \le d-1$, be the cycle graph on the vertices $x_{(j+i(l-k)) \bmod n}$ for $0 \le i \le n/d-1$. Concerning the edges of H_j, we build them as follows. Vertex $x_{(j+i(l-k)) \bmod n}$ is adjacent to vertex $x_{(j+(i+1)(l-k)) \bmod n}$ for $0 \le i \le n/d-2$, and vertex $x_{(j+(n/d-1)(l-k)) \bmod n}$ is adjacent to vertex $x_{j \bmod n}$.

We will show that G is the disjoint union of the graphs H_j.

First, note that n/d is the order of the subgroup generated by $(l-k) \bmod n$; thus H_j has exactly n/d vertices. Let us now prove that the H_j are pairwise disjoint. Suppose there is a common vertex in H_j and $H_{j'}$. Then there exist $i,i' \in [0,n/d-1]$ such that $(j+i(l-k)) \bmod n = (j'+i'(l-k)) \bmod n$. Thus $j-j'-(i'-i)(l-k) \equiv 0 \pmod n$. This implies that $j-j'$ is a multiple of d (both n and $l-k$ are multiples of d), which implies $j = j'$ as $0 \le j,j' \le d-1$.
Thus the H_j are pairwise disjoint, and their union $\bigcup_j H_j$ has cardinality $d \cdot n/d = n$. Since $V(\bigcup_j H_j) \subseteq V(G)$ and G has n vertices, we conclude that $V(G) = V(\bigcup_j H_j)$.

Moreover, let $(x_r,x_{r'})$ be an edge in G. Then $r'-r \equiv l-k \pmod n$ or $r'-r \equiv k-l \pmod n$. Moreover, there exist j,i with $0 \le j \le d-1$ and $0 \le i \le n/d-1$ such that $x_r \in H_j$ and $r \equiv j+i(l-k) \pmod n$. Thus $r' \equiv j+i(l-k)+(l-k) \pmod n$ or $r' \equiv j+i(l-k)+(k-l) \pmod n$. It follows that $r' \equiv j+(i+1)(l-k) \pmod n$ or $r' \equiv j+(i-1)(l-k) \pmod n$, which means $(x_r,x_{r'})$ is an edge of H_j, and $E(G) \subseteq E(\bigcup_j H_j)$.

Now let $(x_r, x_{r'}) \in E(\bigcup_j H_j)$. Since the H_j's are pairwise disjoint, there exists $j \in [0, d-1]$ such that $(x_r, x_{r'}) \in H_j$. It follows that there exists $i \in [0, n/d-1]$ such that $r \equiv j + i(l-k) \pmod n$ and $r' \equiv j + (i+1)(l-k) \pmod n$. This implies $r' - r \equiv l - k \pmod n$, which means $(x_r, x_{r'}) \in E(G)$. Therefore, $E(\bigcup_j H_j) = E(G)$.

We conclude that $G = \bigcup_j H_j$. □

Recall that the goal of this subsection is to determine the probability that the graph G contains exactly a given number, say s, of edges whose endpoints are both set to 1. Since we have demonstrated that G is either a cycle graph or a disjoint union of cycle graphs, we first compute this probability for a cycle graph and then expand the analysis to treat the more general case of a disjoint union of cycle graphs.

4.2.1 Cycle Graphs

Throughout this subsection, i, j, s will denote nonnegative integers, satisfying $0 \leq i \leq n$ and $0 \leq j, s \leq w_x$.

Definition 2 (Cycle-PMF). *For $i > 0$, consider a random i-bit vector $(x_0, \ldots, x_{i-1}) \in \{0,1\}^i$ with fixed weight j.*
$\mathsf{cycle\text{-}pmf}(i, j, s)$ *denotes the probability that the cycle graph $x_0 - \ldots - x_{i-1} - x_0$ contains exactly s edges with both endpoints equal to 1. The probability is taken over all random i-bit vectors $(x_0, \ldots, x_{i-1})$ with weight j.*

To compute $\mathsf{cycle\text{-}pmf}(i, j, s)$, we assign a value to one vertex in the cycle graph $x_0 - \ldots - x_{i-1} - x_0$, say x_0. It then suffices to compute the probability that the linear chain $x_1 - \ldots - x_{i-1}$ contains exactly s or $s-1$ edges with both endpoints equal to 1. We now introduce some auxiliary functions that compute the aforementioned probabilities.

Definition 3 (Functions f_0, f_1, g, f). *Let i, j be nonnegative integers. Furthermore, let $x_0 - \ldots - x_i$ be a linear chain, where $(x_0, \ldots, x_i)$ is an $(i+1)$-bit random vector of weight j.*

- $f_0(i, j, s)$ *denotes the conditional probability that the linear chain $x_0 - \ldots - x_i$ contains exactly s edges with both endpoints equal to* 1, given that *x_0 is set to* 0.
- $f_1(i, j, s)$ *denotes the conditional probability that $x_0 - \ldots - x_i$ contains exactly s edges with both endpoints equal to* 1, given that *x_0 is set to* 1.
- $g(i, j, s, b, c)$ *, for $b, c \in \{0, 1\}$, denotes the conditional probability that $x_0 - \ldots - x_i$ contains exactly s edges with both endpoints equal to* 1, given that *x_0 and x_i are set to b and c, respectively.*
- $f(i, j, s)$ *denotes the probability that $x_0 - \ldots - x_i$ contains exactly s edges with both endpoints equal to* 1.

Let us now proceed to the computation of the above functions. Concerning the functions f_0, f_1, and g, we employ a dynamic programming approach for their computation. This involves first evaluating the functions at their base cases (initial argument values), then providing a recurrence relation that allows to iteratively evaluate the functions at all further values.

Form. 1. and **Form. 2.** give respectively the formulae for computing f_0, f_1, and g; the base cases are immediate, whilst the recurrence relations follow from:

$$f_0(i, j, s) = \Pr[x_1 = 0 \mid x_0 = 0] f_0(i-1, j, s) + \Pr[x_1 = 1 \mid x_0 = 0] f_1(i-1, j, s)$$
$$f_1(i, j, s) = \Pr[x_1 = 0 \mid x_0 = 1] f_0(i-1, j-1, s) + \Pr[x_1 = 1 \mid x_0 = 1] f_1(i-1, j-1, s-1)$$
$$g(i, j, s, b, c) = \Pr[x_1 = 0 \mid x_0 = b, x_i = c] g(i-1, j-b, s, 0, c) + \Pr[x_1 = 1 \mid x_0 = b, x_i = c] g(i-1, j-b, s-b, 1, c)$$

Where

$$\Pr[x_1 = 0 \mid x_0 = 0] = \frac{\binom{i-1}{j}}{\binom{i}{j}} \quad \text{and} \quad \Pr[x_1 = 1 \mid x_0 = 0] = \frac{\binom{i-1}{j-1}}{\binom{i}{j}}$$

$$\Pr[x_1 = 0 \mid x_0 = 1] = \frac{\binom{i-1}{j-1}}{\binom{i}{j-1}} \quad \text{and} \quad \Pr[x_1 = 1 \mid x_0 = 1] = \frac{\binom{i-1}{j-2}}{\binom{i}{j-1}}$$

$$\Pr[x_1 = 0 \mid x_0 = b, x_i = c] = \frac{\binom{i-2}{j-b-c}}{\binom{i-1}{j-b-c}} \quad \text{and} \quad \Pr[x_1 = 1 \mid x_0 = b, x_i = c] = \frac{\binom{i-2}{j-b-c-1}}{\binom{i-1}{j-b-c}}$$

$f_0(i,j,s) = f_1(i,j,s) = 0$ if $(j > i+1) \vee (s > i) \vee (s > j-1) \vee (s < 0)$. Otherwise:

$$f_0(i,j,s) = \begin{cases} (j=0)\cdot(s=0) & \text{if } i = 0 \\ (j \le i)\cdot(s=0) & \text{if } i = 1 \\ \begin{cases} 0 & \text{if } \binom{i}{j} = 0 \\ \frac{\binom{i-1}{j}}{\binom{i}{j}} f_0(i-1,j,s) + \frac{\binom{i-1}{j-1}}{\binom{i}{j}} f_1(i-1,j,s) & \text{otherwise} \end{cases} & \text{otherwise} \end{cases}$$

$$f_1(i,j,s) = \begin{cases} (j=1)\cdot(s=0) & \text{if } i = 0 \\ (j=1)\cdot(s=0) + (j=2)\cdot(s=1) & \text{if } i = 1 \\ \begin{cases} 0 & \text{if } \binom{i}{j-1} = 0 \\ \frac{\binom{i-1}{j-1}}{\binom{i}{j-1}} f_0(i-1,j-1,s) + \frac{\binom{i-1}{j-2}}{\binom{i}{j-1}} f_1(i-1,j-1,s-1) & \text{otherwise} \end{cases} & \text{otherwise} \end{cases}$$

Form. 1. Computation of f_0 and f_1

$g(i,j,s,b,c) = 0$ if $(j > i+1) \vee (s > i) \vee (j > b+c) \vee (s < 0) \vee (i = 0)$. Otherwise:

$$g(i,j,s,b,c) = \begin{cases} (j = b+c)\cdot(s = b\cdot c) & \text{if } i = 1 \\ \begin{cases} 0 & \text{if } \binom{i-1}{j-b-c} = 0 \\ \frac{\binom{i-2}{j-b-c}}{\binom{i-1}{j-b-c}} g(i-1,j-b,s,0,c) + \frac{\binom{i-2}{j-b-c-1}}{\binom{i-1}{j-b-c}} g(i-1,j-b,s-b,1,c) & \text{otherwise} \end{cases} & \text{otherwise} \end{cases}$$

