The impact of error dependencies on Ring/Mod-LWE/LWR based schemes

Jan-Pieter D'Anvers, Frederik Vercauteren, and Ingrid Verbauwhede

imec-COSIC, KU Leuven
Kasteelpark Arenberg 10, Bus 2452, B-3001 Leuven-Heverlee, Belgium
firstname.lastname@esat.kuleuven.be

Abstract Current estimation techniques for the probability of decryption failures in Ring/Mod-LWE/LWR based schemes assume independence of the failures in individual bits of the transmitted message to calculate the full failure rate of the scheme. In this paper we disprove this assumption both theoretically and practically for schemes based on Ring/Mod-Learning with Errors/Rounding. We provide a method to estimate the decryption failure probability, taking into account the bit failure dependency. We show that the independence assumption is suitable for schemes without error correction, but that it might lead to underestimating the failure probability of algorithms using error correcting codes. In the worst case, for LAC-128, the failure rate is 2^{48} times bigger than estimated under the assumption of independence. This higher-than-expected failure rate could lead to more efficient cryptanalysis of the scheme through decryption failure attacks.

Keywords: Lattice cryptography, Ring-LWE, Error Correcting Codes, Decryption Failures

1 Introduction

Due to the recent developments in quantum computing and its threat to current asymmetric key schemes, the cryptographic community has increased its efforts towards the development of post-quantum cryptography, resulting in the NIST Post-Quantum Standardization Process. Several submissions to this process are built on top of the Learning with Errors (LWE) hard problem. These are frequently combined with the usage of polynomial matrix elements, resulting in Ring-LWE or Mod-LWE schemes such as New Hope [1], LAC [14], LIMA [16], R.Emblem [15] and Kyber [2]. Some schemes further reduce their communication bandwidth by replacing the pseudorandomly generated errors terms with rounding errors, resulting in Ring-LWR and Mod-LWR schemes as in Round2 [8] and Saber [3] respectively.

For most of the above encryption schemes there is a small probability of a decryption failure, in which the decryption of the encoded message returns a faulty result, where one or more message bits are flipped. As these failure events depend on the secret key, they might compromise the security of the scheme.

Therefore, most candidates of the Post-Quantum Standardization Process aim for a failure probability around 2^{-128}. To reduce the failure rate, some schemes utilize error correcting codes (ECC) to make the decryption resilient against a certain number of errors. The NIST candidate LAC [14] relies on extensive error correction, and Fritzmann et al. [6] made a study on the positive impact of the usage of ECC's on the security and bandwidth of lattice-based schemes. Another possibility is to eliminate decryption failures altogether and thus eliminate attacks that exploit them, by selecting the parameter of the scheme accordingly. This comes at the price of a higher bandwidth and computational complexity. However, as most submissions to the NIST Post-Quantum Process have a small decryption failure probability, an analysis of the impact of decryption failures is essential.

A chosen ciphertext attack against Ring-Learning with Errors (Ring-LWE) schemes exploiting decryption failures was reported by Fluhrer [5]. This attack uses knowledge of failing ciphertexts to retrieve the secret. D'Anvers et al. [4] analyzed a decryption failure attack on (Ring/Mod)-LWE/LWR schemes that have protection against chosen ciphertext attacks. The security risk of decryption failures is also reflected in the post-quantum versions [11,12] of the Fujisaki-Okamoto transformation [7], which converts a chosen plaintext secure encryption scheme in a chosen ciphertext secure key encapsulation mechanism (KEM). The security bound of these transformations contains a term considering decryption failures. As this term is quadratic in the failure rate of the underlying scheme, it has an important effect on the security bound.

Consequently, the failure probability is an important factor in the security of these schemes and should be determined precisely. The common approach for computing this probability is calculating the failure rate for one bit of the message, from which the full failure rate is determined assuming the failures between the individual bits are independent. Jin and Zhao [13] proved that for some schemes the failures in individual bits are asymptotically independent if the number of bits goes to infinity. Hamburg [9] did an analysis of the independence of the bits for the NIST Post-Quantum Standardization Process submission ThreeBears [10], which is based on the Integer Module Learning with Errors problem. He identified three sources of correlation: the norm of the secret, the norm of the ciphertext and the correlation between the failures of the individual bits due to the ring structure.

