Estimating the strength of horizontal correlation attacks in the hamming weight leakage model: A side-channel analysis on HQC KEM

Guillaume Goy, Antoine Loiseau, Phlippe Gaborit

▶ **To cite this version:**

Guillaume Goy, Antoine Loiseau, Phlippe Gaborit. Estimating the strength of horizontal correlation attacks in the hamming weight leakage model: A side-channel analysis on HQC KEM. WCC 2022: The Twelfth International Workshop on Coding and Cryptography, Mar 2022, Rostock, Germany. WCC_2022_paper_48. cea-04176652

HAL Id: cea-04176652
https://cea.hal.science/cea-04176652v1

Submitted on 19 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Estimating the Strength of Horizontal Correlation Attacks in the Hamming Weight Leakage Model: A Side-Channel Analysis on HQC KEM

Guillaume Goy[1,2], Antoine Loiseau[1], and Philippe Gaborit[2]

[1] Univ. Grenoble Alpes, CEA, Leti, MINATEC Campus, F-38054 Grenoble, France
{guillaume.goy,antoine.loiseau}@cea.fr
[2] XLIM, University of Limoges
gaborit@unilim.fr

Abstract. Hamming Quasi-cyclic (HQC) is an alternate candidate at the NIST call for proposal. Before standardization, the NIST expects studies about Side-Channel Attacks (SCA) for the third round schemes. This paper introduces an horizontal attack against the Reed-Solomon (RS) decoding algorithm of HQC assuming an Hamming weight leakage model. The aim of the attack is to recover the RS codeword generated directly related to the exchanged message with no more knowledge than public codes and parameters of the scheme. This attack targets the syndrome computation of the RS decoder where an error-free codeword is manipulated. The public RS decoder algorithm allows to correct the erroneous inferred bytes from SCA. By using list decoding algorithms, we increase the error correction capability and the strength of the attack. Then we give an analysis based on simulated traces. In practice, experiments show that the measured signal to noise ratio is sufficiently low for a successful attack against the reference implementation of HQC128 by running 2^{96} operations over a Galois field. Finally, we present a random shuffling countermeasure in order to prevent this horizontal attack.

Keywords: HQC · Side-Channel Analysis · Simulation · Error Correcting Codes · Horizontal Attack · Post-Quantum Cryptography

1 Introduction

The main deployed asymmetric cryptographic schemes are based on the difficulty of solving number theory problems such as large integer factorization for RSA [14] or finding a discrete logarithm for the elliptic curve cryptography [9,10]. However, these problems can be solved in polynomial time by the Shor's algorithm on a quantum computer [16]. For this reason, the American National Institute of Standards and Technology (NIST) launched a call for proposal in December 2016 with the aim to standardize quantum resistant signature schemes and Key-Encapsulation Mechanisms (KEM) [12].

Hamming-Quasi-Cyclic [2] (HQC) is a code-based alternate candidate in the third round of the competition. Thus, HQC might be standardized in addition

to the finalists in a fourth round. In addition to specification schemes, the NIST expects studies that address Side-Channel Attacks (SCA) against candidates [11].

In 1996, Kocher introduced SCA, consisting in an analysis of Side-Shannel (SC) leakages (such as power consumption, electromagnetic emanations...) in order to retrieve manipulated secret data. SCA uses the statistical dependence between measured traces and intermediate variables of cryptographic computation. They represent a threat for the security of cryptographic schemes. Afterwards, horizontal attacks were presented by Walter [18] at CHES 2001, against the RSA exponentiation looking for the secret exponent. This kind of attack exploits the dependency between several intermediate results to discriminate secret data using a single trace. Furthermore, Clavier et al. [5] went further by showing that the horizontal information from a single trace can be exploited by a correlation analysis. Their attack computes the correlation coefficient on several trace segments extracted from a single trace.

The state of the art about SCA on HQC only mentions four attacks, 3 Timing Attacks (TA) and a template attack. In 2019 and 2020, two TAs [17,13] used a correlation between the weight of the decoded error and the computation time of the decoder algorithm of BCH codes. Considering this threat, designers developed an improved constant time BCH decoder without a significant performance penalty. In addition, in 2020, a chosen ciphertext template attack [15] used a classification of traces resulting from a BCH decoding process to determine whenever an error is corrected or not. This method, combined with some linear algebra, allows to recover the private key.