Form. 2. Computation of g

Form. 3. is justified by

$$f(i,j,s) = Pr[x_0 = 0] f_0(i,j,s) + Pr[x_0 = 1] f_1(i,j,s)$$

where $Pr[x_0 = 0]$ and $Pr[x_0 = 1]$ are given by $\frac{\binom{i}{j}}{\binom{i+1}{j}}$ and $\frac{\binom{i}{j-1}}{\binom{i+1}{j}}$ respectively.

$$f(i,j,s) = \begin{cases} 0 & \text{if } \binom{i+1}{j} = 0 \\ \frac{\binom{i}{j}}{\binom{i+1}{j}} f_0(i,j,s) + \frac{\binom{i}{j-1}}{\binom{i+1}{j}} f_1(i,j,s) & \text{otherwise} \end{cases}$$

Form. 3. Computation of f

Finally, for cycle-pmf given in **Form. 4.** , it is clear that if $i \le 2$, then cycle-pmf$(i,j,s) = f(i-1,j,s)$, otherwise:

$$\mathsf{cycle\text{-}pmf}(i,j,s) = \Pr[x_0=0] f(i-2,j,s) + \sum_{b,c\in\{0,1\}} \Pr[x_0=1, x_1=b, x_{i-1}=c] \\ g(i-2,j-1,s-b-c,b,c)$$

where $\Pr[x_0=0]$ in the case of the cycle graph considered in cycle-pmf, [1] is given by $\frac{\binom{i-1}{j}}{\binom{i}{j}}$, whilst $\Pr[x_0=1,x_1=b,x_{i-1}=c]$ is given by $\frac{\binom{i-3}{j-1-b-c}}{\binom{i}{j}}$.

$$\mathsf{cycle\text{-}pmf}(i,j,s) = \begin{cases} 0 & \text{if } \binom{i}{j} = 0 \\ f(i-1,j,s) & \text{if } i < 3 \\ \frac{\binom{i-1}{j}}{\binom{i}{j}} f(i-2,j,s) + \sum_{b,c\in\{0,1\}} \frac{\binom{i-3}{j-1-b-c}}{\binom{i}{j}} g(i-2,j-1,s-b-c,b,c) & \text{otherwise} \end{cases}$$

Form. 4. Computation of cycle-pmf

Implementation Consideration We precompute all the binomial coefficients $\binom{i}{j}$ for $0 \le i \le n$ and $0 \le j \le w_x$. Furthermore, we employ a dynamic programming approach to precompute $f_0(i,j,s)$, $f_1(i,j,s)$, and $g(i,j,s,b,c)$ for all nonnegative integers i,j,s with $0 \le j,s \le w_x$ and $0 \le i \le n$, and for all bits b,c. Following the formulae in **Form. 1.** and **Form. 2.** , we fill the tables dp_{f_0}, dp_{f_1}, and dp_g, respectively, in a bottom-up manner. This precomputation step, which requires $O(n \cdot w_x^2)$ time and space, allows for constant-time computation of $\mathsf{cycle\text{-}pmf}(i,j,s)$, as the result is a simple combination of the stored values.

4.2.2 Disjoint Union of Cycle Graphs

Suppose now that graph G is a disjoint union of d cycle graphs $G = \bigcup_{i=0,\ldots,d-1} G_i$, where $d > 1$ (the case $d = 1$ has been treated in the previous subsubsection). The vertices of G are again the coordinates of an n-bit random vector $(x_0,\ldots,x_{n-1})$ with weight w_x.
Let S denote the total number of edges in G that are incident to two vertices both equal to 1. Our goal is to determine the PMF of S.

Let S_i denote the number of edges in G_i, for $0 \le i \le d-1$, that are incident to two vertices both equal to 1. Furthermore, let $\mathsf{weight}(G_i)$ denote the number of vertices in G_i that are equal to 1. Then

$$\Pr[S=s] = \sum_{\substack{w_0,\ldots,w_{d-1} \\ \sum w_i = w_x}} \sum_{\substack{s_0,\ldots,s_{d-1} \\ \sum s_i = s}} \Pr[\bigwedge_{i=0}^{d-1} \mathsf{weight}(G_i) = w_i] \cdot \Pr[\bigwedge_{i=0}^{d-1} S_i = s_i \mid \bigwedge_{i=0}^{d-1} \mathsf{weight}(G_i) = w_i]$$

We know from Theorem 4 that the graphs G_i contain n/d vertices each. It follows that $\Pr[\bigwedge_{i=0}^{d-1} \mathsf{weight}(G_i) = w_i] = \frac{1}{\binom{n}{w_x}} \prod_{i=0}^{d-1} \binom{n/d}{w_i}$. On the other hand, since the graphs G_i are pairwise disjoint, the conditional distributions $S_0|\mathsf{weight}(G_0),\ldots,S_{d-1}|\mathsf{weight}(G_{d-1})$ are mutually independent:

[1] The underlying probability space should be inferred from the context. Specifically, the quantity $\Pr[x_0=0]$ in the case of the cycle graph considered for cycle-pmf is given by $\frac{\binom{i-1}{j}}{\binom{i}{j}}$, whereas the same quantity, when considered for the linear chain underlying f_0, f_1, f, or g, is $\frac{\binom{i}{j}}{\binom{i+1}{j}}$.

$$\Pr[\bigwedge_{i=0}^{d-1} S_i = s_i \mid \bigwedge_{i=0}^{d-1} \mathsf{weight}(G_i) = w_i] = \prod_{i=0}^{d-1} \Pr[S_i = s_i | \mathsf{weight}(G_i) = w_i]$$
$$= \prod_{i=0}^{d-1} \mathsf{cycle\text{-}pmf}(n/d, w_i, s_i)$$

We obtain the following expression:

$$\Pr[S = s] = \frac{1}{\binom{n}{w_x}} \sum_{\substack{w_0,\ldots,w_{d-1} \\ \sum w_i = w_x}} \sum_{\substack{s_0,\ldots,s_{d-1} \\ \sum s_i = s}} \prod_{i=0}^{d-1} \binom{n/d}{w_i} \mathsf{cycle\text{-}pmf}(n/d, w_i, s_i)$$

Form. 5. Computation of the PMF of S

Implementation Consideration. The probability in **Form. 5.** can be computed in polynomial time in n if we employ a dynamic programming approach, as we did previously for the functions f_0, f_1, and g.
Let D be the largest divisor of n. Consider the following function h:

$$h\colon [0, D] \times [0, w_x] \times [0, w_x] \times [0, D] \longrightarrow \mathbb{R}$$
$$(N, w, s, d) \longmapsto \sum_{\substack{w_0,\ldots,w_{d-1} \\ \sum w_i = w}} \sum_{\substack{s_0,\ldots,s_{d-1} \\ \sum s_i = s}} \prod_{i=0}^{d-1} \binom{N}{w_i} \mathsf{cycle\text{-}pmf}(N, w_i, s_i)$$

Then

$$h(N, w, s, d) = \sum_{w_0=0}^{w} \sum_{s_0=0}^{s} \binom{N}{w_0} \mathsf{cycle\text{-}pmf}(N, w_0, s_0) \sum_{\substack{w_1,\ldots,w_{d-1} \\ \sum w_i = w - w_0}} \sum_{\substack{s_1,\ldots,s_{d-1} \\ \sum s_i = s - s_0}} \prod_{i=1}^{d-1} \binom{N}{w_i} \mathsf{cycle\text{-}pmf}(N, w_i, s_i)$$
$$= \sum_{w_0=0}^{w} \sum_{s_0=0}^{s} \binom{N}{w_0} \mathsf{cycle\text{-}pmf}(N, w_0, s_0) h(N, w - w_0, s - s_0, d - 1) \tag{3}$$

We can then precompute all the values of $h(N, w, s, d)$, for the considered set of parameters, using a dynamic programming table that we fill in a bottom-up manner according to the recurrence relation defined in Equation 3. This allows the computation of the PMF of S in time and space polynomial in n.

4.3 Piecing all Together

4.3.1 Probability Law of $|S_k \cap S_l|$

We are now ready to state the algorithm that computes the PMF of $|S_k \cap S_l|$, where $0 \leq k < l < n$ and $S_j = \{(j - i) \bmod n : x_i = 1\}$ for $j = k, l$; the probability space is the set of n-bit vectors $x = (x_0, \ldots, x_{n-1}) \in \{0, 1\}^n$ of weight w_x.