In this paper, we examine the independence assumption for Ring/Mod-LWE/LWR based schemes. First we show both theoretically as well as experimentally that this assumption is not correct. Then, we develop a method to handle the dependency issue in the failure rate calculation. We calculate the failure rate for variants of LAC and validate our method using experimental data[1]. Finally, we discuss the implications of the dependency in different scenarios: for schemes without error correcting codes, we reason that the assumption of independence leads to a slight overestimation of the failure probability. Looking into schemes

[1] The software is available at `https://github.com/KULeuven-COSIC/PQCRYPTO-decryption-failures`

using error correcting codes to reduce the failure rate, we show that the independence assumption can lead to an underestimation of the failure rate, and thus an overestimation of the security of the underlying scheme. In the most extreme case for LAC-128, the failure rate is overestimated by a factor 2^{48}.

2 Preliminaries

2.1 Notation

Let $\mathbb{Z}_q$ denote the ring of integers modulo q, let R_q represent the ring $\mathbb{Z}_q[X]/(X^n+1)$ and let $R_q^{l_1 \times l_2}$ designate the ring of $l_1 \times l_2$ matrices over R_q. Polynomials will be written using lowercase letters, vectors with bold lowercase, and matrices with bold uppercase. The l_2-norm of a polynomial x is defined as $\|x\|_2 = \sqrt{\sum_i x_i^2}$ and the l_2-norm of a vector $\boldsymbol{x}$ as $\|\boldsymbol{x}\|_2 = \sqrt{\sum_i \|x_i\|_2^2}$. The rounding operation $\lfloor x \rceil_{q \to p}$ for $x \in \mathbb{Z}_q$, is calculated as $\lfloor p/q \cdot x \rceil \in \mathbb{Z}_p$. The $\texttt{abs}()$ function takes the absolute value of its input. These operations are extended coefficient-wise for polynomials and vectors. Let a_i, with $a \in R_q$ denote the i^{th} coefficient of a, and denote with $\boldsymbol{a}_i$ for $\boldsymbol{a} \in R_q^{l \times 1}$ the $(i \mod l)^{\text{th}}$ coefficient of the $\lfloor i/l \rfloor^{\text{th}}$ polynomial of $\boldsymbol{a}$.

Let $x \leftarrow \chi(R_q)$ indicate sampling the coefficients of $x \in R_q$ according to distribution χ. The sampling operation is extended coefficient-wise for vectors $\boldsymbol{x} \in R_q^{l \times 1}$ as $\boldsymbol{x} \leftarrow \chi(R_q^{l \times 1})$. Let $\texttt{Binom}(k, n, p)$ be the cumulative binomial distribution with n draws and probability p, so that $\texttt{Binom}(k, n, p) = \sum_{i=0}^{\lfloor k \rfloor} \binom{n}{i} p^i (1-p)^{n-i}$ and let $\texttt{hypergeom}(k, N, K, n)$ be the hypergeometric distribution with population size N, success states K and draws n as defined by:

$$\texttt{hypergeom}(k, N, K, n) = \frac{\begin{pmatrix} K \\ k \end{pmatrix} \begin{pmatrix} N-K \\ n-k \end{pmatrix}}{\begin{pmatrix} N \\ n \end{pmatrix}}, \tag{1}$$

$$\text{where: } \begin{pmatrix} a \\ b \end{pmatrix} = \frac{a!}{b!(a-b)!}. \tag{2}$$

2.2 Ring/Mod-LWE/LWR based encryption

A general framework for Ring/Mod-LWE-LWR based encryption schemes is provided in Algorithms 1 to 3. The algorithm uses the function $\texttt{gen}$ to generate the pseudorandom matrix $\boldsymbol{A}$ from a seed $seed_{\boldsymbol{A}}$, the function $\texttt{enc}$ to encode the message m into an element of R_q and the inverse function $\texttt{dec}$ to decode a polynomial back into a message bitstring. The latter decodes coefficients of the polynomial correctly if the deviation from the initial encoded polynomial coefficient is at most $\pm q/4$. If error correcting codes are used in the scheme, the function $\texttt{ecc\_enc}$ adds extra redundancy to the bitstring m to enable error correction, while $\texttt{ecc\_dec}$ recovers the original message if the number of flipped

Algorithm 1: `PKE.KeyGen()`

```
$seed_{\boldsymbol{A}} \leftarrow \mathcal{U}(\{0,1\}^{256})$
$\boldsymbol{A} \leftarrow \mathsf{gen}(\text{seed}_{\boldsymbol{A}}) \in R_q^{l \times l}$
$\boldsymbol{s}_A \leftarrow \chi_s(R_q^{l \times 1}), \boldsymbol{e}_A \leftarrow \chi_e(R_q^{l \times 1})$
$\boldsymbol{b} = \lfloor \boldsymbol{A}\boldsymbol{s}_A + \boldsymbol{e}_A \rceil_{q \to p}$
return $(pk := (\boldsymbol{b}, seed_{\boldsymbol{A}}), sk := \boldsymbol{s}_A)$
```

