A Side channel attack methodology applied to Code-Based Post Quantum Cryptography

A Side Channel Attack methodology applied to Code-based Post Quantum Cryptography

Kristjane Koleci, Lorenzo Cecchetti, Guido Masera, Maurizio Martina, and Massimo Ruo Roch

DET, Politecnico di Torino,
Corso Duca Degli Abruzzi 24,Torino, Italy
{kristjane.koleci,lorenzo.cecchetti,
guido.masera,maurizio.martina,massimo.ruoroch}@polito.it

Abstract. The present work proposes a Side Channel Attack that targets the multiplier of a code-based Post Quantum Cryptography primitive. The Secret Key has been recovered with the Correlation Power Analysis obtained with the use of a methodology that simulates the power consumption profile of a design and then validates the method with the real device. The methodology is proposed as a useful tool to study weaknesses of designs during their design phase.

Keywords: Post-Quantum Cryptography, Side Channel Attack, VLSI, FPGA

1 Introduction

The security of Public Key Cryptography (PKC) is based on the hardness of recovering the Secret Key (SK) from the Public Key (PK), this is considered infeasible for classical computers, but quantum computers can bridge the gap and put the security of current cryptographic systems at risk. Post Quantum Cryptography (PQC) algorithms are one solution. The primitives have been proposed to a competition launched by the National Institute of Standards and Technology (NIST) in 2016 [1].

The adoption of NP-hard problems is not enough to guarantee the security of the whole system. Indeed, it is important to analyze possible information leakages of either the device where the algorithm is running or the hardware implementation of the primitives. Despite a Side Channel Attack has been successfully applied to a PQC primitive running on a device [2], the case of hardware implementation of such primitives has not been studied yet.

The present work analyzes the power consumption profile of a multiplier employed in PQC primitives. In Section 2, LEDAcrypt/BIKE [3], [4] code-based cryptosystems and the multiplier are described together with the model of the attack. In Section 3, the attack and experimental methodology are described. Finally, in Section 4, the results of the validation are reported and the conclusions are drawn.

2 Architecture for Code Based Cryptosystem

Code-based cryptosystems are based on the idea originally proposed by McEliece [5]: the encryption adds redundancy to the message to hide ($mathbfm$), with Public Key (PK) $\mathbf{G}$, and a error pattern $\mathbf{e}$ to obtain the ciphertext ($\mathbf{x}$), while the decryption removes the errors with the Secret Key (SK) $\mathbf{H}$. In the present work, the focus is on LEDAcrypt/BIKE cryptosystems, where codes are Quasi-Cyclic Low/Moderate Density Parity Check (QC-LDPC/MDPC) codes, as a consequence, $\mathbf{H}$, the SK, is a Quasi Cyclic matrix.

A QC binary matrix is a block matrix with n_0 cyclic blocks and size p ranging from 10^4 to 10^5. In cyclic matrices only the first row (or column) is required to completely describe the matrix, as each row is obtained by cyclic-shifting the first one. The SK is: $\mathbf{H} = [\mathbf{H_0}, \mathbf{H_1}, \ldots, \mathbf{H_{n_0-1}}]$. Moreover, the blocks are sparse and the number of 1s is referred to as d_v, which value is ≈ 100; these 1s are the target of the attack. The encoding and decoding algorithms require to calculate the product between a vector and a cyclic matrix LEDAcrypt [3][4]. As a consequence, any multiplier at the decoder side, being in charge of computing the product between a vector and the $\mathbf{H}$ matrix (the SK), is a critical block. Indeed, fundamental information about the positions of the ones can be derived by monitoring the power trace of the multiplier.

The multiplier proposed in [6] is referred to as `Vector By Circulant` (VbC). It takes as inputs a vector ($\mathbf{v}$) and a set of positions ($\mathbf{P}_v$) (the asserted 1s in the first row of the matrix) to compute the multiplication as a sum of cyclic shifts of the input vector, v_s (partial products). The shift amount is read from $\mathbf{P}_v$ as shown in Algorithm 1. This is provided as `Position` in the multiplier unit of Figure 1. According to Algorithm 1, the module generates one partial product v_s per iteration. The single v_s is generated n_b bits per clock cycles with the `Rotate unit`. The new v_s is summed to the previous result with the `Element wise sum`.