Algorithm 3: Computation of the PMF of $|S_k \cap S_l|$

Input: Nonnegative integers n, w_x, k, l, and s with $k < l < n$ and $s \le w_x \le n$
Output: $\Pr[|S_k \cap S_l| = s] = \mathsf{pmf}_U(n, w_x, k, l, s)$

if $n + k - l = l - k$ **then**

$$res = \begin{cases} 0 & \text{if } s \bmod 2 = 1 \\ \frac{1}{\binom{n}{w_x}} \sum_{w=0}^{w_x} \binom{l-k}{s/2} \binom{l-k-s/2}{w-s/2} \binom{l-k-w}{w_x-w-s/2} & \text{otherwise} \end{cases}$$

else

// We assume the values $f_0(i,j,t)$, $f_1(i,j,t)$, $g(i,j,t,b,c)$, and $h(N,w,t,p)$,
// for all parameter sets under consideration, are precomputed
// using dynamic programming tables as mentioned earlier

$d = \gcd(l - k, n)$;

if $d = 1$ **then**

$res = \mathsf{cycle\text{-}pmf}(n, w_x, s)$; /* cycle-pmf is computed according to Form. 4. */

else

$res = \frac{1}{\binom{n}{w_x}} h(n/d, w_x, s, d)$;

return res;

Theorem 5
Algorithm 3 computes the PMF of $|S_k \cap S_l|$ in time and space polynomial in n. □

Remark 1 (Batching Benefits for the PMF Computation). Since $\mathsf{pmf}_U(n, w_x, k, l, s)$ is obtained from the precomputed tables f_0, f_1, g, and h, and since the same applies to any $\mathsf{pmf}_U(n, w_x, k, l, s)$ with $w \le w_x$, we gain a significant batching advantage. Building these tables once for w_{max} suffices to compute pmf_U for all $w \le w_{\mathsf{max}}$, eliminating redundant computation.

4.3.2 Optimized Computation of the Variance

Note that the PMF of $|S_k \cap S_l|$, besides depending on n and w_x, is a function solely of $\gcd(l - k, n)$. Therefore, $\Pr[|S_k \cap S_l| = s] = \Pr[|S_{(k+t) \bmod n} \cap S_{(l+t) \bmod n}| = s]$ for all integers $t \in [0, n-1]$ and all $s \in [0, w_x]$. This property enables a significant simplification in the computation of the variance $\mathrm{Var}(\mathsf{weight}(x \cdot y))$. Preserving the previous notation, let $Z = \mathsf{weight}(x \cdot y) = \sum_{k=0}^{n} Z_k$, with $Z_k = \bigoplus_{0 \le i < n} x_i y_{(k-i) \bmod n}$. Then, instead of a nested loop over indices k and l when computing the covariance coefficients, the calculation is reduced to a single loop, as detailed below:

$$\begin{aligned} \sum_{k<l} \Pr[Z_k = 1 \wedge Z_l = 1] &= \sum_{k<l} \sum_{s=0}^{w_w} \sum_{b \in \{0,1\}} \mathsf{pmf}_U(n, w_x, k, l, s) \Pr[\bigoplus_{j=1}^{s} y_{i_j} = b \wedge \bigoplus_{j=1}^{w_x - s} y_{k_j} = 1 - b \wedge \bigoplus_{j=1}^{w_x - s} y_{l_j} = 1 - b] \\ &= \frac{n}{2} \sum_{l=1}^{n} \sum_{s=0}^{w_w} \sum_{b \in \{0,1\}} \mathsf{pmf}_U(n, w_x, 0, l, s) \Pr[\bigoplus_{j=1}^{s} y_{i_j} = b \wedge \bigoplus_{j=1}^{w_x - s} y_{k_j} = 1 - b \wedge \bigoplus_{j=1}^{w_x - s} y_{l_j} = 1 - b] \end{aligned}$$

4.3.3 Case of n Prime

Recall that pmf_U depends on $l - k$ solely through the value of $\gcd(l - k, n)$. When n is prime, which is the case for the HQC scheme, this gcd equals 1 for every nonzero difference. Hence, pmf_U takes the same value for all distinct coordinate pairs, which implies that all pairwise covariance coefficients are equal. Consequently, the entries of the error vector are identically distributed and pairwise equicorrelated—that is, they are first- and second-order exchangeable.

From a coding perspective, this symmetry is advantageous for designing robust codes whose decoding algorithms can actively exploit this invariance. From a cryptographic standpoint, this

property offers a significantly stronger guarantee than merely knowing the overall Hamming weight distribution of the error vector. Indeed, the uniform correlation structure prevents an adversary from biasing the noise toward specific coordinates, as such a bias would break the symmetry that the scheme inherently enforces.

4.4 Experimental Results

We experimentally validate our formula for $\Pr[|S_k \cap S_l| = s]$ over a sample of 100,000 trials. In each trial, we generate a uniformly random polynomial x with degree less than $n = 250$ and fixed weight $w_x = 65$, then record the size $|S_k \cap S_l|$. The empirical value of $\Pr[|S_k \cap S_l| = s]$ is estimated as the proportion of trials in which $|S_k \cap S_l| = s$.

Since $\Pr[|S_k \cap S_l| = s]$, besides depending on n and w_x, is a function solely of $l - k$ and s, we plot this PMF in Figure 6 for the three values: (i) $l - k = 17$, corresponding to the case '$\gcd(l - k, n) = 1$', (ii) $l - k = 25$, corresponding to the case '$\gcd(l - k, n) \neq 1$', and (iii) $l - k = 125$ corresponding to the case '$l - k = n + k - l$'. The perfect concordance between the 'Theory' and 'Experiments' curves graphically verifies our theoretical derivations.

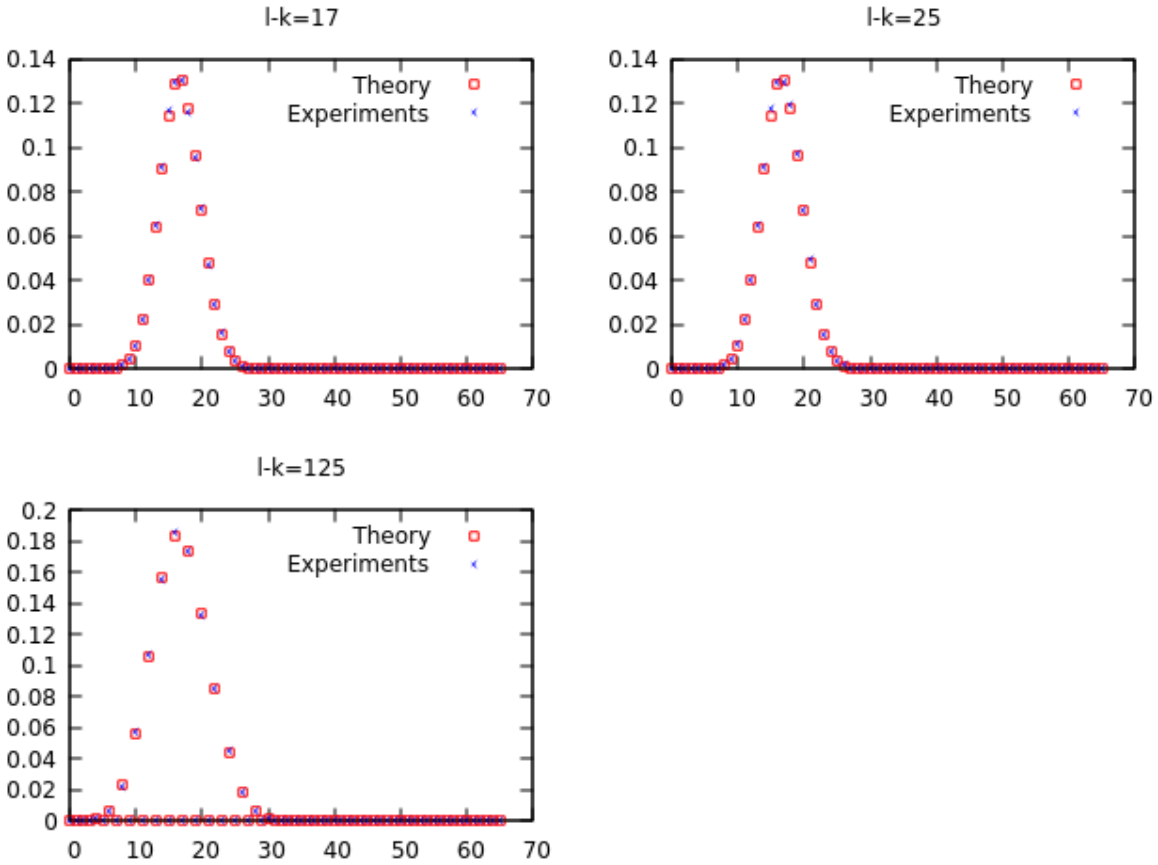


Fig. 6: Probability $\Pr[|S_k \cap S_l| = s]$ for $n = 250$ and $w_x = 65$. The x-axis corresponds to the values $s \in [0, w_x]$

5 Hamming Quasi-Cyclic Codes

In this section, we compute the expectation and variance of the weight of the error vector in the HQC scheme [7]. We derive our results for both the uniform and binomial models.

We adopt the following convention in this section: $\{v_k\}_{k=0}^{n-1}$ refers to the set of coordinates of the vector $v = (v_0, \ldots, v_{n-1})$; specifically, $\{x_k\}_{k=0}^{n-1}$ denotes the set of coordinates of x, and $\{r_{2,k}\}_{k=0}^{n-1}$ refers to the coordinates of r_2, etc. Also, n-bit vectors are identified with polynomials in the ring $\mathbb{F}_2[X]/(X^n+1)$, and thus their arithmetic (addition and multiplication) is performed modulo $X^n + 1$. Finally, $\mathbb{R}_w$ denotes the set of n-bit vectors with weight w.