Algorithm 2: `PKE.Enc`$(pk = (\boldsymbol{b}, seed_{\boldsymbol{A}}), m, r)$

```
$\boldsymbol{A} \leftarrow \mathsf{gen}(\text{seed}_{\boldsymbol{A}}) \in R_q^{l \times l}$
$\boldsymbol{s}'_B \leftarrow \chi_s(R_q^{l \times 1}), \boldsymbol{e}'_B \leftarrow \chi_e(R_q^{l \times 1})$
$e''_B \leftarrow \chi_e(R_q)$
$\boldsymbol{b}_r = \lfloor \boldsymbol{b} \rceil_{p \to q}$
$\boldsymbol{b}' = \lfloor \boldsymbol{A}^T \boldsymbol{s}'_B + \boldsymbol{e}'_B \rceil_{q \to p}$
$m_{ecc} = \mathtt{ecc\_enc}(m)$
$v' = \lfloor \boldsymbol{b}_r^T \boldsymbol{s}'_B + e''_B + \mathtt{enc}(m_{ecc}) \rceil_{q \to t}$
return $c = (v', \boldsymbol{b}')$
```

bits between m_{ecc} and m'_{ecc} is less than a threshold d, which depends on the chosen error correcting code (ECC). When no error correcting codes are used, the functions `ecc_enc` and `ecc_dec` act as the identity and return their input. The encryption algorithm `PKE.Enc` uses the seed r to pseudorandomly generate $\boldsymbol{s}'_B, \boldsymbol{e}'_B$ and e''_B.

By choosing $l = 1$, one obtains a Ring based scheme, while a bigger value of l indicates a module (Mod) based scheme. In Mod/Ring-LWE based schemes, the error distribution χ_e is nonzero, in contrast to Mod/Ring-LWR based schemes where $\chi_e = 0$. In the latter case, parameters p and t are smaller than q, so that the rounding operations $\lfloor \cdot \rceil_{q \to p}$ and $\lfloor \cdot \rceil_{q \to t}$ introduce the errors necessary for security. The rounding additionally compresses the ciphertexts. The rounding operations $\lfloor \cdot \rceil_{p \to q}$ and $\lfloor \cdot \rceil_{t \to q}$ decompress the input back to approximately the original value. The error introduced by these rounding and reconstruction operations will be denoted as follows:

$$\boldsymbol{u}_A = \boldsymbol{A}\boldsymbol{s}_A + \boldsymbol{e}_A - \boldsymbol{b}_r, \tag{3}$$

$$\boldsymbol{u}'_B = \boldsymbol{A}^T \boldsymbol{s}'_B + \boldsymbol{e}'_B - \boldsymbol{b}'_r, \tag{4}$$

$$u''_B = \boldsymbol{b}_r^T \boldsymbol{s}'_B + e''_B + \mathtt{enc}(m_{ecc}) - v'_r \ . \tag{5}$$

As a first step in determining the error probability of the encryption scheme, we can calculate the value of $v'_r - v$ as follows:

$$v'_r - v = (\boldsymbol{b}_r^T \boldsymbol{s}'_B + e''_B + \lfloor q/2 \rfloor \mathtt{enc}(m_{ecc}) + u''_B) - \boldsymbol{b}'^T_r \boldsymbol{s}_A \tag{6}$$

$$= \lfloor q/2 \rfloor \mathtt{enc}(m_{ecc}) + (\boldsymbol{e}_A + \boldsymbol{u}_A)^T \boldsymbol{s}'_B - (\boldsymbol{e}'_B + \boldsymbol{u}'_B)^T \boldsymbol{s}_A + (u''_B + e''_B) \tag{7}$$

Algorithm 3: $\texttt{PKE.Dec}(sk = \boldsymbol{s}_A, c = (v', \boldsymbol{b'}))$

```
$\boldsymbol{b}'_r = \lfloor \boldsymbol{b}' \rceil_{p \to q}$
$v'_r = \lfloor v' \rceil_{t \to q}$
$v = \boldsymbol{b}'^T_r \boldsymbol{s}_A$
$m'_{ecc} = \texttt{dec}(v'_r - v)$
$m' = \texttt{ecc\_dec}(m'_{ecc})$
return $m'$
```