Algorithm 1 `VbC`

Input: $\mathbf{v}(1,n)$,$\mathbf{P}_v(1,d_v)$;
Output: $\mathbf{r}(1,n)$;
$\mathbf{r} \Leftarrow \mathbf{0}$;
for $i_P = 0$ **to** d_v **do**
  $k = \mathbf{P}_v(i_P)$;
  $\mathbf{v}_s = [\mathbf{v}(k:n), \mathbf{v}(1:k-1)]$;
  $\mathbf{r} \Leftarrow \mathbf{r} + \mathbf{v}_s$
end for

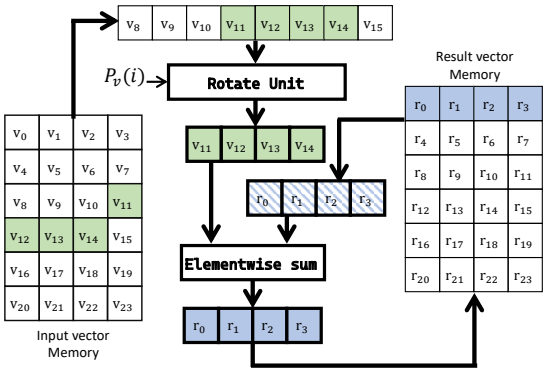

Fig. 1. VbC with $n_b = 4$.

The described multiplier is then analyzed in order to find its weaknesses. In general, the target of Side-Channel Attacks are the steps of the algorithms that employs the Secret Key (SK) to evaluate an intermediate variable, this is the case of the Syndrome ($\mathbf{s}$) which is the result in Algorithm 1, $\mathbf{H}$ is the set of position, $\mathbf{x}$ is the input vector.

3 Attack Methodology

The present work analyzes the dynamic power consumption to predict possible information leakage of the architecture. The study presents an initial analysis of the multiplier by relying on simulations, with an approach similar to the one adopted in [7]. Then, results are validated on an FPGA device.

The Correlation Power Analysis (CPA) attack described in [8] is applied to the dynamic power consumption (P_{dyn}) of the multiplier in the code-based decoder. The idea is to retrieve the Secret Key (SK), i.e. the decoding matrix $\mathbf{H}$, by exploiting the leakages in the computation of the Syndrome, $\mathbf{s}$.

The attack iteratively guesses the asserted positions in the first row of $\mathbf{H}$. The number of elements to search depends on the type of code that is employed in the algorithm, and for BIKE[4]/LEDAcrypt[3] is in the order of 10^2.

The model used to predict P_{dyn} is the Hamming Distance (HD) [8], then the *intermediate values* of the method are selected, two consecutive sequences of $\mathbf{r}_i$ after the `Elementwise sum` unit, and targeted by the attack.

3.1 Power Traces Derivation

The Power Traces, required by the CPA attack, have been obtained via post-synthesis simulation of the multiplier netlist, during the computation of the quasi-cyclic multiplication to evaluate the Syndrome $\mathbf{s}$.

The netlist has been obtained with Synopsys Design Compiler using the UMC CMOS 65 nm technology. Then, with a SK and a random message as the inputs, the decoding has been simulated with Questa Sim by Mentor, to derive the information on the switching activity of the design. Finally, the single power trace has been generated with Synopsys PrimeTime. An example of power trace is shown in Figure 2.

The process has been automatized through a Python script to launch the tool with random inputs and to generate the desired number of traces.

The correlation traces have been computed from the power traces and power prediction with the detailed description presented in [8] for all the asserted positions in the first row of the matrix (d_v). The example of a single correlation trace is shown in Figure 3. The analysis clearly shows the presence of a peak: this is the correlation peak that identifies the clock cycle in which the transition between the two intermediate values occurs.

In detail, the *intermediate values* for the HD power model are computed, for this purpose two matrices are required, IV_1 and IV_2, which contains all the possible consecutive intermediate values we can have, the number of rows of the

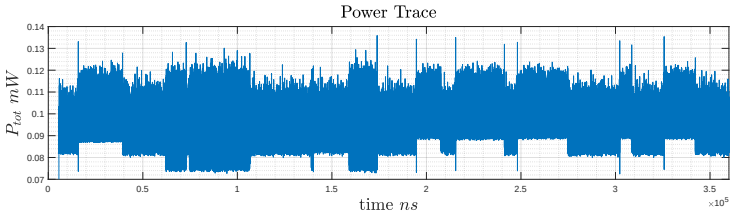

Fig. 2. Example of a Power Trace derived with Synopsys Prime time during the execution of a complete product.