5.1 Uniform Slice Model

Definition 4 (HQC Error Vector – Uniform Model). *Let n be a nonnegative integer, and let w, w_1, w_2, w_e be nonnegative integers less than or equal to n. The error vector is given by $e' = x \cdot r_2 + y \cdot r_1 + e$, where*

- $x, y \in \mathbb{R}_w$.
- $r_1 \in \mathbb{R}_{w_1}$ *and* $r_2 \in \mathbb{R}_{w_2}$.
- $e \in \mathbb{R}_{w_e}$.

For convenience, we denote the products $x \cdot r_2$ and $r_1 \cdot y$ by X and Y, respectively.

5.1.1 Expectation

The coordinates $\{e'_k\}_{k=0}^{n-1}$ of e' satisfy $e'_k = X_k \oplus Y_k \oplus e_k$. Let $W = \mathsf{weight}(e')$ denote the weight of e', so that $W = \sum_{k=0}^{n-1} e'_k$. By linearity of expectation, the expectation of $\mathsf{weight}(e')$ is given by

$$\begin{aligned}\mathbb{E}(W) &= \sum_{k=0}^{n-1} \mathbb{E}(e'_k) = \sum_{k=0}^{n-1} \Pr[e'_k = 1] \\ &= \sum_{k=0}^{n-1} \sum_{b,c \in \{0,1\}} \Pr[X_k = b] \Pr[e_k = c] \Pr[Y_k = 1 \oplus b \oplus c]\end{aligned}$$

We have

$$\Pr[e_k = c] = \frac{\binom{n-1}{w_e - c}}{\binom{n}{w_e}} = c\frac{w_e}{n} + (1-c)\frac{n - w_e}{n}$$

Let E_X denote the expectation of $\mathsf{weight}(X)$, and let E_Y denote the expectation of $\mathsf{weight}(Y)$. E_X and E_Y are computed according to Lemma 1. It has been established that $E_X = n \Pr[X_k = 1]$ and $E_Y = n \Pr[Y_k = 1]$ for $k = 0, \ldots, n-1$. Therefore:

$$\Pr[X_k = b] = b\frac{E_X}{n} + (1-b)\frac{n - E_X}{n}$$

$$\Pr[Y_k = 1 \oplus b \oplus c] = (1 \oplus b \oplus c)\frac{E_Y}{n} + (b \oplus c)\frac{n - E_Y}{n}$$

Note that $\Pr[e'_k = 1]$ does not depend on k. Thus $\mathbb{E}(W) = n \Pr[e'_k = 1]$. We therefore derive the following theorem.

Theorem 6 (Expectation)
The expectation of the error vector in the HQC scheme, as defined in Definition 4, is given by

$$\mathbb{E}(\mathsf{weight}(e')) = \frac{E_Y}{n}(n - w_e)(1 - \frac{E_X}{n}) + \frac{E_X}{n}(n - w_e)(1 - \frac{E_Y}{n}) + w_e(1 - \frac{E_X}{n})(1 - \frac{E_Y}{n}) + w_e\frac{E_X E_Y}{n^2}$$

where E_X and E_Y denote the expectations of $\mathsf{weight}(X)$ *and* $\mathsf{weight}(Y)$*, respectively, computed via Lemma 1.* □

5.1.2 Variance

Let E denote the expectation of $\mathsf{weight}(e')$. We compute the variance of $\mathsf{weight}(e')$ by determining the covariances $\mathrm{Cov}(e'_k, e'_l)$:

$$\begin{aligned}
\mathrm{Var}(\mathsf{weight}(e')) &= \sum_{0\le k,l<n} \mathrm{Cov}(e'_k, e'_l) \\
&= \sum_{k=0}^{n-1} \mathrm{Cov}(e'_k, e'_k) + 2 \sum_{0\le k<l<n} \mathrm{Cov}(e'_k, e'_l) \\
&= n \Pr[e'_k = 1](1 - \Pr[e'_k = 1]) - n(n-1)(\Pr[e'_k = 1])^2 + 2 \sum_{0\le k<l<n} \Pr[e'_k = 1, e'_l = 1] \\
&= E - E^2 + 2 \sum_{0\le k<l<n} \Pr[e'_k = 1, e'_l = 1]
\end{aligned}$$

Let us now determine the joint probability $\Pr[e'_k = 1, e'_l = 1]$.

$$\begin{aligned}
\Pr[e'_k = 1, e'_l = 1] &= \Pr[X_k \oplus Y_k \oplus e_k = 1, X_l \oplus Y_l \oplus e_l = 1] \\
&= \sum_{b,b',c,c' \in \{0,1\}} \Pr[X_k = b, Y_k = c, e_k = 1 \oplus b \oplus c, X_l = b', Y_l = c', e_l = 1 \oplus b' \oplus c'] \\
&= \sum_{b,b',c,c' \in \{0,1\}} \Pr[(X_k = b, X_l = b'), (Y_k = c, Y_l = c'), (e_k = 1 \oplus b \oplus c, e_l = 1 \oplus b' \oplus c')] \\
&= \sum_{b,b',c,c' \in \{0,1\}} \Pr[X_k = b, X_l = b'] \Pr[Y_k = c, Y_l = c'] \Pr[e_k = 1 \oplus b \oplus c, e_l = 1 \oplus b' \oplus c']
\end{aligned}$$

The probability $\Pr[e_k = 1 \oplus b \oplus c, e_l = 1 \oplus b' \oplus c']$ is straightforward and equals $\frac{\binom{n-2}{w_e - (1 \oplus b \oplus c) - (1 \oplus b' \oplus c')}}{\binom{n}{w_e}}$. For $\Pr[X_k = b, X_l = b']$ and $\Pr[Y_k = c, Y_l = c']$, we proceed as follows.

$$\begin{aligned}
\Pr[X_k = b, X_l = b'] &= \Pr[\bigoplus_{i+j \equiv k \pmod n} x_i r_{2,j} = b, \bigoplus_{i+j \equiv l \pmod n} x_i r_{2,j} = b'] \\
&= \sum_{r_2 \in \mathbb{R}_{w_2}} \Pr[\bigoplus_{i \in S_k} x_i = b, \bigoplus_{i \in S_l} x_i = b'] \\
&= \sum_{s=0}^{w_2} \sum_{d \in \{0,1\}} \mathsf{pmf}_U(n, w_2, k, l, s) \Pr[\bigoplus_{j=1}^{s} x_{i_j} = d, \bigoplus_{j=1}^{w_2 - s} x_{k_j} = b \oplus d, \bigoplus_{j=1}^{w_2 - s} x_{l_j} = b' \oplus d]
\end{aligned} \tag{4}$$

The probability $\mathsf{pmf}_U(n, w_2, k, l, s)$ is computed according to Algorithm 3 over the probability space of n-bit vectors with weight w_2. The probability $\Pr[\bigoplus_{j=1}^{s} x_{i_j} = d, \bigoplus_{j=1}^{w_2 - s} x_{k_j} = b \oplus d, \bigoplus_{j=1}^{w_2 - s} x_{l_j} = b' \oplus d]$ can be computed using Theorem 1, over the sample space of n-bit vectors with weight w.
Using the same argument, we get

$$\Pr[Y_k = c, Y_l = c'] = \sum_{s=0}^{w_1} \sum_{d \in \{0,1\}} \mathsf{pmf}_U(n, w_1, k, l, s) \Pr[\bigoplus_{j=1}^{s} y_{i_j} = d, \bigoplus_{j=1}^{w_1 - s} y_{k_j} = c \oplus d, \bigoplus_{j=1}^{w_1 - s} y_{l_j} = c' \oplus d] \tag{5}$$

Therefore, the variance of the weight of the error vector e' is given by the following theorem.

Theorem 7 (Variance)
Consider the error vector e' in the HQC scheme, defined in Definition 4. Let further E denote the expectation of $\mathsf{weight}(e')$*. The variance of* $\mathsf{weight}(e')$ *is given by:*

$$\mathrm{Var}(\mathsf{weight}(e')) = E - E^2 + 2 \sum_{0 \le k < l < n} \sum_{b,b',c,c' \in \{0,1\}} \frac{\binom{n-2}{w_e - (1 \oplus b \oplus c) - (1 \oplus b' \oplus c')}}{\binom{n}{w_e}} \Pr[X_k = b, X_l = b'] \Pr[Y_k = c, Y_l = c']$$

where $\Pr[X_k = b, X_l = b']$ *and* $\Pr[Y_k = c, Y_l = c']$ *are given by Equations 4 and 5, respectively.* □

5.2 Binomial Case

In the HQC scheme, the error vector for the binomial case is defined as follows.

Definition 5 (HQC's Error Vector—Binomial case). *Let n be a nonnegative integer, and let p, p_1, p_2, p_e be real numbers in $[0, 1]$. The error vector is given by $e' = x \cdot r_2 + y \cdot r_1 + e$, where*

- $x, y \sim \mathsf{Binomial}(n, p)$*;*
- $r_1 \sim \mathsf{Binomial}(n, p_1)$ *and* $r_2 \sim \mathsf{Binomial}(n, p_2)$*;*
- $e \sim \mathsf{Binomial}(n, p_e)$*.*

We maintain the same notation as in Definition 4, namely $X = x \cdot r_2$ and $Y = r_1 \cdot y$.