The distribution of one coefficient of $-(\boldsymbol{e}'_B + \boldsymbol{u}'_B)^T \boldsymbol{s}_A + (\boldsymbol{e}_A + \boldsymbol{u}_A)^T \boldsymbol{s}'_B + (u''_B + e''_B)$ can be calculated exhaustively. For the sake of convenience, we will rewrite this as $\boldsymbol{c}^T \boldsymbol{s} + g$, where $\boldsymbol{s}$ is the vector constructed as the concatenation of $-\boldsymbol{s}_A$ and $(\boldsymbol{e}_A + \boldsymbol{u}_A)$, where $\boldsymbol{c}$ is constructed similarly as the concatenation of $(\boldsymbol{e}'_B + \boldsymbol{u}'_B)$ and $\boldsymbol{s}'_B$, and where $g = u''_B + e''_B$:

$$\boldsymbol{s} = \begin{pmatrix} -\boldsymbol{s}_A \\ \boldsymbol{e}_A + \boldsymbol{u}_A \end{pmatrix}, \quad \boldsymbol{c} = \begin{pmatrix} \boldsymbol{e}'_B + \boldsymbol{u}'_B \\ \boldsymbol{s}'_B \end{pmatrix}, \quad g = u''_B + e''_B. \tag{8}$$

A coefficient of the polynomial $v'_r - v$ decodes correctly if the absolute value of the corresponding coefficient of the error term $\boldsymbol{c}^T \boldsymbol{s} + g$ is smaller than $q/4$. A higher value results in a flipped bit after decoding, which will be called a bit error and will be denoted with F_i with i the position of the bit in the message. If the number of bit errors exceeds the threshold for error correction d, a decryption failure occurs, which we will denote with the symbol F. A correct decryption will be denoted with S, so that by definition $P[S] = 1 - P[F]$.

In Table 1, the parameters for LAC-128 and LAC-256 [14] are given. These schemes are used throughout this paper to validate our methodology, as their high failure rate and significant error correction causes their failure rate calculation to be more sensitive to error dependencies. Due to the choices of the moduli q, p and t, the rounding errors $\boldsymbol{u}_A$, $\boldsymbol{u}'_B$ equal the zero vector and u''_B is the zero polynomial.

	q	p	t	n	l	d
LAC-128	251	251	251	512	1	29
LAC-256	251	251	251	1024	1	55

Table 1: Parameters for LAC

2.3 Key Encapsulation Mechanism

From an IND-CPA secure encryption scheme, an IND-CCA secure Key Encapsulation Mechanism (KEM) can be constructed using a post-quantum version [11]

Algorithm 4: $\texttt{KEM.Encaps}(pk)$

```
$m \leftarrow \mathcal{U}(\{0,1\}^{256})$
$r = \mathcal{G}(m)$
$c = \texttt{PKE.Enc}(pk, m, r)$
$K = \mathcal{H}(r)$
return $(c, K)$
```

Algorithm 5: $\texttt{KEM.Decaps}(sk, pk)$

```
$m' = \texttt{PKE.Dec}(sk, c)$
$r' = \mathcal{G}(m')$
$c' = \texttt{PKE.Enc}(pk, m', r')$
if $c = c'$ then
    return $K = \mathcal{H}(r)$
else
    return $K = \perp$
```

of the Fujisaki-Okamoto transformation. The key generation phase is the same as Algorithm 1 and the Encapsulation and Decapsulation functions are defined in Algorithms 4 and 5 respectively, with $\mathcal{G}$ and $\mathcal{H}$ hash functions that model Random Oracles.

3 Error dependency

The typical method to calculate the failure rate, is to determine the error probability of a single bit of m'_{ecc}, calculated as $p_b = P[|(\boldsymbol{c}^T\boldsymbol{s} + g)_i| > q/4]$, and then assume independence to extend this error probability to the full failure rate. For a scheme that does not use any error correction, this can be expressed as $1 - (1 - p_b)^{l_m}$ or $1 - \texttt{Binom}(0, l_m, p_b)$, with l_m the length of the encoded message m_{ecc}. For schemes that deploy error correcting codes with a correction capability of d errors, the failure rate amounts to $1 - \texttt{Binom}(d, l_m, p_b)$.

However, this assumption of independence is not correct. In this section we will show both theoretically and experimentally that there is a positive correlation between the errors of the bits in m'_{ecc}. Intuitively, one can make the following reasoning: $(\boldsymbol{c}^T\boldsymbol{s} + g)$ with high norm for $\boldsymbol{s}$ and $\boldsymbol{c}$ is more likely to produce bit errors, and conversely, bit errors are also more likely to stem from high norm $\boldsymbol{s}$ and $\boldsymbol{c}$. Therefore, a bit error at a certain location, increases the expected norm of $\boldsymbol{s}$ and $\boldsymbol{c}$, therefore increasing the bit error probabilities at other locations. In conclusion, bit errors are expected to be positively correlated.