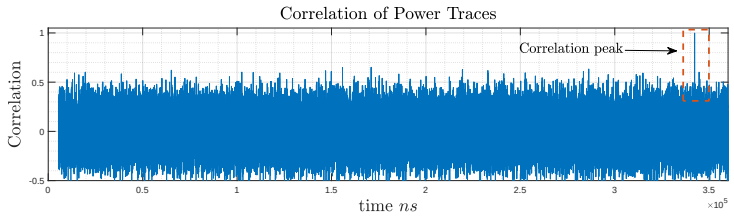

Fig. 3. Correct Correlation trace of the position 11 in d_v in SK. The correlation peak is highlighted in the dashed box.

matrices corresponds to all possible input of the `Rotate Unit` and the number of columns to the possible $\mathbf{P_v}$ selection of the unit. The hamming distance of IV_1 and IV_2 is evaluated (HD model) to derive the modeled power matrix **PM**. The measured power traces (for a fixed value of the $\mathbf{P}_v$) are collected to derive matrix **PT**, with the number of rows that corresponds to the number of traces and the columns that corresponds to the number samples. The information in **PM** and **PT** are combined with the Pearson Correlation Coefficient to derive the Correlation Matrix **CP**, in our case the formula becomes:

$$cp_{i,j} = \frac{\sum_{n=1}^{N}(pm_{n,i} - \bar{pm}_i) \cdot (pt_{n,j} - \bar{pt}_j)}{\sqrt{\sum_{n=1}^{N}(pm_{n,i} - \bar{pm}_i)^2 \cdot \sum_{n=1}^{N}(pt_{n,j} - \bar{pt}_j)^2}} \tag{1}$$

The i, j lower case variables represent the i, j value in the corresponding upper case matrix.

3.2 Secret Key Recovery

The CPA attack is structured as an algorithm that iteratively recovers one position in $\mathbf{P}_v$. In general, the candidate values for the position coming out from the CPA algorithm are associated to correlation traces that show a peak very close to 1. However, multiple candidate positions for each iteration may be found, due to the rotations in `VbC`, depending on the position we are trying to guess. In particular, this is the case of the first position in $\mathbf{P}_v$. In the first position the

correct value has to be computed from the first peak that appears in the considered range of time. The guess on the second position is simpler, because only one correlation trace presents a peak, thus there is only one possible candidate. On the contrary, the third position has the same behaviour as the first one. In this case, we can have two possible scenarios: three candidates or two candidates. In the second case, the candidates are $\mathbf{P}_v(2)$ and $\mathbf{P}_v(3)$. The candidates are three if $\mathbf{P}_v(2) - \mathbf{P}_v(1)$ is a multiple of n_b, with $\mathbf{P}_v(1) < \mathbf{P}_v(2)$. In this case the candidates are $\mathbf{P}_v(1)$, $\mathbf{P}_v(2)$ and $\mathbf{P}_v(3)$, then the position is found by exclusion.

In all the other cases, the $i-th$ (with $4 \leq i \leq d_v$) position approximated by the CPA attack corresponds to a choice between $\mathbf{P}_v(i)$ and $\mathbf{P}_v(i-1)$, and again the correct position if found by exclusion.

In the end, the attack is successful and the asserted positions in SK are discovered; this makes the `VbC` multiplier not secure against such attacks.

3.3 Threshold frequency limit detection

A study on the operating frequency has been carried out in order to find the limit of the CPA attack presented in the previous paragraphs. In particular, we want to find the minimum operating frequency for which no correlation peak can be detected.

The study on this limit is motivated by the fact that the P_{dyn} of the correlation traces strongly depends on the operating frequency. If the dynamic power has a contribution too negligible to the static power, it is not possible to compute a reliable correlation value. The results shown in Figure 4 have been obtained guessing the second value of the key and represent the behaviour of the correlation peak of the correct key as a function of the operating frequency.

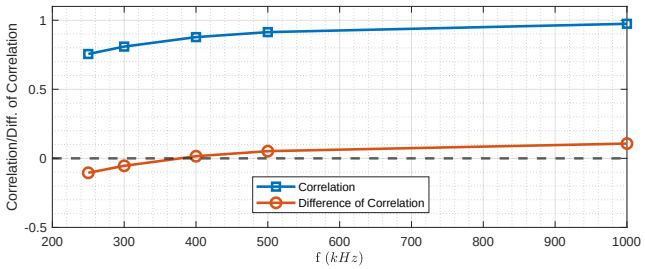

Fig. 4. Graph representing the behaviour of the correlation peak of the correct key value and an indication on the accuracy of the attack depending on the working frequency of the multiplier.

