cryptography

MDPI

Article

Polar Codes for Module-LWE Public Key Encryption: The Case of Kyber

Iason Papadopoulos [1] and Jiabo Wang [2,*]

[1] Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK
[2] Strategic Center for Research in Privacy-Preserving Technologies & Systems, Nanyang Technological University, Singapore 639798, Singapore
* Correspondence: jiabo.wang@ntu.edu.sg

Abstract: In modern society, the Internet is one of the most used means of communication. Thus, secure information transfer is inevitably of major importance. Computers nowadays use encryption methods based on arithmetic operations to turn messages into ciphertexts that are practically impossible for an attacker to reverse-engineer using a classical computer. Lately, it has been proven that this is possible in a post-quantum setting where quantum computers of considerable size are available to attackers. With the advance of technology of quantum computers, it is now more necessary than ever before to construct encryption schemes that cannot be broken either using a classical or a quantum computer. The National Institute of Technology and Standards (NIST) has orchestrated a competition, and numerous encryption schemes have been proposed. The NIST has identified one algorithm to be standardized for the post-quantum era. This algorithm is called CRYSTALS-Kyber and is based on module learning with errors (MLWE). This paper investigates how to apply error correcting codes in order to create some excess decryption failure rate (DFR) and to take advantage of that in order to re-tune Kyber's parameters in the pursuit of higher security. By applying Polar Codes, Kyber's security was managed to be increased by 54.4% under a new set of parameters, while keeping the decryption failure rate well below the upper acceptable bound set by the NIST.

Keywords: Kyber; Polar Codes; public key encryption; module LWE; decryption failure rate; post-quantum security

Citation: Papadopoulos, I.; Wang, J. Polar Codes for Module-LWE Public Key Encryption: The Case of Kyber. *Cryptography* **2023**, *7*, 2. https://doi.org/10.3390/cryptography7010002

Academic Editor: Josef Pieprzyk

Received: 30 November 2022
Revised: 25 December 2022
Accepted: 30 December 2022
Published: 10 January 2023

1. Introduction

Cryptography refers to a communication technique of altering a message using mathematical concepts in a way that only allows the receiver to have access to its original content. Modern day cryptography aims to encrypt digital information. The idea behind encrypting schemes is related to a field of mathematics called "number theory" and more specifically "large number factoring" [1]. This is because it has been proven that computers cannot find the prime factors of a number efficiently, making these schemes secure and thus suitable for commercial use nowadays.

Over the last few decades, researchers have been trying to build quantum computers with limited success. Their work has motivated other scientists to investigate the capabilities of such computers. This has led Peter Shor, an American mathematician, to prove that number factorization becomes trivial for quantum computers using an algorithm that takes advantage of certain quantum phenomena, inevitably breaking current encryption mechanisms [2]. It is therefore necessary to develop an encryption technique that is unbreakable both for classical and quantum computers, and to migrate to it before the completion of the development of the first quantum computer. Otherwise, having access to a quantum computer would immediately imply having access to any encrypted information that exists online such as bank transfers, emails, passwords, and messages. The National Institute of Standards and Technology (NIST) has held a competition which aimed to identify and

standardize the best quantum-safe encryption scheme [3]. Many ideas were put on the table, but the winner of the competition was Kyber [4], which will be shortly standardized. This means that every online data transmission will be encrypted using Kyber. The following paper will be focused on how to increase Kyber's security using error correction.

Error correction refers to the ability to detect and correct errors that emerge when a message is transmitted through a noisy channel. There has been extensive research on this field, and researchers have developed many error correcting codes (ECC). The efficiency of such codes is measured with the Shannon capacity [5]. Polar Codes are a type of ECC, were introduced by Arıkan in [6], and have been proven to be as efficient as possible by reaching the theoretical maximum that was set by Shannon. This makes it very attractive for commercial use, and that is why these codes are used in new technologies such as the 5G. Moreover, Polar Codes can be implemented in quasilinear time complexity $O(N\log_2 N)$, giving them a comparative advantage over other ECCs such as low-density parity check (LDPC) or Bose–Chaudhuri–Hocquenghem (BCH) codes. Having such an outstanding performance, Polar Codes made the best candidate for this paper.

The idea is to alter the parameters of Kyber in order to achieve a more secure scheme. This increase in security comes at the cost of an increase in the decryption failure rate (DFR). Although this is straightforward, one must also take into consideration that the DFR has to be below the bound set by the NIST (2^{-128}) in order to be safe against decryption failure attacks [7]. That is why Polar Codes are introduced, in order to decrease the DFR and to take advantage of this excess DFR in order to be used to increase the security.