We now proceed to compute the expectation and variance of $\mathsf{weight}(e')$.

Theorem 8 (Expectation)
Let e' be the error vector described in Definition 5. The expectation $\mathbb{E}(\mathsf{weight}(e'))$ is given by

$$\mathbb{E}(\mathsf{weight}(e')) = n \sum_{b,c \in \{0,1\}} \Pr[X_k = b] \Pr[e_k = c] \Pr[Y_k = 1 \oplus b \oplus c],$$

where

$$\Pr[X_k = b] = \frac{b}{2}\big(1 - (1 - 2pp_2)^n\big) + \frac{1-b}{2}\big(1 + (1 - 2pp_2)^n\big),$$

$$\Pr[e_k = c] = cp_e + (1 - c)(1 - p_e),$$

$$\Pr[Y_k = 1 \oplus b \oplus c] = \frac{1 \oplus b \oplus c}{2}\big(1 - (1 - 2pp_1)^n\big) + \frac{b \oplus c}{2}\big(1 + (1 - 2pp_1)^n\big).$$

Proof. This follows directly from the linearity of expectation and Lemma 3. □

Theorem 9 (Variance)
Let e' be the error vector described in Definition 5, and let E denote the expectation of $\mathsf{weight}(e')$*. The variance of* $\mathsf{weight}(e')$ *is given by*

$$\mathrm{Var}(\mathsf{weight}(e')) = E - E^2 + 2 \sum_{k<l} \sum_{\substack{b,b',c,c' \\ \in \{0,1\}}} \Pr[X_k = b, X_l = b'] \Pr[Y_k = c, Y_l = c'] \Pr[e_k = 1 \oplus b \oplus c,\ e_l = 1 \oplus b' \oplus c']$$

where

$$\Pr[X_k = b, X_l = b'] = \sum_{r=0}^{n} \sum_{s=0}^{r} \sum_{d \in \{0,1\}} \mathsf{pmf}_B(n, p_2, k, l, r, s) \Pr\Big[\bigoplus_{j=1}^{s} x_{i_j} = d\Big] \Pr\Big[\bigoplus_{j=1}^{r-s} x_{k_j} = b \oplus d\Big] \Pr\Big[\bigoplus_{j=1}^{r-s} x_{l_j} = b' \oplus d\Big]$$

$$\Pr[Y_k = c, Y_l = c'] = \sum_{r=0}^{n} \sum_{s=0}^{r} \sum_{d \in \{0,1\}} \mathsf{pmf}_B(n, p_1, k, l, r, s) \cdot \Pr\Big[\bigoplus_{j=1}^{s} y_{i_j} = d\Big] \Pr\Big[\bigoplus_{j=1}^{r-s} y_{k_j} = c \oplus d\Big] \Pr\Big[\bigoplus_{j=1}^{r-s} y_{l_j} = c' \oplus d\Big]$$

$$\Pr[e_k = 1 \oplus b \oplus c, e_l = 1 \oplus b' \oplus c'] = \big((1 \oplus b \oplus c)p_e + (b \oplus c)(1 - p_e)\big) \cdot \big((1 \oplus b' \oplus c')p_e + (b' \oplus c')(1 - p_e)\big)$$

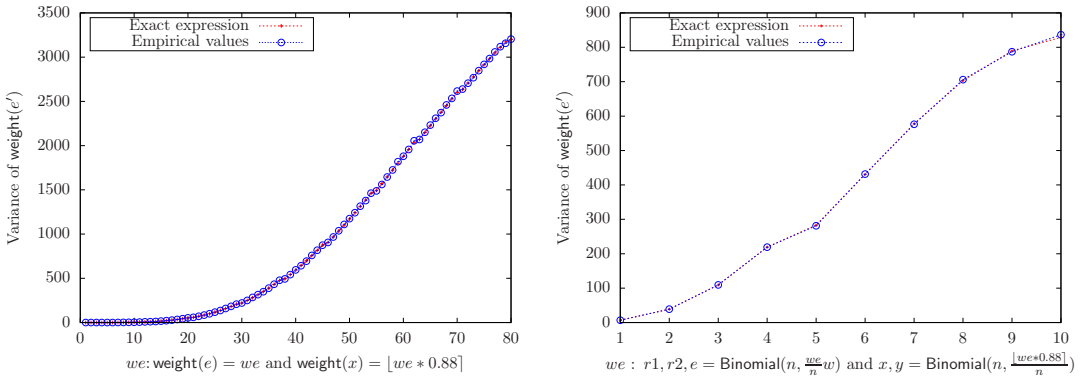
The probabilities $\mathsf{pmf}_B(n, p_1, k, l, r, s)$ *and* $\mathsf{pmf}_B(n, p_2, k, l, r, s)$ *are computed according to Algorithm 4 over the sample spaces* $\mathsf{Binomial}(n, p_1)$ *and* $\mathsf{Binomial}(n, p_2)$*, respectively. The probabilities* $\Pr[\bigoplus_{i=1}^{N} x_i = b]$ *and* $\Pr[\bigoplus_{i=1}^{N} y_i = b]$*, for an arbitrary bit* b *and a nonnegative integer* N*, are computed according to Lemma 3 for* $x, y \sim \mathsf{Binomial}(n, p)$*.* □

5.3 Validation and Comparative Analysis

For our experiments, we consider the following parameter sets.

- **Uniform slice**: We consider the HQC parameters of degree $n = 17669$. The polynomials r_1, r_2, e are we-weighted, while x, y are w-weighted, with $w = \lfloor we \cdot 0.88 \rceil$ and $we \leq 80$.
- **Binomial**: All involved polynomials r_1, r_2, e, x, y have degree less than $n = 503$. The polynomials r_1, r_2, e have Bernoulli coefficients with parameter $\frac{we}{n}$, whereas x, y have Bernoulli coefficients with parameter $\frac{w}{n}$, where $w = \lfloor we \cdot 0.88 \rceil$ and $we \leq 10$. Again, we reduced the parameters while preserving the same ratios as in the uniform case, because the overhead of the binomial setting (computation of the PMF tables) was infeasible on our available hardware.

Experimental Validation. We provide a graphical visualization of the derived variance expressions for the HQC scheme. The empirical variance of $\mathsf{weight}(e')$ is computed over 100,000 trials as follows. For each trial, we generate e' from the required distribution, record the weight $Z = \mathsf{weight}(e')$ and its square Z^2, and compute the empirical variance as $\langle Z^2 \rangle - \langle Z \rangle^2$, where $\langle \cdot \rangle$ denotes the sample mean. The perfect match between the "Exact expression" and "Empirical values" curves in Figure 7 graphically corroborates our derived expressions.

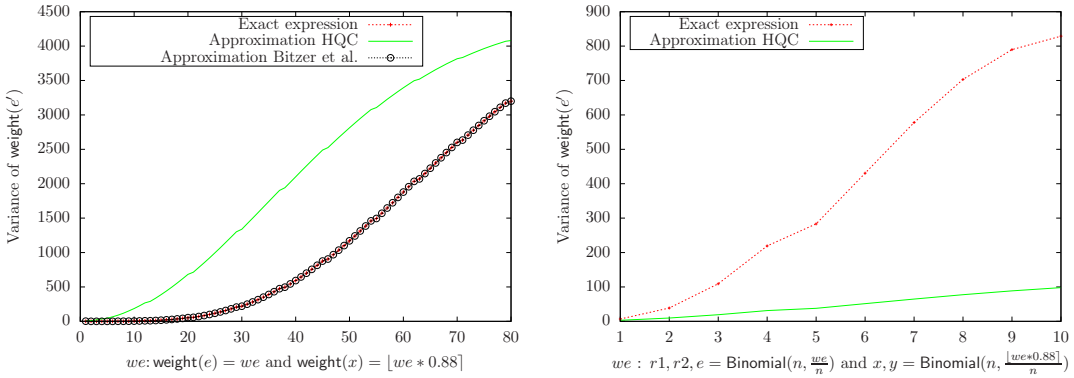


(a) Uniform slice case (b) Binomial case

Fig. 7: Variance of the weight of the error vector e' in the HQC scheme

Comparison with Existing Approximations. Figure 8 compares the exact value of the variance of $\mathsf{weight}(e')$ in both the uniform and binomial settings with existing approximations in the literature. In the uniform setting, we compare our derived formula for the variance against the HQC approximation [7] and the approximation from [4], where we computed the approximated variance using the estimated value of the probability mass function (PMF) of the weight $\mathsf{weight}(e')$ available from the authors' repository [5]. In the binomial setting, we compare our derived value only against the approximation [10].

It is clear that [5] provides an extremely accurate model for the PMF of $\mathsf{weight}(e')$, since the derived variance matches perfectly the exact variance we computed.