However, the last version of the HQC KEM is no longer built with BCH codes. Since October 2020, new version of HQC uses a concatenated Reed-Muller (RM) and Reed-Solomon (RS) code in order to reduce keys size. In 2021, a new TA [6] uses the non-constant time rejection sampling during the re-encryption of the HQC RMRS KEM. Authors show that the observed timing variation is dependent on the secret key allowing them to fully recover it with high probability. To the best of our knowledge, this last timing attack is the only attack targeting the HQC RMRS KEM.

Our Contribution. In this paper we present the first horizontal SCA against the HQC RMRS KEM targeting the RS decoder and assuming an Hamming Weight (HW) leakage model with aims to recover the exchanged message. Each byte of the codeword is manipulated several times during the RS decoding step of the decapsulation which allow to deduce these byte values using correlation analysis. In HQC, the decoder is public, we can used it to correct the erroneous side-channel deductions. Since this decoder has a limited error correction capability, we go further by using better decoders for RS codes in order to increase the success rate of the attack. In this paper, with aim to determine the limits and use case of our attack, we analyze the leakage with simulated traces. This construction allows to get a full control over noise, and give analysis depending on the Signal to Noise Ratio (SNR). We compare these simulated results

with practical results and show that a real attack is possible to fully recover the exchanged message with less than 2^{96} Galois field operations.

Paper Organization. The paper is organized as follows: Section 2 recalls some mathematical backgrounds and the HQC Framework. Section 3 gives a description of the attack, focusing on the horizontal correlation analysis. Section 4 addresses our practical result and compare them with simulated results. Section 5 presents improvements of the attack using list decoding algorithms and listing strategy. Section 6 is about the complexity of the attack. And finally Section 7 refers to a countermeasure.

2 Background

2.1 Mathematical Background

Signal to Noise Ratio (SNR). A SC trace is related to an intermediate manipulated value which determine the class (the label) of the trace. SNR is a tool that permits to quantify the noise within labelled traces, given by the equation (1). The SNR decreases when the noise increases and a SNR greater than 1 indicates more signal than noise.

$$\text{snr} = \frac{Var(E(X|Z))}{E(Var(X|Z))} \tag{1}$$

Hamming Weight (HW) Leakage Model (LM). A leakage model describes the behaviour of the SC traces depending on their classes. We say that a set of traces follow an HW LM if each trace can be described by the equation (2):

$$\begin{aligned} &tr : \mathbb{Z} \to \mathbb{R} \\ &tr : z \to a \times HW(z) + b + \mathcal{N}(0, \sigma) \end{aligned} \tag{2}$$

where a, b are parameters of the leakage to be determined and $HW(z)$ the hamming weight of a cryptographic intermediate value z. These traces are also affected by a measure noise $\mathcal{N}(0, \sigma)$, which may be due to various factors, which can be represented with a zero centered normal distribution.

Pearson Correlation Coefficient. This is a tool to determine the correlation between two random variables X and Y (see equation (3)). In our case, we calculate the correlation between traces and intermediate values.

$$\rho_{\mathbf{X},\mathbf{Y}} = \frac{\text{cov}(\mathbf{X}, \mathbf{Y})}{\sigma_{\mathbf{X}} \sigma_{\mathbf{Y}}} = \frac{\text{E}\left[(\mathbf{X} - \mu_{\mathbf{X}})(\mathbf{Y} - \mu_{\mathbf{Y}})\right]}{\sqrt{\text{E}\left[(\mathbf{X} - \mu_{\mathbf{X}})^2\right] \text{E}\left[(\mathbf{Y} - \mu_{\mathbf{Y}})^2\right]}} \tag{3}$$

where μ_X and $\sigma_{\mathbf{X}}$ are, respectively, the measured expected value and the measured variance of $\mathbf{X}$.

The equation (3) returns a value between -1 and 1 for element of Y. The closer the value of $|\rho_{\mathbf{X},\mathbf{Y}}|$ is to 1, the stronger the correlation between the two variables is. The elements are sorted depending on their correlation values.

2.2 Hamming Quasi-Cyclic (HQC)

HQC [2] is a post-quantum resistant KEM based on error-correcting code. A classical construction is to turn a PKE into a KEM. For HQC, a quantum adapted FO transformation called Hofheinz-Hövelmanns-Kiltz transformation [7] is used. Details and algorithms are given in HQC specifications [2]. The framework of HQC KEM is summarized with Figure 1. The exchanged message is protected by a random error with a significant weight which is too big to be decoded by the public decoder. The knowledge of the secret key allows to reduce the weight of the error which falls below the error correction capability of the code. The exchanged message can be decoded and a shared key is derived from it.