A similar approach has been taken for Ring LWE schemes in [8] where Polar Codes where used to increase the security of NewHope [9]. In that paper, the security of the specified scheme was increased by 9.4%. Moreover, in [10], BCH and LDPC codes as well as a combination of the two have been applied to NewHope, enhancing the security by 31.76%. It is mentioned that these types of ECC can be applied to Kyber as well. Although, at first glance, the latter method achieves better performance, it should be mentioned that the results are based on an "independence assumption", which is not proven in the paper. Finally, it should be mentioned that no previous research has focused on applying ECC to Kyber, and given that it is a matter of time until this scheme becomes the global standard for Public Key Encryption, it is sensible to explore how to make it more secure.

2. Preliminary

2.1. Kyber

As mentioned before, modern cryptography is based on arithmetic operations, which cannot guarantee security in the post quantum era. Scientists have turned their attention to Lattices and more specifically Lattice cryptography. Lattice cryptography is a field in mathematics that uses sets of points in the n-dimensional space in order to create difficult problems that can be used to encrypt information. One such problem that is widely used in post quantum cryptography is the learning with errors (LWE) problem introduced in [11]. It has been proven that the average case of an LWE problem is as difficult as its worst case, and its difficulty is based on several lattice problems such as the shortest vector problem (SVP) which has been proven to be hard. Moreover, the advantage that sets LWE apart from the rest when it comes to cryptosystems is that there are some variations such as the Ring-LWE (RLWE) and the Module-LWE (MLWE) that make LWE time- and space-efficient for real life applications while keeping the problems of the variations relatively difficult. Kyber is based on MLWE. Before explaining how Kyber works, it is useful to introduce the following notions:

R_q is the polynomial ring $\mathbb{Z}_q[X]/(X^n+1)$;
B_η is the binomial distribution: $Bi(2\eta, 0.5) - \eta$, centered around 0;
$\text{Decompress}_q(x,d) = \left\lceil \left(q/2^d\right) \cdot x \right\rfloor$;
$\text{Compress}_q(x,d) = \left\lceil \left(2^d/q\right) \cdot x \right\rfloor \bmod {}^+ 2^d$;
"$\leftarrow$" will be interpreted as "sampled from" .

Kyber's three steps (key generation, encryption, and decryption) for secure data transmission are as follows:

1. $\mathbf{A} \in R_q^{k \times k} \leftarrow B_{\eta 1}$, $\mathbf{s}$,$\mathbf{e} \in R_q^k \leftarrow B_{\eta 1}$,
   Public Key: $\mathbf{t} = \mathbf{A}\mathbf{s} + e$, Secret Key: $\mathbf{s}$;
2. $\mathbf{r} \in R_q^k \leftarrow B_{\eta 1}, e_1 \in R_q^k \leftarrow B_{\eta 2}, e_2 \in R_q \leftarrow B_{\eta 2}$,
   $\mathbf{u} = A^T\mathbf{r} + \mathbf{e}_1$, v = $\mathbf{t}^T\mathbf{r}$+e_2 + Decompress$_q(m, 1)$
   Transmit Compress$_q$($\mathbf{u}$, d$_u$), Compress$_q$(v,d$_v$);
3. m = Compress$_q$(Decompress$_q$(v, d$_v$) $-$ $\mathbf{s}^T$Decompress$_q$($\mathbf{u}$, 1),d$_u$).

All of these parameters determine Kyber's security. The parameters suggested by the development team are shown in Table 1:

Table 1. Kyber parameters.

	n	**k**	**q**	η_1	η_2	$\mathbf{d}_u$	$\mathbf{d}_v$
KYBER768	256	3	3329	2	2	10	4

2.2. Polar Codes

Now, let us introduce error correcting codes. In communications, channels are used to send data which might be corrupted during transmission due to the noisy nature of the channel. In order to prevent this from happening, one could encode the data that are sent in a clever way, such that in the case where a bit is inverted while transmitting, it would be obvious to the receiver and could thus be corrected. This error controlling mechanism is called error correction. To perform error correction, we use error correcting codes (ECC). There are several types of ECC, but the main idea behind all of them is that before transmitting the data, one encodes it using redundant information, which will then help the receiver detect and correct inverted (flawed) bits without needing to retransmit the whole message. Error correction is, thus, essential for efficient communication. The maximum performance limit of such codes was set by Shannon and can be achieved by a few ECCs. In this paper, we focus on one of them, namely Polar Codes, but similarly to our research, other ECCs are expected to achieve analogous results. The extent to which other ECCs can perform alongside Kyber is a field that could be further explored in the future.