In Figure 1, the probability of various number of bit errors in m'_{ecc} is plotted for LAC-256, both experimentally by running the protocol for approximately 2^{31} times, and theoretically under the independence assumption as $1 - \texttt{Binom}(0, l_m, p_b)$, where p_b is determined experimentally. The choice for LAC stems from the fact that the error probability of a bit of m'_{ecc} is large compared to other schemes, making it possible to experimentally obtain enough errors to get accurate estimations. In Figure 1, one can see that the errors are clustered: there are more messages without errors and more messages with a high number of errors than predicted by the theoretical model, which confirms our hypothesis that the bit errors are positively correlated. Note that the error probability of a single bit is the same for the model and the experimental data, and that the errors are just more clustered compared to the prediction of the model.

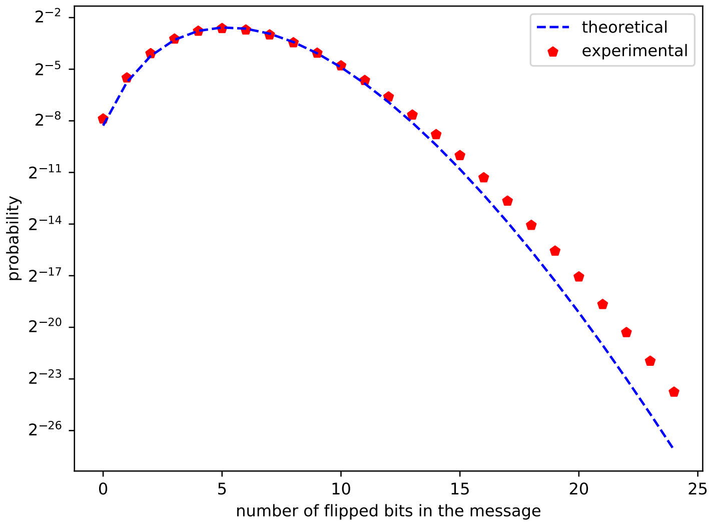

Figure 1: The probability of a certain number of errors in m'_{ecc}

3.1 Handling the dependency

In this section, we will develop a methodology to calculate the failure rate taking into account the dependency between the errors in the bits of m'_{ecc}. For the sake of simplicity, we will first assume that there is no error correcting code.

$$\begin{aligned} 1 - P[F] &= P[S] && (9) \\ &= P[S_0 \cdots S_n] && (10) \end{aligned}$$

Under the independence assumption, one can derive the formulas of the previous section as follows:

$$\begin{aligned} 1 - P[F] &= \prod_i P[S_i] && (11) \\ &= (1 - P[F_0])^n && (12) \end{aligned}$$

However step (11) is not valid if this assumption does not hold. To work around this issue, we involve conditional information in the form of $\boldsymbol{s}, \boldsymbol{c}$ and g:

$$1 - P[F] = \sum_{\boldsymbol{s},\boldsymbol{c},g} P[S_0 \cdots S_n \,|\, \boldsymbol{s}, \boldsymbol{c}, g] P[\boldsymbol{s}, \boldsymbol{c}, g] \tag{13}$$

As the S_i's are fully determined conditioned on $\boldsymbol{s}, \boldsymbol{c}$ and g, the error or success of other bits does not convey any extra information. Therefore, the bit successes S_i are independent conditioned on the extra information, so we can write:

$$\begin{aligned} 1 - P[F] &= \sum_{\boldsymbol{s},\boldsymbol{c},g} \prod_i \left(P[S_i \,|\, \boldsymbol{s}, \boldsymbol{c}, g]\right) P[\boldsymbol{s}, \boldsymbol{c}, g] && (14) \\ &= \sum_{\boldsymbol{s},\boldsymbol{c},g} \left(1 - P[F_0 \,|\, \boldsymbol{s}, \boldsymbol{c}, g]\right)^n P[\boldsymbol{s}, \boldsymbol{c}, g] && (15) \end{aligned}$$

Unfortunately, this expression is not efficiently computable.