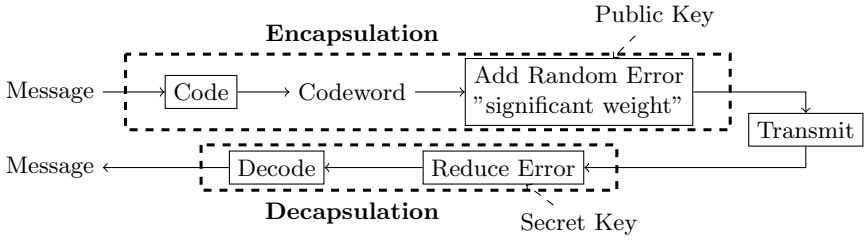

Fig. 1: HQC KEM Framework

HQC uses a concatenated code with a duplicated Reed-Muller (RM) code for internal code and a Reed-Solomon (RS) code for the external one. The decoder is then a double decoder that decodes the internal code first and then the external. An analysis of the Decryption Failure Rate (DFR) in the HQC specification [2] (section 2.5.8) shows that the RS codeword, between the two decoders is error-free with a probability $1 - 2^{-10.96}$ for HQC128.

Reed-Solomon Decoder. Let $\mathbf{r}$ be the received message, input of the RS decoder. The RS decoder is a minimum distance decoding algorithm that works by finding the roots of an Error Locator Polynomial (ELP). ELP is computed by solving a system of equations depending on the syndromes of the received message $\mathbf{r}$. We focus on the algorithm that computes the syndromes of the received message. Considering the received message as a polynomial, the syndrome computing is the evaluation of this polynomial for all the roots of the RS generator polynomial α_i. In HQC, this procedure is given by Algorithm 1 and the roots α_i

are parameters of HQC (see section 2.5.2 of specifications [2]). $\delta = 15$, $n_1 = 46$ are also two parameters of HQC128 that describe the size of the parity check matrix.

Algorithm 1 COMPUTE_SYNDROMES

Input: (received message $\mathbf{r}$, param $= (\alpha_i, \delta, n_1)$)
Output: syndrome **synd**

1: **for** $i = 0 \rightarrow 2 \times \delta$ **do**
2: **for** $j = 1 \rightarrow n_1$ **do**
3: $\mathbf{synd}[i] = \mathbf{synd}[i] \oplus \text{GF\_MUL}(\mathbf{r}[j], \alpha_i^{j-1})$
4: $\mathbf{synd}[i] = \mathbf{synd}[i] \oplus \mathbf{r}[0]$
5: **return synd**

3 Attack Description

In this attack, the target is the input of the RS decoder which is also the output of the RM decoder. The target is independent from the secret key. Indeed, when the RS decoder is applied, the intermediate codeword is no longer hidden by the secret key (see Figure 1). For a sake a clarity and given the low DFR of the RM code, we will consider that this intermediate codeword is error-free and we will refer to it as "the codeword" for the rest of the paper. If we are able to recover the codeword, we can deduce the exchanged message of the KEM using the public RS decoder.

The goal is to retrieve the 46 bytes of a RS codeword $\mathbf{c}$. Each byte is involved in $2 \times \delta = 30$ multiplications in a Galois Field (GF) with powers of α_i (see Algorithm 1). Our analysis considers the output of the GF_MUL function as intermediates values. The attack is developed thanks to two observations:

1. Each byte of the received message is manipulated 30 times which gives information redundancy allowing an horizontal attack. Given one measured trace of the decoding step, one has in fact 30 informative sub-traces for each target byte.
2. We use of the knowledge of the decoder. This last point is the most crucial, it allows to cope with eventual attack errors. Indeed, if the number of erroneous recovered bytes is smaller than the error correction capability of the RS decoder, the original message is recovered by applying the RS decoder.

Correlation Analysis. Our study is based on an horizontal correlation analysis [4] assuming an HW LM (see equation (2)).