A Polar Code is an error correction code that uses a linear block of length $N = 2^n$. It was introduced by Arikan and has been proven to achieve Shannon capacity for binary discrete memoryless symmetric (BDMS) channels. To measure the performance of Polar Codes, one can use mutual information and the Bhattacharyya parameter of BDMS channels. These two are defined as follows:

Definition 1. *Mutual information ($I(W)$) is a measure of the rate at which information can be transmitted. $I(W) \in [0, 1]$ and is given by the following equation:*

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y \mid x) \log \frac{W(y \mid x)}{\frac{1}{2}W(y \mid 0) + \frac{1}{2}W(y \mid 1)}$$

A larger $I(W)$ indicates a better transmission rate.

Definition 2. *The Bhattacharyya parameter ($Z(W)$) is a measure of reliability of the channel. $Z(W) \in [0, 1]$ and is given by the following equation:*

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y \mid 0)W(y \mid 1)}.$$

A smaller $Z(W)$ indicates a more reliable channel and is preferred.

The idea behind Polar Codes is to split a noisy channel into $N = 2^n$ sub-channels and to polarize them, resulting in some channels having $I(W) \approx 0$ and some others having $I(W) \approx 1$. Then, use the more reliable channels ($I(W) \approx 1$) in order to transmit a message without much interference. Then, decide how to decode the received bit (after the effect of noise), based on some likelihoods that will indicate the bit that was sent. The better the channel, the more obvious it would be to decode the received bits by observing the likelihoods.

For the base case, where $n = 1$, there are $N = 2$ channels, as shown below.

Suppose that there are two independent inputs Z_1 and Z_2, which are random variables of a discrete distribution:

The output y_i shown in Figure 1 of the channel W_2 is given by the following:

$$y_1 = u_1, y_2 = u_1 + u_2 \tag{1}$$

which, using matrices, can be written as follows:

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \mathbf{u}$$

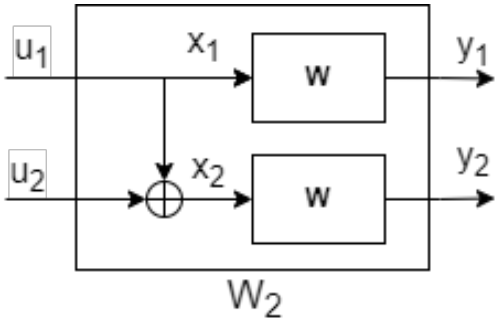


Figure 1. The channel W_2.

For a larger $n = 2$ ($N = 4$), one should expect the following matrix:

$$\mathbf{y} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \mathbf{u}$$

To achieve larger N, smaller channels are used recursively, resulting in the following formula.

$$\mathbf{y} = G_2^{\otimes n} \mathbf{u}$$

where $\otimes$ represents the Kronecker product defined as follows:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}$$

where $n = log_2 N$, N = number of channels.

As shown by Arıkan in [6], as $N \to \infty$, the N channels are split into two categories:

1. Channels where $I(W_N^{(i)}) \in (1 - \delta, 1]$;
2. Channels where $I(W_N^{(i)}) \in [0, \delta)$

for $\delta \in (0,1)$ and

$$\lim_{N\to\infty} \delta = 0$$

This proves that, for a larger N, a fraction of the channels are reliable ($I(W_N^{(i)}) \approx 1$) and the rest are unreliable ($I(W_N^{(i)}) \approx 0$). In the two cases introduced, for W_2, u_1 would be the reliable one and u_2 is the unreliable one. For W_4, u_1 and u_2 are the most reliable, whereas u_3 and u_4 are the least reliable. The list of channels sorted from most reliable to least reliable is called reliability sequence and has been found for every N experimentally through simulations [12], and it is thus beyond the scope of the paper to investigate any further.

2.3. Security against Side Channel Attacks

In this section, we prove why wrapping Kyber with Polar Codes is safe against side channel attacks (SCAs). As explained in [13,14], Polar Coding does not leak any sensitive information to the attacker. To prove this, we must look at encoding and decoding separately.