The blue curve represents the highest correlation value found in the correct key and it is directly proportional to the frequency, as expected. The orange curve shows the difference between the correlation peak in the correct key and the highest correlation value found from all the wrong key values. If the difference

is positive, the attack was successful. On the contrary, if the difference is negative it means that the highest correlation value belongs to one of the incorrect keys. As it can be observed, the threshold is around 400 kHz but since the data were obtained through simulations, it is expected that in a real case this value would be higher.

4 Results validation and conclusions

The same CPA attack, in order validate the results obtained via simulations for the ASIC implementation, has been conducted on a real implementation of the multiplier. Despite the resulting value of the power is different, the results of the power model remains unchanged and the difference is in the value of the collected power traces.

To this purpose, the VirtLab board [9][10], equipped with a Cyclone 10 LP 10CL025YE144C8G FPGA and an STM32L496VET6 microcontroller, was used. The multiplier architecture has been implemented on the FPGA and the microcontroller is used to stimulate the circuit and collect the measures, working as a Digital Storage Oscilloscope (DSO). The maximum DSO sampling frequency is 500 kHz, thus limiting the working frequency of the multiplier to 250 kHz in order to record two samples per clock cycle. The attack applied un such a lower frequency tourned out to be unsuccessful since SK has not been guessed. One proof of it could be the maximum working frequency analysis performed in 3.3, the value of 250 kHz is lower than the threshold we found. Moreover, the Static Power has been examined from the report of the tool employed to program the FPGA and it showed that the P_{static} was comparable to the measured P_{tot}, then the result of the correlation showed no peak since the comparison was among to uncorrelated quantities.

The proposed method of analysis is useful to prevent the security issues of an architecture during the design phase by identifying the effects of a CPA attack on a PQC multipler. The method can be applied to the whole decoder and for different types of Side-channel attacks. The future work will be dedicated to make the architecture immune to this type of attacks.

References

[1] https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization.

[2] Mélissa Rossi et al. *A Side-Channel Assisted Cryptanalytic Attack Against QcBits*. Cryptology ePrint Archive, Paper 2017/596. `https://eprint.iacr.org/2017/596`. 2017. URL: `https://eprint.iacr.org/2017/596`.

[3] M.Baldi et al. *LEDAcrypt Home*. URL: `https://www.ledacrypt.org/`.

[4] https://bikesuite.org/.

[5] Robert J. McEliece. "A Public-Key Cryptosystem Based on Algebraic Coding Theory". In: *DSN Progress Report* 44 (1978), pp. 114–116.

[6] Kristjane Koleci et al. “Efficient Hardware Implementation of the LEDAcrypt Decoder”. In: *IEEE Access* 9 (2021), pp. 66223–66240. DOI: 10.1109/ACCESS.2021.3076245.

[7] Alessandro Barenghi et al. “On the Efficiency of Design Time Evaluation of the Resistance to Power Attacks”. In: *2011 14th Euromicro Conference on Digital System Design*. 2011, pp. 777–785. DOI: 10.1109/DSD.2011.103.

[8] Eric Brier, Christophe Clavier, and Francis Olivier. “Correlation Power Analysis with a Leakage Model”. In: *Cryptographic Hardware and Embedded Systems - CHES 2004*. Ed. by Marc Joye and Jean-Jacques Quisquater. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29. ISBN: 978-3-540-28632-5.

[9] Massimo Ruo Roch and Maurizio Martina. “vrLab: A Virtual and Remote Low Cost Electronics Lab Platform”. In: *Applications in Electronics Pervading Industry, Environment and Society*. Ed. by Sergio Saponara and Alessandro De Gloria. Cham: Springer International Publishing, 2021, pp. 213–220. ISBN: 978-3-030-66729-0.

[10] Massimo Ruo Roch and Maurizio Martina. “VirtLAB: A Low-Cost Platform for Electronics Lab Experiments”. In: *Sensors* 22.13 (June 2022), p. 4840. ISSN: 1424-8220. DOI: 10.3390/s22134840. URL: http://dx.doi.org/10.3390/s22134840.