Note that the e''_B term of g_j does not add any information to S_i if $j \neq i$ and that its coefficients are independent. We will assume that this is also the case for u''_B, so we can write:

$$P[S_i | \boldsymbol{s}, \boldsymbol{c}, g] \approx P[S_i | \boldsymbol{s}, \boldsymbol{c}, g_i] \tag{16}$$

From this result we can see that g has little or no contribution to the dependency between the S_i. As discussed in Section 3, the norm of $\boldsymbol{s}$ and $\boldsymbol{c}$ is an important cause of dependency. For rings of the form $\mathbb{Z}[X]/(X^n + 1)$ we could assume that this is the main cause of correlation, as different coefficients of $\boldsymbol{c}^T \boldsymbol{s}$ are calculated with different combinations of elements of $\boldsymbol{c}$ and $\boldsymbol{s}$, which can be formalized as follows:

Assumption 1. *For $\boldsymbol{s}, \boldsymbol{c}$ and g as described in equation (8), where g and the coefficients of $\boldsymbol{s}$ and $\boldsymbol{c}$ are elements of the ring $\mathbb{Z}[X]/(X^n + 1)$, we can approximate $S_0 \cdots S_n$ to be independent conditioned on $\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2$, which is equivalent to $P[S_0 \cdots S_n \,|\, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] \approx \prod_i P[S_i \,|\, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2]$.*

Using this assumption we write:

$$\begin{aligned} 1 - P[F] &= \sum_{\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2} P[S_0 \cdots S_n \,|\, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] P[\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] && (17) \\ &\approx \sum_{\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2} \prod_i \left(P[S_i \,|\, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2]\right) P[\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] && (18) \\ &\approx \sum_{\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2} \left(P[S_0 \,|\, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2]\right)^n P[\|\boldsymbol{s}\|_2] P[\|\boldsymbol{c}\|_2] && (19) \\ &\approx \sum_{\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2} \left(1 - P[F_0 \,|\, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2]\right)^n P[\|\boldsymbol{s}\|_2] P[\|\boldsymbol{c}\|_2] && (20) \end{aligned}$$

Using a similar derivation, the failure rate for schemes with error correction under Assumption 1 can be calculated as:

$$1 - P[F] \approx \sum_{\|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2} (1 - \mathtt{Binom}(d, l_m, p_b))\, P[\|\boldsymbol{s}\|_2] P[\|\boldsymbol{c}\|_2] \tag{21}$$

$$\text{where: } p_b = P[F_0 \mid \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] \tag{22}$$

To conclude, one has to calculate the failure rate for every value of $\|\boldsymbol{s}\|_2$ and $\|\boldsymbol{c}\|_2$, after which the failure rate can be found by taking a weighted average. The model from equation (20) can be seen as an intermediate between the model from equation (12) that was constructed using the independence assumption, and the exact but incalculable model from equation (15). In this intermediate model, the main source of correlation between the S_i, following Assumption 1, is taken into account. In the next section we will experimentally assess our intermediate model and observe that it closely represents the experimental data, thus validating our assumption.

3.2 Experiments

To validate the developed methodology, we ran LAC-256 approximately 2^{31} times to get experimental data on the probability of a certain number of failures in m'_{ecc}. We calculated the same probability using the assumption of independence and our dependency aware model.

In general $P[F_0 \mid \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2]$ can be calculated using a Gaussian assumption on the distribution of $\boldsymbol{c}^T\boldsymbol{s} + g$ as described in [4]. For our calculations of LAC we use a more exact algorithm using the fact that the elements of $\boldsymbol{c}, \boldsymbol{s}$ and g are ternary. Intuitively, we first calculate the probability that a certain number l of nonzero coefficients of $\boldsymbol{c}$ and $\boldsymbol{s}$ coincide during the multiplication, expressed as $P[(\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l \mid \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2]$. Then, we assume the term $(\boldsymbol{c}^T\boldsymbol{s})_0$ given $(\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l$ to be a sum of l elements randomly picked as plus or minus 1. The full derivation can be expressed as follows:

$$p_b = P[\mathtt{abs}(\boldsymbol{c}^T\boldsymbol{s} + g)_0 > q/4 \mid \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] \tag{23}$$

$$= \sum_l \begin{pmatrix} P[\mathtt{abs}(\boldsymbol{c}^T\boldsymbol{s} + g)_0 > q/4 \mid (\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] \cdot \\ P[(\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l \mid \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] \end{pmatrix} \tag{24}$$

$$= \sum_l \begin{pmatrix} P[\mathtt{abs}(\boldsymbol{c}^T\boldsymbol{s} + g)_0 > q/4 \mid (\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l] \cdot \\ P[(\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l \mid \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] \end{pmatrix} \tag{25}$$

$$= \sum_l \sum_{g_0} \begin{pmatrix} P[\mathtt{abs}(\boldsymbol{c}^T\boldsymbol{s} + g)_0 > q/4 \mid (\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l, g_0] \cdot \\ P[(\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l \mid \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2] \cdot P[g_0] \end{pmatrix} \tag{26}$$