For encoding, it is sufficient to say that the number of logic XOR gates used to perform encoding is constant, independent of input, and is equal to $\frac{nlogn}{2}$, where n = block length.

Decoding needs a three part proof. Firstly, calculating the transition probabilities W does not leak sensitive information as exactly $nlogn$ floating point operations take place. Secondly, the time needed for the comparison between the likelihoods is only dependent on how close the two floating numbers that we are comparing are, but that does not give information about the output of the comparison. Finally, the amount of XOR operations used to decode is identical to this to perform encoding, and as explained before, it is independent of the plaintext input.

3. Materials and Methods

3.1. Kyber Analysis

As mentioned, the goal is to increase Kyber's security by adding Polar Codes. The idea is to encode the message before encrypting it and to decode it after decryption. This would look like the following flowchart:

With Figure 2 in mind, one can interpret the encryption–transmission–decryption (Kyber) as the channel over which the encoded message will be transmitted and can thus calculate its signal-to-noise ratio (SNR). To achieve this, Kyber must be treated as a fading channel in order to separate the information from the noise. The following derivation is used to perform this separation [4,15] and uses the equations explained in Section 2.1:

$$output = \mathbf{y} = v - \mathbf{s}^T\mathbf{u} \tag{2}$$

where $\mathbf{s}$,$\mathbf{u} \in R_q^k$ and $v \in R_q$.

$$\mathbf{t}^T = \mathbf{A}\mathbf{s} + \mathbf{e} \tag{3}$$

where $\mathbf{A} \in R_q^{kxk}$ and $s, e \in R_q^k$.

$$\mathbf{u} = \text{Decompress}_q\Big(\text{Compress}_q\Big(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1, d_u\Big), d_u\Big) = \mathbf{A}^T\mathbf{r} + \mathbf{e}_1 + \mathbf{c}_u \tag{4}$$

where $\mathbf{r}$, $\mathbf{e}_1$, $\mathbf{c}_u \in R_q^k$.

$$v = \text{Decompress}_q\Big(\text{Compress}_q\Big(\mathbf{t}^T\mathbf{r} + e_2 + \lceil q/2 \rfloor \cdot m, d_v\Big), d_v\Big) = \mathbf{t}^T\mathbf{r} + e_2 + \lceil q/2 \rfloor \cdot m + c_v \tag{5}$$

where $\mathbf{t}$,$\mathbf{r}$, $\mathbf{c}_u \in R_q^k$ and e_2, $c_v \in R_q$.

Using these new equations, one can obtain the following:

$$\mathbf{y} = v - \mathbf{s}^T\mathbf{u} = \lceil q/2 \rfloor \cdot m + \mathbf{e}^T\mathbf{r} - \mathbf{s}^T\mathbf{e}_1 + e_2 + c_v - \mathbf{s}^T\mathbf{c}_u \tag{6}$$

so the information term is $\lceil q/2 \rfloor \cdot m$ and the noise term is

$$N = \mathbf{e}^T \mathbf{r} + e_2 - \mathbf{s}^T \mathbf{e_1} + c_v - \mathbf{s}^T \mathbf{c}_u \tag{7}$$

Now, we need to analyse the noise term to find its distribution. To achieve this, the script `distributions.ipynb` has been developed (This script can be accessed at https://github.com/Jason-Papa/KyberPC/blob/main/distributions.ipynb (accessed on 22 June 2022)) which samples each term of N 100,000 times using in this case the parameters set for Kyber768 and produces the following plot. lad

Clearly, from Figure 3, one can conclude that, as the number of samples increase, the distribution of N can be approximated by a normal distribution. The mean and variance can be found in Table 2:

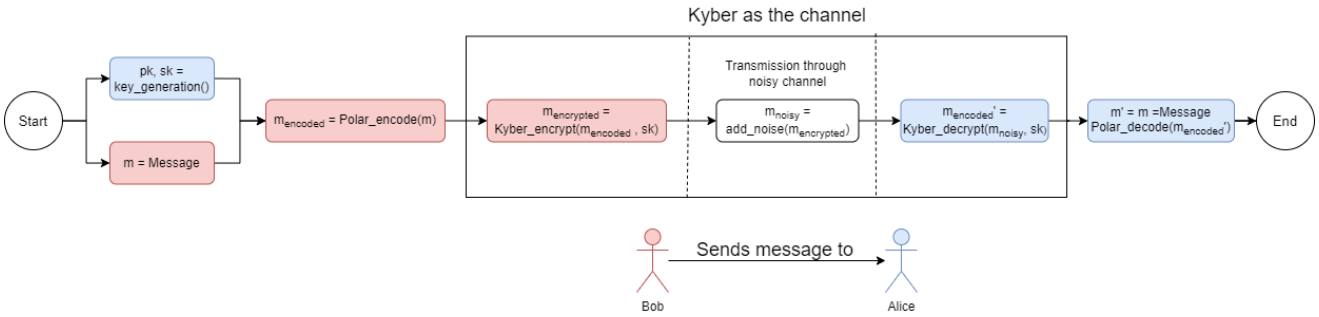
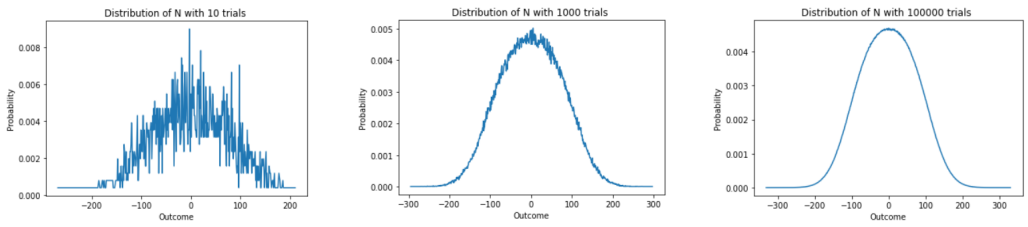


Figure 2. Kyber with Polar Codes (Kyber as a channel).



(**a**) 10 samples (**b**) 1000 samples (**c**) 100,000 samples

Figure 3. Experimental distribution of N_{Kyber} with Kyber768 parameters.

Table 2. Mean and variance of the N_{Kyber}.

# of Trials	Mean	Variance
10	0.1953	6068.48
100	0.3911	5859.28
1000	−0.06599	5838.79
10,000	0.01449	5856.93
50,000	0.0578	5854.60
100,000	0.01642	5855.87

From the results in Table 2 above, one observe that

$$\lim_{\#of trials \to \infty} \mu = 0, \sigma^2 = 5856$$

Moreover, using the Kyber equation derived before, one can define the signal-to-noise ratio as follows:

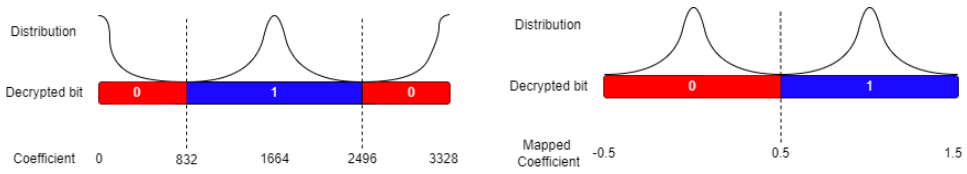
$$SNR_{Kyber} = \frac{q^2}{4\sigma^2} = \frac{3329^2}{4 \times 5856} \approx 473 = 26.75 \text{ dB} \tag{8}$$

3.2. Polar Code Selection

Now that the SNR of the channel has been calculated, we need to select the code rate (r = $\frac{N}{K}$, where N = block length and K = codeword length) of the Polar Code that will used. Since the input to Kyber encryption is fixed (n = 256), the only decision that has to be made is that of K. We now come against a trade-off between channel usage and performance of Polar Codes. A small value of K will lead to better performance as only the very reliable subchannels will be used for transmission, but on the other hand, this is not efficient. As K increases, less reliable subchannels will be used but the amount of unused channels will decrease. Another important consideration that has to be taken into account is that the scheme must be able to send 256 bits of useful information in every transmission. Since n = 256 and, by definition, adding any Error Correcting Code creates some redundant bits, we can see that this is not possible. To accommodate for this issue, the value of K selected was 128, and thus all 256 bits of useful information will be split into 2128-bit chunks and sent separately using the same key. That way, n remains the same, which results in the complexity of polynomial multiplications being the same as before while maximizing the error-correcting effect of Polar Codes.

