Side-channel attack on Rainbow post-quantum signature

David Pokorný, Petr Socha, Martin Novotný
Czech Technical University in Prague, Faculty of Information Technology, Czech Republic
{pokord11,petr.socha,martin.novotny}@fit.cvut.cz

***Abstract*—Rainbow, a layered multivariate quadratic digital signature, is a candidate for standardization in a competition-like process organized by NIST. In this paper, we present a CPA side-channel attack on the submitted 32-bit reference implementation. We evaluate the attack on an STM32F3 ARM microcontroller, successfully revealing the full private key. Furthermore, we propose a simple masking scheme with minimum overhead.**

***Index Terms*—post-quantum cryptography, digital signature, multivariate quadratic, side-channel analysis, embedded systems**

I. Introduction

Most currently used standards for asymmetric cryptography are based on factorization or solving of a discrete logarithm [1], i.e., problems solvable in polynomial time using Shor's algorithm [2] on a quantum computer. One of the current NIST's standardization candidates for a post-quantum digital signature is Rainbow [3], a layered generalization of the Unbalanced Oil and Vinegar scheme [4]. The scheme is based on a multivariate quadratic system of equations over a finite field; the general problem of solving a set of quadratic equations is NP-hard [5]. The winning standards should allow efficient implementations in embedded systems to consider the upcoming IoT era. Moreover, cryptographic implementations in embedded environment are known to be vulnerable to side-channel attacks [6], [7]. A Correlation Power Analysis (CPA) attack on a naïve 8-bit implementation of Rainbow is presented in [8].

In this work, we propose a CPA attack on the 32-bit reference implementation of Rainbow from the NIST competition's second round, which has been submitted as a candidate for NIST post-quantum digital signature standardization and is currently in the third round. We propose a way to extract the private key from the device and evaluate our attack on an STM32F3 ARM microcontroller. Finally, we propose a simple masking countermeasure.

II. Preliminaries

A. Rainbow

Let $\mathbb{F}$ be a finite field and $v_1, v_2, \dots, v_u, v_{u+1} \in \mathbb{N}$ be $u+1$ parameters, such that $0 < v_1 < v_2 < \dots < v_u < v_{u+1} = n$. Define two sets for each layer, where a layer is indexed by i:

$$\forall i \in \{1, 2, \dots, u\} : V_i = \{1, 2, \dots, v_i\}, |V_i| = v_i, \tag{1}$$

$$O_i = \{v_i + 1, \dots, v_{i+1}\}, |O_i| = o_i. \tag{2}$$

Both sets contain indices of variables. V_i contains indices of vinegar variables in i-th layer, and O_i contains indices of its oil variables. Using these parameters, define a set of $m = n - v_1$ multivariate quadratic polynomials with n variables called central map F, structured to u layers.

Central map $F = (f^{\ (v_1+1)}, f^{\ (v_1+2)}, \dots, f^{\ (n)})$ is a set of multivariate quadratic polynomials with

$$f^{\ (k)}(x_1, \dots, x_n) \coloneqq \sum_{\substack{i,j \in V_l \\ i \leq j}} \alpha_{i,j}^{(k)} x_i x_j + \sum_{\substack{i \in V_l \\ j \in O_l}} \beta_{i,j}^{(k)} x_i x_j, \tag{3}$$

where $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)} \in \mathbb{F}$ are quadratic coefficients (other quadratic coefficients are set to zero) and $l \in \{1, \dots, u\}$ such that $k \in O_l$. In this definition, linear and absolute coefficients are omitted for simplicity as the reference implementation does not use them and therefore they do not affect the attack. With this structure of quadratic polynomials, F^{-1} can be solved using a linear equation solver [3].

In the public key, oil and vinegar variables are mixed to hide the unique structure of quadratic polynomials. The mixing process is realized using two affine maps, $S : \mathbb{F}^m \rightarrow \mathbb{F}^m, T : \mathbb{F}^n \rightarrow \mathbb{F}^n$. Again, the translations are omitted for simplicity, so two linear maps represented by matrices $S \in \mathbb{F}^{m \times m}$ and $T \in \mathbb{F}^{n \times n}$ are considered.