We can model $P[(\boldsymbol{c}^T\boldsymbol{s})_0 > q/4 - g_0 \mid (\mathtt{abs}(\boldsymbol{c})^T\mathtt{abs}(\boldsymbol{s}))_0 = l, g_0]$ as the survival function of a binomial distribution, which can be calculated as $\mathtt{Binom}(\frac{l-q/4+g_0}{2}, l, 1/2)$.

Similarly, $P[(\boldsymbol{c}^T\boldsymbol{s})_0 < -q/4 - g_0 \,|\, (\texttt{abs}(\boldsymbol{c})^T\texttt{abs}(\boldsymbol{s}))_0 = l, g_0]$ can be modelled as $\texttt{Binom}(\frac{l-q/4-g_0}{2}, l, 1/2)$, so that $P[\texttt{abs}(\boldsymbol{c}^T\boldsymbol{s}+g)_0 > q/4 \,|\, (\texttt{abs}(\boldsymbol{c})^T\texttt{abs}(\boldsymbol{s}))_0 = l, g_0]$ is the sum of both probabilities. The distribution $P[(\texttt{abs}(\boldsymbol{c})^T\texttt{abs}(\boldsymbol{s}))_0 = l \,|\, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2]$ can be seen as a hypergeometric distribution $\texttt{hypergeom}(l, n, \|\boldsymbol{s}\|_2, \|\boldsymbol{c}\|_2)$.

The probability of a decryption failure is plotted for various error correction capabilities of the ECC in Figure 2. We can see that our new dependency aware model outputs a much better estimate of the probabilities of a certain maximum number of errors. Another observation to be made is that the independency based model deviates further from the experimental data as the number of errors increases, which is the case for codes with higher error correction capabilities. This makes the dependency issue especially important for schemes with extensive error correction.

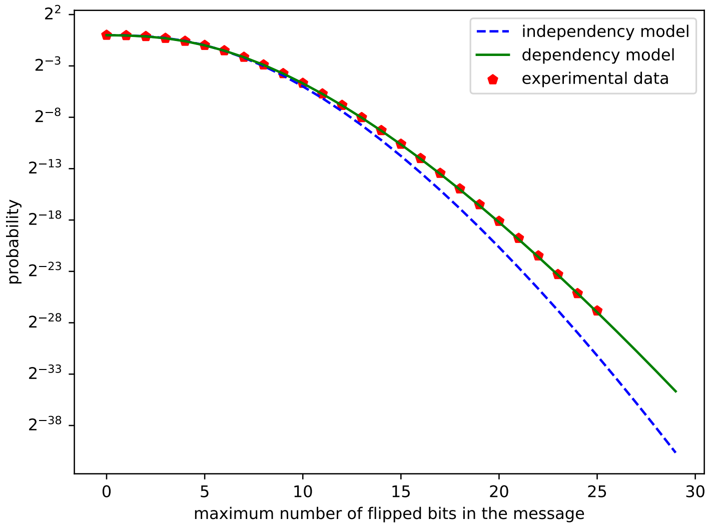

Figure 2: Probability of failure for various error correction capabilities of `ecc_enc`

4 Implications

As seen in previous sections, the errors in m'_{ecc} are positively correlated, meaning that an error at a certain position is more likely to happen if another error is present. The inverse is also true: a correct bit of m'_{ecc} enlarges the probability of other bits in m'_{ecc} to be correct. Therefore, due to the dependency, there will be more fully correct messages than one would expect under the assumption of independence. However, as one can see in Figure 2, the impact of the dependency is small for schemes without error correction. To conclude, an estimate using the assumption of independence will slightly overestimate the failure rate, and thus underestimate the security of the scheme with a small margin. As a result, the approximation using an assumption of independence is legitimate for schemes without an error correction step.

	LAC-128	LAC-256
Independency model	2^{-233}	2^{-114}
Dependency model	2^{-185}	2^{-92}
Overestimation factor	2^{48}	2^{22}

Table 2: The failure rate of different versions of LAC under the different models

In the case of schemes with error correction, one has to be more careful. As can be seen in Figure 2, the independence model gives an underestimation of the failure rate, which corresponds to an overestimation of the security of the scheme. This overestimation grows as d, the error correction capability of the ECC, becomes larger. In Table 2, the estimated failure rate of different versions of LAC is compared under both models. The discrepancy between both models reaches a factor 2^{48} in case of LAC-128. Therefore, the assumption of independence is not valid for schemes with error correction, and that it could lead to a serious overestimation of the security of the underlying algorithm.