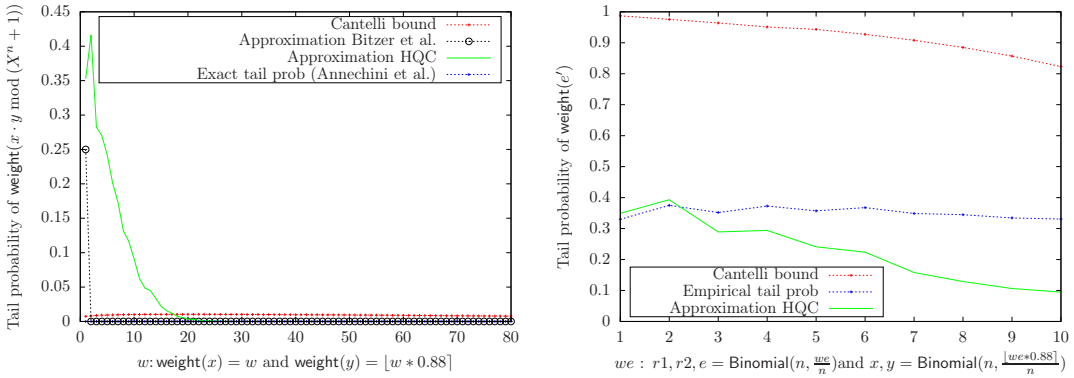
(a) Uniform slice case (b) Binomial case

Fig. 8: Comparison of the variance with existing approximations [7,10,4]

Tail Probability of weight(e'). We compute an upper bound on the tail probability $\Pr[\mathsf{weight}(e') > 1.1 \cdot \mathbb{E}(\mathsf{weight}(e'))]$ for the parameter sets described above using Cantelli's bound with our derived variance. In the uniform setting, we compare our bound against the HQC approximation [7], the PMF-based tail probability from [2], and the approximation from [4], where the approximated variance is computed using the estimated PMF of weight(e') available from [5]. In the binomial setting, we compare against the HQC approximation and the empirical tail probability obtained from 100,000 trials.

Note that we cannot employ the approximation from [8], as it does not carry over to the HQC error vector.

It is clear that Cantelli's bound provides a tight upper bound on the tail probability in the uniform setting. The approximation from [4] again proves to be extremely accurate, as the derived tail probability matches the exact one.



(a) Uniform setting (b) Binomial setting

Fig. 9: Upper bound on the tail probability $\Pr[\mathsf{weight}(e') > 1.1 \cdot \mathbb{E}(\mathsf{weight}(e'))]$

References

1. Laila El Aimani, *Weight of Polynomial Products Mod (X^n+1)-Application to the HQC Cryptosystem-*, IACR Cryptol. ePrint Arch. **2025** (2025), 2180.

2. Alessandro Annechini, Alessandro Barenghi, and Gerardo Pelosi, *On the Decoding Failure Rate of HQC*, IEEE International Symposium on Information Theory, ISIT 2026, 2026, p. To appear.
3. R. Avanzi, S. Bai, J. Bos, J. Ding, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, *CRYSTALS-KYBER*, FIPS 203, 2022, URL: https://pq-crystals.org/kyber/index.shtml.
4. Sebastian Bitzer, Jean-Christophe Deneuville, Emma Munisamy, Bharath Purtipli, Stefan Ritterhoff, and Antonia Wachter-Zeh, *HQC Beyond the Standard: Ciphertext Compression and Refined DFR Analysis*, Advances in Cryptology - EUROCRYPT 2026 - Part VI (Joan Daemen and Emmanuel Thomé, eds.), LNCS, vol. 16546, Springer, pp. 544–574.
5. ———, *HQC with ciphertext compression*, https://gitlab.com/HQCCiphertextCompression/hqc-with-ciphertext-compression (2025).
6. M. Bombar, N. Resch, and E. Wiedijk, *On the Independence Assumption in Quasi-Cyclic Code-Based Cryptography*, IEEE International Symposium on Information Theory, ISIT 2025, IEEE, 2025, pp. 1–6.
7. P. Gaborit, C. Aguilar Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J-C. Deneuville, E. Persichetti, G. Zémor, J. Bos, A. Dion, J. Lacan, J-M. Robert, P. Véron, P. L. Barreto, S. Ghosh, S. Gueron, T. Güneysu, R. Misoczki, J. Richter-Brokmann, N. Sendrier, J-P. Tillich, and V. Vasseur, *HQC (Hamming Quasi-Cyclic)*, FIPS coming soon, 2025, URL: https://pqc-hqc.org/index.html.
8. A. Kawachi, *Hamming Weight of Product of Random Sparse Polynomials*, ISITA 2020, IEEE, 2020, pp. 368–371.
9. ———, *Exponential Concentration Bounds for Hamming Weight of Product of Random Sparse Polynomials*, Unpublished manuscript. Available from the author's website, 2026.
10. Carlos Aguilar Melchor, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor, *Efficient Encryption From Random Quasi-Cyclic Codes*, IEEE Trans. Inf. Theory **64** (2018), no. 5, 3927–3943.

6 Joint PMF of $(|S_k|, |S_k \cap S_l|)$ in the Binomial Model

In this section, we compute the joint PMF of the random variables $(|S_k|, |S_k \cap S_l|)$, for $S_j = \{(j-i) \bmod n : x_i = 1\}$, $j = k, l$, over the set $\mathsf{Binomial}(n, p)$ of n-bit vectors $(x_0, \ldots, x_{n-1})$ with iid Bernoulli(p) coordinates.

As in the uniform slice case, we have:

$$\begin{aligned} e \in S_k \cap S_l &\iff \exists i, j \in \mathbb{Z}_n \colon x_i = x_j = 1 \text{ and } e = (k-i) \bmod n \text{ and } e = (l-j) \bmod n \\ &\iff \exists i, j \in \mathbb{Z}_n \colon x_i = x_j = 1 \text{ and } \{j - i = l - k \text{ or } i - j = n + k - l\} \end{aligned}$$

Therefore, $\Pr[|S_k| = r, |S_k \cap S_l| = s]$ corresponds to the probability that $\mathsf{weight}(x) = r$, and there exist exactly s pairs (i, j) among the following pairs:

$$\{(0, l-k), \ldots, (n+k-l-1, n-1)\} \cup \{(n+k-l, 0) \ldots (n-1, l-k-1)\}$$

satisfying $x_i = x_j = 1$.

We need to distinguish two cases: (1) the case '$n+k-l = l-k$', and (2) the case '$n+k-l \neq l-k$'.

6.1 The Case '$n + k - l = l - k$'

Theorem 10

Given a random n-bit vector $x = (x_0, \ldots, x_{n-1}) \sim \mathsf{Binomial}(n, p)$. For nonnegative integers k and l satisfying $k < l$ and $l - k = n + k - l$, the joint probability mass function of $(|S_k|, |S_k \cap S_l|)$ is given by

$$\Pr[|S_k| = r, |S_k \cap S_l| = s] = \begin{cases} 0 & \text{if } [(s \bmod 2 = 1) \text{ or } (r > n) \text{ or } (s > r)] \\ \displaystyle\sum_{r_0=0}^{r} \binom{l-k}{s/2} \binom{l-k-s/2}{r_0-s/2} \binom{l-k-r_0}{r-r_0-s/2} p^r (1-p)^{n-r} & \text{otherwise} \end{cases} \tag{6}$$

Proof (Sketch). Note that $l - k = n/2$. Since the special cases are straightforward, we focus on the general case:

$$\begin{aligned}
\Pr[|S_k| = r, |S_k \cap S_l| = s] &= \Pr[\mathsf{weight}(x) = r, |S_k \cap S_l| = s] \\
&= \sum_{r_0=0}^{r} \Pr[\mathsf{weight}(x_0, \ldots, x_{l-k-1}) = r_0, \mathsf{weight}(x) = r, |S_k \cap S_l| = s] \\
&= \sum_{r_0=0}^{r} \binom{l-k}{s/2} \binom{l-k-s/2}{r_0 - s/2} p^{r_0} (1-p)^{n/2-r_0} \binom{l-k-r_0}{r-r_0-s/2} p^{r-r_0} (1-p)^{n/2-(r-r_0)} \\
&= \sum_{r_0=0}^{r} \binom{l-k}{s/2} \binom{l-k-s/2}{r_0 - s/2} \binom{l-k-r_0}{r-r_0-s/2} p^{r} (1-p)^{n-r}
\end{aligned}$$

The penultimate equation is justified by an argument analogous to that in Theorem 2's proof, with the following modification: the process of setting the coordinates for the subvector $(x_0, \ldots, x_{n/2-1})$ now requires multiplication by $p^{r_0}(1-p)^{n/2-r_0}$, and likewise, setting the coordinates for $(x_{n/2}, \ldots, x_{n-1})$ requires multiplication by $p^{r-r_0}(1-p)^{n/2-(r-r_0)}$. □

6.2 The Case '$n + k - l \neq l - k$'

To compute the joint PMF of $(|S_k|, |S_k \cap S_l|)$ in case $n + k - l \neq l - k$, we proceed by analogy with the method established for the uniform case. In fact, we consider an undirected graph $\tilde{G}$ on the vertices $x_0, \ldots, x_{n-1}$ (which are iid Bernoulli random variables with parameter p), where adjacency is defined by the rule: vertex x_i is adjacent to vertex x_j if and only if $(i - j) \bmod n \in \{l - k, n + k - l\}$.

We therefore reduce the probability that $(|S_k|, |S_k \cap S_l|) = (r, s)$ to the probability that $\mathsf{weight}(x_0, \ldots, x_{n-1}) = r$ and there exist exactly s edges in $\tilde{G}$ that are incident to vertices both equal to 1.