In this attack, we recover the bytes of the codeword one by one. With this method, we can have information about only 45 bytes of the codeword. Indeed, the first byte $\mathbf{r}[0]$ is computed in such a way that we cannot get information

about it with the same attack method (see Algorithm 1). For this byte, we have no choice but to choose it uniformly at random. In other words, we proceed 45 different attacks and we finally concatenate the results and obtain 46 bytes length word. The goal is to determine from which hypothesis the traces are closer given the HW LM. We can describe the behaviour of the traces for each key hypothesis. This is done by computing the 45 correlation matrices (see equation (4)), corresponding to all the possible powers of α. These matrices described what the traces would be following a perfect HW LM without noise. Matrices are of size number of hypothesis * number of targeted operations, in our case $(256, 30)$.

$$Mat\_Cor_k = HW(\text{GF\_MUL}(i, \alpha_j^k)), i \in [0, 255], j \in [1, 30], k \in [1, 45] \quad (4)$$

Finally, the correlation value is computed using the Pearson Correlation Coefficient (see equation (3)). The bytes are sorted depending on their correlation values and we retained the one with the highest one. Among the 46 finalist bytes returned by the attack, some of them are wrong, because of mistakes induced by the attack. Then the public RS decoder is applied to correct these errors. The study of the number of occurred errors must be carefully considered in order to stay below the error correction capability of the chosen RS decoder. We stress that the number of erroneous bytes in the final result depends on the noise within the traces. We will categorize this noise within the next section.

4 Results

Practical Results. We used the reference implementation of HQC from June 2021 [1] on a ARM Cortex-M4 microcontroller with a clock frequency of 168 MHz. We measured the electromagnetic emanations with a LANGER EMV-Technik near field microprobe ICR HH 100-6. We registered these traces with an oscilloscope ROHDE&SCHWARZ RTO2014 with a sampling rate of 10G samples per second. A dedicated GPIO pin of the microcontroller was used to trigger the oscilloscope. Two other UART pins were used for the communication between the microcontroller and the computer during the acquisitions. We acquired a set of 256 thousand EM traces of the GF_MUL execution. For the best points of interest, we obtained a SNR value between 0.77 and 1.50 with an average value of 1.15.

Simulated Results. A set of simulated traces can be generated given the equation (2), by selecting the GF_MUL function outputs as intermediate values. With no loss of generality, the parameters are selected to be $a = 1$ and $b = 0$. For this paper, we generated more than 60 thousand sets of $30 * 45$ traces with different σ noise parameter values. For each set of traces, the SNR value is computed given the equation (1). Then, we proceeded with the correlation analysis on theses traces. On the Figure 2, we plot the average percentage of recovered bytes (in first position of the correlation list returned) depending on the SNR of the set of traces considered. The decoder allows to correct 16 errors within

the retrieved word. In average, this attack is a success for SNR greater than 2.35 that is the limit above which the error correcting capability is sufficiently high to correct the SCA errors. The average SNR observed with measured traces is also plotted with red dashed line. Given this observed SNR value and the error correcting capability of the classical RS decoder, the attack cannot succeed with a single trace. In the following sections, we present improvements that allow a successful attack.

5 Attack Improvements

5.1 List Decoding Algorithms

RS codes can be decoded using a list decoding algorithm. This kind of decoder provides a list of l closest words from the received message. With these decoders, the error correcting capability increases. There are two known list decoders, respectively the Sudan (S) and the Gurusami-Sudan (GS) decoder [8]. It is known that the last one can decode asymptotically up to $n - \sqrt{k \cdot n}$ errors instead of $\frac{n-k}{2}$ errors for a classical decoder (see Figure 3). Given the HQC128 parameters, $n = 46, k = 16$, for a good choice of parameters, GS decoder is able to decode up to 19 errors instead of 15 for the classical decoder. This first improvement reduces the SNR boundary imposed by our former analysis.

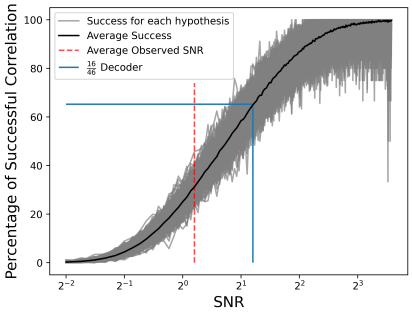

Fig. 2: Average percentage of successful (first position within list) correlation analysis depending on the value of the SNR for each set of traces.

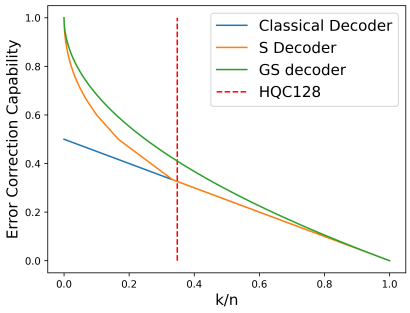

Fig. 3: Comparison of error correcting capabilities for three Reed-Solomon decoders depending on the rate of the code. In red the rate for the HQC128 Reed-Solomon code.