3.3. Kyber–Polar Codes Compatibility

Now that we are ready to put everything together, we must make sure that Polar encoding is compatible with Kyber encryption and Kyber decryption is compatible with Polar decoding. For the former, the larger encoded message is broken down into 256-bit chunks and each one is encrypted individually. All chunks of the same encoded message can use the same samples, thus saving some computations (polynomial multiplications). For the latter, a more complex process must be followed. More specifically, the decrypted output of Kyber must update the Log-likelihoods. To achieve this, a mapping function was developed to map Kyber 0 s (which were mapped around 0 mod q) to 0 and Kyber 1 s (which were mapped around q/2 mod q) to 1. This is shown visually in Figure 4:



(**a**) Kyber Output (**b**) Mapped Output

Figure 4. Mapping algorithm.

This allows for the Log-likelihoods to be updated by multiplying the mapped output by a constant, which is based on the SNR and the code rate, allowing for Polar Codes to decode the bits. Now, we are ready to put everything together and to start experimenting with the security parameters while calculating the DFRs that emerge due to the new parameters used.

4. Results

In order to analyse the results, we must first understand how the changes in Kyber parameters affect them. First, security comes solely from the Kyber parameters as Polar Codes do not affect it at all but are used to ensure that the DFR is below the boundary. Moreover, increasing the security by changing some Kyber parameters comes at the cost of an increased DFR, thus reaching values that surpass the upper bound set by the NIST (2^{-128}). By applying Polar Codes to Kyber, we can calculate the block error rate (BLER), which in this case, is the same as the DFR of the scheme. To achieve this, we treat Kyber as the channel, and thus, Kyber's parameters correspond to a certain SNR value, which is used to calculate the upper bound of the DFR by finding the sum of the Bhattacharya parameters, as suggested in [6]. To alter the security, we first use the security parameter

k, as suggested by the CRYSTALS development team in [15]. The security estimates for different values of k when $\eta_1, \eta_2 = 2$ are presented in Table 3.

Table 3. Kyber with Polar Codes security estimates for different k.

k	SNR (dB)	DFR (Kyber Only)	BLER	Primal Attacks Classic/Quantum (bits)	Dual Attacks Classic/Quantum (bits)	Time per Transmission (s)
3	26.75	2^{-164}	2^{-1638}	182/165	181/164	0.419615
4	26.2	2^{-126}	2^{-1442}	256/232	253/230	0.4833
5	25.75	2^{-102}	2^{-1299}	332/301	327/297	0.6016
6	25.35	2^{-85}	2^{-1184}	409/371	403/365	0.6807
7	24.95	2^{-73}	2^{-1080}	487/442	479/434	0.8353
8	24.6	2^{-64}	2^{-995}	567/514	556/504	0.9253
10	23.95	2^{-51}	2^{-856}	727/660	715/650	1.1895

In Table 3, DFR refers to the decryption failure rate that the Kyber has without including Polar Codes, whereas BLER refers to the probability that a message is incorrectly received after decoding. This is the same as the DFR of the new scheme (Kyber with Polar Codes). The BLER estimates were obtained by the FERestimate variable in https://github.com/mcba1n/polar-codes/tree/master/polarcodes (accessed on 30 June 2022), which is the sum of the Bhattacharya parameters . The security estimates were obtained by the script provided by the CRYSTALS team, which is available at https://github.com/pq-crystals/security-estimates/blob/master/Kyber.py (accessed on 5 November 2022). In this script, there are some functions that find the best possible quantum attack for the given MLWE parameter set, both for primal and dual attacks, and return the security in bits. The same script was used by the CRYSTALS team to calculate Kyber's security in [4].

The values obtained for BLER are extremely small, but we need to keep in mind that, by viewing Kyber as the channel, we end up with a very large SNR, and given that the code rate is 0.5, meaning that we only use the best 50% of the subchannels created, we can expect that the probability of too many bits being inverted through the addition of Kyber's error terms (e_1, e_2, etc.), making Polar decoding unable to detect them, is extremely small. Note also that these are estimates, not exact values, and testing is only indicative of the scheme working, as finding a case where a message would fail to decode is practically impossible.

As we can see, increasing k leads to better security but, at the same time, the time per transmission is also increased. Although the software is not optimized and the time to complete every transmission is largely depending on the hardware used, the percentage increase in time is indicative of the number of polynomial multiplications increasing linearly as k increases. Moreover, a security above 300 quantum bits is considered overkill, thus suggesting that a value of k above 6 is not sensible, but is just given as an indication for possible future needs and will not be explored any further.