Likewise, the analysis established for the shape of the graph $\tilde{G}$ in the uniform case holds verbatim here. Thus, we omit a detailed repetition and proceed immediately to compute the aforementioned probability when (1) $\tilde{G}$ is a cycle graph, and (2) $\tilde{G}$ is a disjoint union of cycle graphs.

6.2.1 Cycle Graphs

Let i, j, s be nonnegative integers less than or equal to n. Furthermore, let p be a real number in the closed interval $[0, 1]$.

Definition 6 (Cycle-PMF in the Binomial Model). *For $i > 0$, consider an i-bit vector $(x_0, \ldots, x_{i-1}) \in \{0,1\}^i$ with iid Bernoulli(p) coordinates.*
$\mathsf{cycle\text{-}pmf}_b(i, j, s)$ *denotes the probability that the cycle graph $x_0 - \ldots - x_{i-1} - x_0$ has exactly j vertices equal to* 1 *and s edges incident to vertices both equal to* 1*. The probability is taken over all $(x_0, \ldots, x_{i-1}) \sim \mathsf{Binomial}(i, p)$.*

Remark 2. Note that $\mathsf{cycle\text{-}pmf}_b(i, j, s)$ and $\mathsf{cycle\text{-}pmf}(i, j, s)$ are defined over different probability spaces. For $\mathsf{cycle\text{-}pmf}_b(i, j, s)$, the space consists of all i-bit vectors where each coordinate is set independently with probability p. In contrast, for $\mathsf{cycle\text{-}pmf}(i, j, s)$, the space consists of all i-bit vectors with *exactly* j bits set. In the latter case, the weight j is a fixed parameter of the space itself, not a random outcome.

We introduce now the auxiliary functions needed to compute $\mathsf{cycle\text{-}pmf}_b(i, j, s)$.

Definition 7 (Functions $\tilde{f}_0, \tilde{f}_1, \tilde{g}, \tilde{f}$). *Let i, j, s be nonnegative integers, and p a real number in $[0, 1]$. Furthermore, let $x_0 - \ldots - x_i$ be a linear chain, where $(x_0, \ldots, x_i)$ is an $(i + 1)$-bit random vector with iid Bernoulli(p) coordinates.*

$\tilde{f}_0(i,j,s)$ *denotes the conditional probability that the linear chain* $x_0 - \ldots - x_i$ *has weight* j and *contains exactly* s *edges with both endpoints equal to* 1, given that x_0 *is equal to* 0.

$\tilde{f}_1(i,j,s)$ *denotes the conditional probability that the linear chain* $x_0 - \ldots - x_i$ *has weight* j and *contains exactly* s *edges with both endpoints equal to* 1, given that x_0 *is equal to* 1.

$\tilde{g}(i,j,s,b,c)$ *, for* $b, c \in \{0,1\}$*, denotes the conditional probability that the linear chain* $x_0 - \ldots - x_i$ *has weight* j and *contains exactly* s *edges with both endpoints equal to* 1, given that x_0 *and* x_i *are equal to* b *and* c*, respectively.*

$\tilde{f}(i,j,s)$ *denotes the probability that the linear chain* $x_0 - \ldots - x_i$ *has weight* j and *contains exactly* s *edges with both endpoints equal to* 1.

The probability is taken over all $(i+1)$*-bit vectors* $(x_0, \ldots, x_i)$*, where each coordinate is set independently with parameter* p*.*

Remark 3. Note again the difference between the uniform slice and binomial cases in the definition of the auxiliary functions $\tilde{f}_0, \tilde{f}_1, \tilde{g}, \tilde{f}$: the probability is taken, in the binomial case, over $(i+1)$-bit vectors $(x_0, \ldots, x_i)$, where each coordinate is set independently with parameter p. Thus, the weight j is a random outcome and not a fixed parameter of the probability space, unlike in the uniform slice case.

Let us now proceed to the computation of the above functions. Again, we employ a dynamic programming approach to compute $\tilde{f}_0$, $\tilde{f}_1$, and $\tilde{g}$.

Form. 6. and **Form. 7.** give respectively the formulae for computing $\tilde{f}_0$, $\tilde{f}_1$, and $\tilde{g}$; the base cases are immediate, whilst the recurrence relations follow from

$$\begin{aligned}
\tilde{f}_0(i,j,s) &= \Pr[x_1 = 0 \mid x_0 = 0]\tilde{f}_0(i-1,j,s) + \Pr[x_1 = 1 \mid x_0 = 0]\tilde{f}_1(i-1,j,s) \\
\tilde{f}_1(i,j,s) &= \Pr[x_1 = 0 \mid x_0 = 1]\tilde{f}_0(i-1,j-1,s) + \Pr[x_1 = 1 \mid x_0 = 1]\tilde{f}_1(i-1,j-1,s-1) \\
\tilde{g}(i,j,s,b,c) &= \Pr[x_1 = 0 \mid x_0 = b, x_i = c]\tilde{g}(i-1,j-b,s,0,c) + \Pr[x_1 = 1 \mid x_0 = b, x_i = c] \\
&\quad \tilde{g}(i-1,j-b,s-b,1,c)
\end{aligned}$$

Where

$$\begin{aligned}
\Pr[x_1 = 0 \mid x_0 = 0] = \Pr[x_1 = 0 \mid x_0 = 1] = \Pr[x_1 = 0] = 1-p \\
\Pr[x_1 = 1 \mid x_0 = 0] = \Pr[x_1 = 1 \mid x_0 = 1] = \Pr[x_1 = 1] = p \\
\Pr[x_1 = 0 \mid x_0 = b, x_i = c] = 1-p \text{ and } \Pr[x_1 = 1 \mid x_0 = b, x_i = c] = p
\end{aligned}$$

$\tilde{f}_0(i,j,s) = \tilde{f}_1(i,j,s) = 0$ if $(j > i+1) \vee (s > i) \vee (s > j-1) \vee (s < 0)$. Otherwise:

$$\tilde{f}_0(i,j,s) = \begin{cases} (j=0)\cdot(s=0) & \text{if } i = 0 \\ (1-p)(s=0)(j=0) + p(s=0)(j=1) & \text{if } i = 1 \\ \begin{cases} 0 & \text{if } j > i \\ (1-p)\tilde{f}_0(i-1,j,s) + p\tilde{f}_1(i-1,j,s) & \text{otherwise} \end{cases} & \text{otherwise} \end{cases}$$

$$\tilde{f}_1(i,j,s) = \begin{cases} (j=1)\cdot(s=0) & \text{if } i = 0 \\ (1-p)(s=0)(j=1) + p(s=1)(j=2) & \text{if } i = 1 \\ \begin{cases} 0 & \text{if } j = 0 \text{ or } j-1 > i \\ (1-p)\tilde{f}_0(i-1,j-1,s) + p\tilde{f}_1(i-1,j-1,s-1) & \text{otherwise} \end{cases} & \text{otherwise} \end{cases}$$

Form. 6. Computation of $\tilde{f}_0$ and $\tilde{f}_1$

$\tilde{g}(i,j,s,b,c) = 0$ if $(j > i+1) \vee (s > i) \vee (j > b+c) \vee (s < 0) \vee (i == 0)$. Otherwise:

$$\tilde{g}(i,j,s,b,c) = \begin{cases} (j = b+c) \cdot (s = b \cdot c) & \text{if } i = 1 \\ (1-p)\tilde{g}(i-1, j-b, s, 0, c) + p\tilde{g}(i-1, j-b, s-b, 1, c) & \text{otherwise} \end{cases}$$

Form. 7. Computation of $\tilde{g}$

$$\tilde{f}(i,j,s) = (1-p)\tilde{f}_0(i,j,s) + p\tilde{f}_1(i,j,s)$$

Form. 8. Computation of $\tilde{f}$

$$\mathsf{cycle\text{-}pmf}_b(i,j,s) = \begin{cases} 0 & \text{if } (i=0) \vee (j > i) \vee (s > i) \vee (s > j) \\ \tilde{f}(i-1,j,s) & \text{if } (i < 3), (i > 0) \\ p^i (s=i)(j=i) & \text{if } (s=i) \vee (j=i) \\ \Big[(1-p)\tilde{f}(p, i-2, j, s) + & \\ p \sum\limits_{b,c \in \{0,1\}} [(pb + (1-p)(1-b))(pc + (1-p)(1-c)) & \\ \qquad \tilde{g}(i-2, j-1, s-b-c, b, c)]\Big] & \text{otherwise} \end{cases}$$

Form. 9. Computation of $\mathsf{cycle\text{-}pmf}_b$

Remark 4. Following the same approach outlined in the uniform case, we precompute $\tilde{f}_0(i,j,s)$, $\tilde{f}_1(i,j,s)$, and $\tilde{g}(i,j,s,b,c)$, for all nonnegative integers $0 \le i,j,s \le n$ and all bits b,c, via bottom-up dynamic programming. The $O(n^3)$ precomputation cost in both time and space allows for a constant-time computation of $\mathsf{cycle\text{-}pmf}_b$.

6.2.2 Disjoint Union of Cycle Graphs

Suppose now that $\tilde{G}$ is a disjoint union of d cycle graphs $\tilde{G} = \bigcup_{i=0,\ldots,d-1} \tilde{G}_i$, where $d > 1$. Recall that $\tilde{G}$ is defined on the vertices $x_0, \ldots, x_{n-1}$, which are iid Bernoulli(p).