In this section, we assume that we have an erroneous received message $\mathbf{r} = \mathbf{c} + \mathbf{e}$ of length n, generated by a RS codeword $\mathbf{c}$ and an error $\mathbf{e}$ of HW τ. We also assume that the error follows an uniform distribution among the n coordinates of the codeword. We note that RS decoders return the location and value of the errors before returning the corresponding message.

5.2 Increasing the Decoder by listing erroneous locations

The first idea to decrease the number of errors to be corrected by the decoder by assuming u erroneous locations are known in the codeword. By removing these locations, one has to decode a $[n-u, k]$ RS code with $\tau - u$ errors. If $\tau - u$ is lower than the error correction capability, we end up with a n length codeword with the u remaining errors which can be decoded by another decoding step. Knowing the locations of u errors is not free. The probability of having u erroneous locations, assuming u randomly sampled locations is given by $\left(\frac{\tau}{n}\right)^u$. In average, one has to make $T = \left(\frac{n}{\tau}\right)^u$ draws to succeed. For each draw, two decoders are computed, which gives the complexity of this first method.

5.3 Increasing the Decoder by listing error-free locations

The second idea is to assume that u error-free locations are known within the codeword. We used the shortened RS code construction, by truncated the u coordinates, we end up with a $[n-u, k-u]$ RS code with τ errors. The number of errors to decodes remains the same, but the ratio $\frac{k}{n}$ decreases, which allows a better error correction capability given by the Figure 3. After finding these errors, one has to correct them from the initial codeword to retrieve the message. The average number of draw to find these u locations is given by $T = \left(\frac{n}{n-\tau}\right)^u$. Figure 4 summarizes the required number of draws to be able to decode τ errors with the GS decoder. The two methods need the same number of draw on average, but the second method seems better given that only one decoding step is required to end the attack, instead of two for the first one.

5.4 Looking at the x first candidates returned by the Correlation

This last idea is to consider several candidates instead of one as output of the correlation step. The last step of the attack must be proceeded for each retained candidates. Figure 5 summarizes the average number of successful attacks depending on the SNR and on the number of retained candidates. This strategy can be very costly, indeed for x considered candidates, the number of required decoding step increases by a factor x^{46}.

6 Attack Complexity

Considering an average SNR of 1.15 and the top two candidates returned by the correlation. The average percentage of correct bytes found is more than 46%. This strategy returns 2^{46} candidates to test. On the other hand, by assuming known $u = 13$ error-free locations, we can exhibit a decoder able to decode 25 bytes out of the 46. It represents around 54% of the total amount of bytes, for a cost of 2^{15} decoding steps. We decode enough bytes to retrieve the error-free codeword and succeeding in the attack. The decoder has a maximum complexity of $\mathcal{O}(n^7)$ operations over the field [3], with n the length of the code. We apply

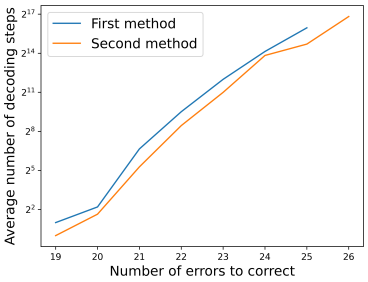

Fig. 4: Average number of required decoding step to be able to correct τ errors

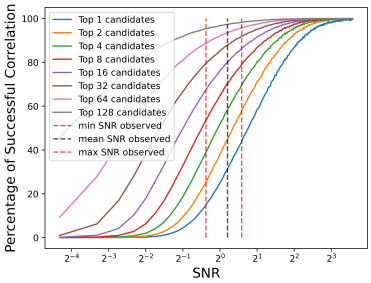

Fig. 5: Average percentage of successful correlation analysis depending on the value of the SNR and on the number of candidates returned.

the decoder on words of size $n - u = 46 - 13$, which corresponds to less than 2^{35} operations in our case. The complexity of our attack is then given by $2^{46} \times 2^{15} \times 2^{35} = 2^{96}$ operations over the Galois field.

7 Countermeasure

We propose here one countermeasure that prevents our attack from succeeding without significantly increasing the total computation time of the decoding process. Our attack is based on the labelization of EM traces for GF_MUL computation. This is why the countermeasure consists in randomizing the order of execution within the COMPUTE_SYNDROME computation. This idea was first proposed in 2010 by Clavier et al. [5] to counter horizontal attack against RSA exponentiation. Random permutation raises to $(2 \cdot \delta)! \cdot n_1! \approx 2^{294}$ the number of possibilities. Under this assumption, it becomes really difficult to label traces.