Even though the results obtained are satisfactory, we can achieve better by introducing a new method of increasing the security, that is, to change the binomial parameter η. This method's most important benefit is that security can be strengthen without the need for more calculations. In the specification, it is not recommended that η be used to enhance the security because there is a strong relationship between that and the DFR. This is because the variance of the noise term increases and, thus, decryption becomes more likely to fail. In our case, where Kyber is the channel, this increase in η is reflected by a decrease in SNR, but the increase in BLER does not exceed the bounds. It is thus possible to use it in combination with a higher k to achieve better results.

From Table 4, one can observe that, by increasing η, the classical bit security for primal attacks for the case where k = 4 can be increased from 256 to 281 bits (10% increase) while keeping the transmission time constant. Although theoretically, this idea can be extrapolated until the DFR hits the boundaries set by the NIST, it is safer to leave a generous

gap between the predicted value and that boundary. It is also not sensible to try to further increase the security as even the base case, which is 164 qubits secure, is considered safe for the time being.

Table 4. Kyber security parameters for different values of η and k.

k	η	SNR (dB)	DFR (Kyber Only)	BLER	Primal Attacks Classic/Quantum (bits)	Dual Attacks Classic/Quantum (bits)	Time per Transmission (s)
3	2	26.75	2^{-164}	2^{-1638}	182/165	181/164	0.4196
3	3	25.6	2^{-83}	2^{-1255}	193/175	191/174	0.4196
3	4	23.9	2^{-50}	2^{-847}	201/182	199/181	0.4196
4	2	26.2	2^{-126}	2^{-1442}	256/232	253/230	0.4833
4	3	24.8	2^{-63}	2^{-1043}	270/245	267/242	0.4833
4	4	23.0	2^{-37}	2^{-687}	281/254	278/252	0.4834
5	3	24.0	2^{-50}	2^{-866}	349/316	345/313	0.6016
5	4	22.3	2^{-29}	2^{-584}	362/328	359/325	0.6016
5	5	20.9	2^{-18}	2^{-421}	373/338	369/335	0.6016

In summary, in this section, we have shown that the introduction of Polar Codes around Kyber gives us the flexibility to use higher security parameters (k). The problem with that is that increasing k comes at the expense of more mathematical operations. To avoid this extra complexity, changing another parameter which was not mentioned by Kyber's development team, namely η, can lead to a security increase without an increase in time. We suggest a combination of these two methods and, thus, propose the following new parameters:

The parameters presented in Table 5 can obtain a classical bit security of 281, which compared to the initial security, which was 182, is a 54.4% increase in security.

Table 5. Suggested Kyber parameters with Polar Codes.

	n	**k**	**q**	η_1	η_2	$\mathbf{d}_u$	$\mathbf{d}_v$
KYBER-PC	256	4	3329	4	4	10	4

5. Discussion

Now that the results have been presented, it is time to talk about their influence to the future of post-quantum cryptography. Although quantum computers that are reliable and capable of breaking modern day encryption schemes are still under development, it is never too early to develop tools that give us flexibility in terms of increasing security. This paper has thoroughly presented one such tool and opens the way for further investigation on the effect of other error correcting codes such as low-density parity checks (LDPC) and BCH codes. It is also straightforward to apply this technique to other MLWE cryptosystems by leaving out the error terms in the noise that arise from compressing and decompressing values.

Moreover, this research motivates researchers to develop hardware-efficient implementation of Kyber with Polar Codes that could be beneficial for real-time communication. Similar to [16,17], where FPGAs were used to improve Kyber key generation, encryption and decryption time, one can implement an FPGA solution that performs Polar encoding and decoding steps as well, in order to get the best of both worlds.

It should also be noted that, if one wants to decrease the key size of Kyber, our findings can be used to keep the security at a safe level by altering the parameters appropriately.

Since this field is relatively new, there has not been any previous research that focuses on Kyber. A similar approach to post-quantum encryption was taken in [8] where Polar Codes were applied to RLWE schemes with NewHope [9] in mind. The results on that research are compliant with those found here.

Before concluding this paper, it is essential to discuss the limitations of the proposed solution. As shown by the results, increasing the value of the security parameter k as a negative effect in time per transmission, which is not desirable for real life communications. Moreover, this increase in transmission time will be further increased by the Polar encoding and decoding steps that have been added. In order though to draw more accurate solutions, software/hardware-optimised versions of the suggested solution must be implemented to be compared with plain Kyber.