We maintain the notation from the uniform case: S denotes the total number of edges in $\tilde{G}$ that are incident to two vertices both assigned the value 1. Moreover, $\mathsf{weight}(\tilde{G})$, which we denote by R, is the total number of vertices that are equal to 1. Correspondingly, for $0 \le i \le d-1$, S_i denotes the total number of edges in $\tilde{G}_i$ that are incident to two vertices both assigned the value 1, and $\mathsf{weight}(\tilde{G}_i)$ is the number of vertices equal to 1 in $\tilde{G}_i$.

$$\begin{aligned} \Pr[R=r, S=s] &= \sum_{\substack{r_0,\ldots,r_{d-1} \\ \sum r_i = r}} \sum_{\substack{s_0,\ldots,s_{d-1} \\ \sum s_i = s}} \Pr[\bigwedge_{i=0}^{d-1} (\mathsf{weight}(\tilde{G}_i) = r_i, S_i = s_i)] \\ &= \sum_{\substack{r_0,\ldots,r_{d-1} \\ \sum r_i = r}} \sum_{\substack{s_0,\ldots,s_{d-1} \\ \sum s_i = s}} \prod_{i=0}^{d-1} \mathsf{cycle\text{-}pmf}_b(p, n/d, r_i, s_i) \end{aligned}$$

Likewise, $\Pr[R=r, S=s]$ can be computed in a dynamic programming approach. Let D be the largest divisor of n, and consider the following function:

$$\begin{aligned} \tilde{h} \colon [0,D] \times [0,n] \times [0,n] \times [0,D] &\longrightarrow \mathbb{R} \\ (N,r,s,d) &\longmapsto \sum_{\substack{r_0,\ldots,r_{d-1} \\ \sum r_i = r}} \sum_{\substack{s_0,\ldots,s_{d-1} \\ \sum s_i = s}} \prod_{i=0}^{d-1} \mathsf{cycle\text{-}pmf}_b(N, r_i, s_i) \end{aligned}$$

Then, $\tilde{h}$ can be computed recursively using the following recurrence:

$$\begin{aligned}\tilde{h}(N,r,s,d) &= \sum_{r_0=0}^{r}\sum_{s_0=0}^{s} \mathsf{cycle\text{-}pmf}_b(N,r_0,s_0) \sum_{\substack{r_1,\ldots,r_{d-1}\\ \sum r_i = r-r_0}} \sum_{\substack{s_1,\ldots,s_{d-1}\\ \sum s_i = s-s_0}} \prod_{i=1}^{d-1} \mathsf{cycle\text{-}pmf}_b(N,r_i,s_i) \\ &= \sum_{w_0=0}^{w}\sum_{s_0=0}^{s} \mathsf{cycle\text{-}pmf}_b(N,r_0,s_0)\tilde{h}(N,r-r_0,s-s_0,d-1)\end{aligned} \tag{7}$$

The algorithm for computing the joint PMF of $(|S_k|, |S_k \cap S_l|)$ is straightforward.

Algorithm 4: Computation of the joint PMF of $(|S_k|, |S_k \cap S_l|)$

Input: nonnegative integers n, k, l, r, s with $k < l < n$ and $s, r \le n$, and a real number $p \in [0,1]$

Output: $\Pr[|S_k| = r, |S_k \cap S_l| = s] = \mathsf{pmf}_B(n,p,k,l,r,s)$

if $n + k - l = l - k$ **then**

$$res = \begin{cases} 0 & \text{if } [(s \bmod 2 = 1) \text{ or } (r > n) \text{ or } (s > r)] \\ \sum_{r_0=0}^{r} \binom{l-k}{s/2}\binom{l-k-s/2}{r_0-s/2}\binom{l-k-r_0}{r-r_0-s/2} p^r (1-p)^{n-r} & \text{otherwise} \end{cases}$$

else

```
// We assume the values f̃_0(i,j,t), f̃_1(i,j,t), g̃(i,j,t,b,c), and
// h̃(N,j,t,f), for all parameter sets under consideration, are
// precomputed using dynamic programming as mentioned earlier
```

$d = \gcd(l-k, n)$;

if $d = 1$ **then**

$res = \mathsf{cycle\text{-}pmf}_b(n,r,s)$; `/* cycle-pmf_b is computed according to Form. 9. */`

else

$res = \tilde{h}(n/d, r, s, d)$;

return res;

Remark 5. We remark that, besides depending on n and p, the joint PMF of $(|S_k|, |S_k \cap S_l|)$ is a function solely of $\gcd(l-k, n)$. Consequently, the optimization and the result when n is prime, detailed in Subsubsections 4.3.2 and 4.3.3 respectively, remain applicable here for the binomial case.

Remark 6. Unlike the uniform slice case, the batching benefit no longer applies here. Since the tables f_0, f_1, g, h are functions of the Bernoulli parameter p, they must be rebuilt for each p, which precludes any reuse across different parameter values.

6.3 Experimental Validation

As in the uniform slice case, we experimentally validate our theoretical formulae for the joint probability $\Pr[|S_k| = r, |S_k \cap S_l| = s]$. We do this using a sample size of $100,000$. In each sample, we generate a polynomial of degree less than $n = 250$ and with Bernoulli($p = 0.26$) coefficients, then record the values $R = |S_k|$ and $S = |S_k \cap S_l|$; the empirical value of $\Pr[|S_k| = r, |S_k \cap S_l| = s]$ is estimated as the proportion of trials where $R = r$ and $S = s$. We plot this joint PMF in Figure 10 for the three values: (i) $l - k = 17$, corresponding to the case '$\gcd(l-k, n) = 1$', (ii) $l - k = 25$, corresponding to the case '$\gcd(l-k, n) \neq 1$', and (iii) $l - k = 125$, corresponding to the case '$l - k = n + k - l$'. The perfect correspondence between the 'Theory' and 'Experiments' curves verifies our formulae.

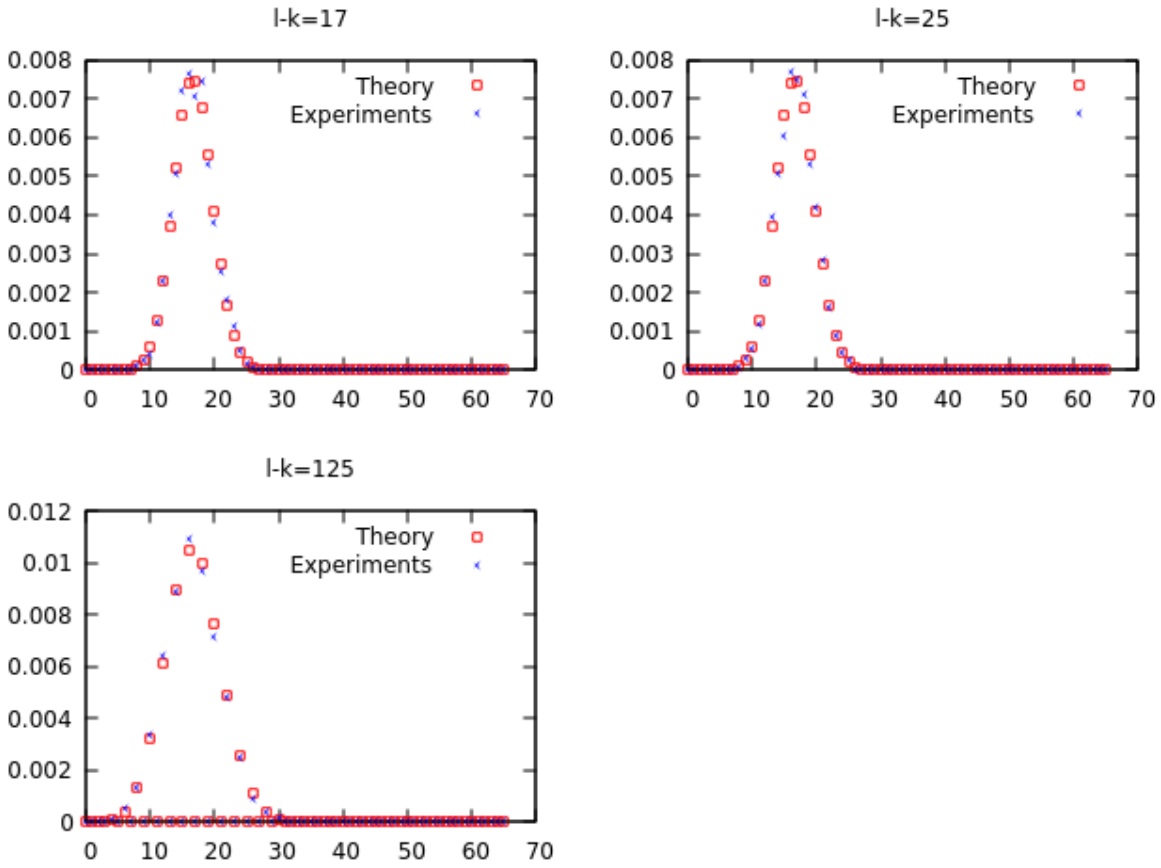


Fig. 10: Joint probability $\Pr[|S_k| = r, |S_k \cap S_l| = s]$ for $n = 250$, $p = 0.26$, and $r = 65$. The x-axis corresponds to the values $s \in [0, r]$