8 Conclusion

In this paper, we have presented an analysis of an horizontal attack against the HQC KEM assuming an hamming weight leakage model. The aim was to recover the intermediate codeword between internal and external decoding, which lead to the knowledge of the shared key. We exploit the low decryption failure rate of internal code in HQC specifications. This DFR implies an error-free manipulation of the exchanged message by the RS decoder. We computed a correlation attack on several segments extracted from a single trace. A Reed-Solomon list decoding algorithm was used to correct the wrongly inferred bytes from this attack. Combined with a listing strategy of erroneous locations of the SCA output, we were able to significantly increase the error correction capability and therefore the strength of the attack. With simulated traces, we showed that the

measured noise in a real acquisition is sufficiently low to succeed the attack. Under current conditions, the attack require 2^{96} operations over the Galois field for a successful attack. We exhibit a leakage that can be exploit to recover the shared key of the HQC KEM with less than 2^{128} that is the claimed security of the scheme. Finally, we present a random shuffling countermeasure that prevent this attack from succeeding.

9 Acknowledgements

We would like to thank our colleagues, J. Maillard, V. Cristiani, M. Lecomte, from CEA Grenoble, who provided insight and expertise that greatly assisted the research. This work was supported by the French National Agency in the framework of the "*Investissements d'avenir*" (future-oriented investments) program (ANR-10-AIRT-05) and by the defense innovation agency (AID) from the french ministry of armed forces.

References

1. Aguilar-Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Persichetti, E., Zémor, G.: HQC reference implementation, https://pqc-hqc.org/implementation.html
2. Aguilar-Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Persichetti, E., Zémor, G.: Hamming quasi-cyclic (hqc) (2017)
3. Barbier, M.: Décodage en liste et application à la sécurité de l'information. Ph.D. thesis, Ecole Polytechnique X (2011)
4. Brier, E., Clavier, C., Olivier, F.: Correlation power analysis with a leakage model. In: International workshop on cryptographic hardware and embedded systems. pp. 16–29. Springer (2004)
5. Clavier, C., Feix, B., Gagnerot, G., Roussellet, M., Verneuil, V.: Horizontal correlation analysis on exponentiation. In: International Conference on Information and Communications Security. pp. 46–61. Springer (2010)
6. Hlauschek, C., Lahr, N., Schröder, R.L.: On the timing leakage of the deterministic re-encryption in hqc kem. Cryptology ePrint Archive, Report 2021/1485 (2021), https://ia.cr/2021/1485
7. Hofheinz, D., Hövelmanns, K., Kiltz, E.: A modular analysis of the fujisaki-okamoto transformation. In: Kalai, Y., Reyzin, L. (eds.) Theory of Cryptography. pp. 341–371. Springer International Publishing, Cham (2017)
8. Justesen, J., Høholdt, T.: A course in error-correcting codes, vol. 1. European Mathematical Society (2004)
9. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of computation **48**(177), 203–209 (1987)
10. Miller, V.S.: Use of elliptic curves in cryptography. In: Conference on the theory and application of cryptographic techniques. pp. 417–426. Springer (1985)
11. Moody, D., Alagic, G., Apon, D.C., Cooper, D.A., Dang, Q.H., Kelsey, J.M., Liu, Y.K., Miller, C.A., Peralta, R.C., Perlner, R.A., et al.: Status report on the second round of the nist post-quantum cryptography standardization process (2020)

12. NIST: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. . (2016)
13. Paiva, T.B., Terada, R.: A timing attack on the hqc encryption scheme. In: International Conference on Selected Areas in Cryptography. pp. 551–573. Springer (2019)
14. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
15. Schamberger, T., Renner, J., Sigl, G., Wachter-Zeh, A.: A power side-channel attack on the cca2-secure hqc kem. In: 19th Smart Card Research and Advanced Application Conference (CARDIS2020) (2020)
16. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
17. Wafo-Tapa, G., Bettaieb, S., Bidoux, L., Gaborit, P., Marcatel, E.: A practicable timing attack against hqc and its countermeasure. Advances in Mathematics of Communications (2020)
18. Walter, C.D.: Sliding windows succumbs to big mac attack. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 286–299. Springer (2001)