6. Conclusions

To conclude, it has been shown that error correction can be proven to be beneficial when wrapping a Kyber-encrypted transmission as it can lower the decryption failure rate. One can thus take advantage of the new, smaller DFR to increase security parameters and to achieve higher security. Under our suggested values for the Kyber's parameters, the security can be increased by 54.4% when compared to Kyber768, while keeping DFR below the acceptable boundary. We also gave motivation to interested researchers for relevant topics to be explored.

Author Contributions: Conceptualization, I.P. and J.W.; methodology, I.P.; software, I.P.; validation, I.P. and J.W.; formal analysis, I.P.; investigation, I.P.; resources, I.P.; data curation, I.P.; writing—original draft preparation, I.P.; writing—review and editing, I.P. and J.W.; supervision, J.W.; project administration, I.P. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by Imperial College London.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available at https://github.com/Jason-Papa/KyberPC (accessed on 22 June 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RSA	Rivest–Shamir–Adleman
NIST	National Institute of Standards and Technology
ECC	Error Correcting Code
LWE	Learning With Error
RLWE	Ring-Learning With Error
MLWE	Module-Learning With Error
BLER	Block Error Rate
DFR	Decryption Failure Rate
SCA	Side Channel Attacks

References

1. Rivest, R.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
2. Shor, P. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. [CrossRef]
3. Csrc.nist.gov. Post-Quantum Cryptography | CSRC. 2022. Available online: https://csrc.nist.gov/projects/post-quantum-cryptography (accessed on 16 July 2022).
4. Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS—Kyber: A CCA-Secure Module-Lattice-Based KEM. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 24–26 April 2018.
5. Shannon, C. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]

6. Arikan, E. Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels. *IEEE Trans. Inf. Theory* **2009**, *55*, 305–3073. [CrossRef]
7. D'Anvers, J.P.; Batsleer, S. *Multitarget Decryption Failure Attacks and Their Application to Saber and Kyber*; Cryptology ePrint Archive, Paper 2021/193; Springer: Cham, Switzerland, 2021.
8. Wang, J.; Ling, C. How to Construct Polar Codes for Ring-LWE-Based Public Key Encryption. *Entropy* **2021**, *23*, 938. [CrossRef] [PubMed]
9. Alkim, E.; Ducas, L.; Pöppelmann, T.; Schwabe, P. Post-quantum key exchange—A new hope. In Proceedings of the 25th USENIX Security Symposium (USENIX Security 16), Austin, TX, USA, 10–12 August 2016; pp. 327–343.
10. Fritzmann, T.; Pöppelmann, T.; Sepulveda, J. Analysis of Error-Correcting Codes for Lattice-Based Key Exchange. In *International Conference on Selected Areas in Cryptography—SAC 2018*; Springer: Cham, Switzerland, 2019; pp. 369–390.
11. Regev, O. The Learning with Errors Problem; ACM: New York, NY, USA, 2005.
12. Bioglio, V.; Condo, C. Design of Polar Codes in 5G New Radio. *IEEE Commun. Surv. Tutor.* **2018**, *23*, 29–40. [CrossRef]
13. Wang, J.; Ling, C. Polar coding for Ring-LWE-based public key encryption. *Cryptogr. Commun.* **2022**, 1–35. [CrossRef]
14. Howe, J.; Prest, T.; Ricosset, T.; Rossi, M. *Isochronous Gaussian Sampling: From Inception to Implementation*; Post-Quantum Cryptography; Ding, J., Tillich, J.P., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 53–71.
15. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.; Schwabe, P.; Seiler, G.; Stehlé, D. Algorithm Specifications and Supporting Documentation, Version 3.01. 2022. Available online: https://pq-crystals.org/kyber/data/kyber-specification-round3-20210131.pdf (accessed on 17 July 2022).
16. Bisheh-Niasar, M.; Azarderakhsh, R.; Mozaffari-Kermani, M. High-Speed NTT-based Polynomial Multiplication Accelerator for Post-Quantum Cryptography. In Proceedings of the IEEE 28th Symposium on Computer Arithmetic (ARITH), Lyngby, Denmark, 14–16 June 2021; pp. 94–101.
17. Bisheh-Niasar, M.; Azarderakhsh, R.; Mozaffari-Kermani, M. A Monolithic Hardware Implementation of Kyber: Comparing Apples to Apples in PQC Candidates. In *International Conference on Cryptology and Information Security in Latin America*; Springer: Cham, Switzerland, 2021.