More specifically, a higher failure probability suggests that the scheme might be more vulnerable to a decryption failure attack similar to the attack described by D'Anvers et al. [4], where the secrets are estimated statistically based on failing ciphertexts. Moreover, an attacker can reduce the failure probability by performing a precomputation for weak ciphertexts with higher failure probability. As LAC does not have any security against multi-target attacks that exploit decryption failures, this precomputation only needs to be performed once.

5 Conclusions

In this paper, we challenged the independency assumption of bit errors in messages encrypted with (Ring/Mod)-(LWE/LWR) based schemes. We showed both theoretically and experimentally that the occurrence of errors is positively correlated. Then we devised a method to calculate the failure rate of a scheme, taking into account the dependency of failures. Finally, we showed that the assumption of independence is appropriate for schemes without error correcting codes, but that it might lead to a substantial underestimation of the failure rate for schemes with error correcting codes. This underestimation attains a factor of 2^{48} for LAC-128. A higher-than-expected failure rate could have a serious impact on the security of the scheme through a decryption failure attack.

6 Acknowledgements

This work was supported in part by the Research Council KU Leuven grants: C16/15/058, C14/18/067 and STG/17/019 and by the European Commission through the Horizon 2020 research and innovation programme Cathedral ERC Advanced Grant 695305.

References

1. E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange – a New Hope. In *USENIX Security 2016*, 2016.
2. J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, and D. Stehlé. Crystals – kyber: a CCA-secure module-lattice-based kem. Cryptology ePrint Archive, Report 2017/634, 2017. http://eprint.iacr.org/2017/634.
3. J. D'Anvers, A. Karmakar, S. S. Roy, and F. Vercauteren. Saber: Module-lwr based key exchange, CPA-secure encryption and CCA-secure KEM. In *AFRICACRYPT 2018*, pages 282–305, 2018.
4. J.-P. D'Anvers, F. Vercauteren, and I. Verbauwhede. On the impact of decryption failures on the security of LWE/LWR based schemes. Cryptology ePrint Archive, Report 2018/1089, 2018. https://eprint.iacr.org/2018/1089.
5. S. Fluhrer. Cryptanalysis of ring-lwe based key exchange with key share reuse. Cryptology ePrint Archive, Report 2016/085, 2016. https://eprint.iacr.org/2016/085.
6. T. Fritzmann, T. Pöppelmann, and J. Sepulveda. Analysis of error-correcting codes for lattice-based key exchange. Cryptology ePrint Archive, Report 2018/150, 2018. https://eprint.iacr.org/2018/150.
7. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology*, 26(1):80–101, Jan 2013.
8. O. Garcia-Morchon, Z. Zhang, S. Bhattacharya, R. Rietman, L. Tolhuizen, and J.-L. Torre-Arce. Round2. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
9. M. Hamburg. Integer module lwe key exchange and encryption: The three bears - draft 8. https://www.shiftleft.org/papers/threebears/threebears-draft8.pdf, 2017.
10. M. Hamburg. Threebears. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
11. D. Hofheinz, K. Hövelmanns, and E. Kiltz. A modular analysis of the Fujisaki-Okamoto transformation. Cryptology ePrint Archive, Report 2017/604, 2017. http://eprint.iacr.org/2017/604.
12. H. Jiang, Z. Zhang, L. Chen, H. Wang, and Z. Ma. Post-quantum IND-CCA-secure KEM without additional hash. Cryptology ePrint Archive, Report 2017/1096, 2017. https://eprint.iacr.org/2017/1096.
13. Z. Jin and Y. Zhao. Optimal key consensus in presence of noise. Cryptology ePrint Archive, Report 2017/1058, 2017. https://eprint.iacr.org/2017/1058.
14. X. Lu, Y. Liu, D. Jia, H. Xue, J. He, and Z. Zhang. LAC. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.

15. M. Seo, J. H. Park, D. H. Lee, S. Kim, and S.-J. Lee. Emblem and R.Emblem. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.
16. N. P. Smart, M. R. Albrecht, Y. Lindell, E. Orsini, V. Osheter, K. Paterson, and G. Peer. LIMA. Technical report, National Institute of Standards and Technology, 2017. available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-1-submissions.