The private key is then defined as $SK \coloneqq (S^{-1}, F, T^{-1})$ and the public key as $PK \coloneqq (P)$, where $P = S \circ F \circ T : \mathbb{F}^n \rightarrow \mathbb{F}^m$.

B. Reference implementation

Three variants of the Rainbow signature scheme are proposed in the NIST competition, and their reference implementations are available. The Cyclic variant is motivated by Petzoldt's cyclic Rainbow scheme [9], the Compressed variant stores the private key in the form of a 512-bit seed, and the Classic variant stores plain matrices. We will discuss only the Classic variant Ia[1] with parameters selected to fit NIST security categories I and II.

This two-layered ($u = 2$) variant uses $m = 64$ quadratic polynomials with $n = 96$ variables over $\mathbb{F} = GF(2^{2^2}) = GF(16)$. Layers are structured as $(v_1, v_2, v_3) = \ (32, 64, 96)$

[1] Classic variant Ia from the second round of NIST competition, i.e., $\mathbb{F} = GF(16), (v_1, o_1, o_2) = (32, 32, 32)$. The third round candidate uses $(v_1, o_1, o_2) = (36, 32, 32)$, requiring only marginal alterations of our work.

and $(o_1, o_2) = (32, 32)$. This variant uses matrices S and T of the form

$$S = S^{-1} = \begin{pmatrix} \mathbb{I} & S' \\ \mathbb{O} & \mathbb{I} \end{pmatrix}, \tag{4}$$

$$T = \begin{pmatrix} \mathbb{I} & T^{(1)} & T^{(2)} \\ \mathbb{O} & \mathbb{I} & T^{(3)} \\ \mathbb{O} & \mathbb{O} & \mathbb{I} \end{pmatrix}, \; T^{-1} = \begin{pmatrix} \mathbb{I} & T^{(1)} & T^{(4)} \\ \mathbb{O} & \mathbb{I} & T^{(3)} \\ \mathbb{O} & \mathbb{O} & \mathbb{I} \end{pmatrix}, \tag{5}$$

where $S \in \mathbb{F}^{64\times 64}$, $T \in \mathbb{F}^{96\times 96}$, their submatrices are elements of $\mathbb{F}^{32\times 32}$, $\mathbb{O}$ is zero matrix, $\mathbb{I}$ is identity matrix, and $T^{(4)} := T^{(1)} \cdot T^{(3)} - T^{(2)}$. The storage of the central map is not relevant to our attack. In the following text, we denote $\widetilde{x}, y \in \mathbb{F}^{64}$ and $x, \widetilde{y} \in \mathbb{F}^{96}$, where $\widetilde{x} = S^{-1} \cdot y$ and $x = T^{-1} \cdot \widetilde{y}$

Each element of $GF(2^{2^2})$ is identified by 4 bits, while the considered reference implementation uses a 32-bit word. A suitable Galois field was selected so that multiplication of a vector by one element can be performed using simple bit operations, allowing for word-level data parallelism. Consequently, a vector of eight elements can be multiplied by one element using only a few instructions.

E.g., consider a computation of $\widetilde{x} = S^{-1} \cdot y$. First, the product $S' \cdot y_{33:64}$ is computed in column-wise order. Using word-level parallelism, each matrix column is processed as four vectors of eight elements (i.e., the external loop iterates across all columns, and the internal loop iterates across four vectors). Finally, the matrix-vector product $\widetilde{x} = S^{-1} \cdot y$ is obtained by addition of the y vector, to take the identities $\mathbb{I}$ in S^{-1} into account: $\widetilde{x}_{1:32} = S' \cdot y_{33:64} + y_{1:32}$, $\widetilde{x}_{33:64} = y_{33:64}$.

III. Attack

A. Correlation power analysis

Correlation Power Analysis [6], [7] is a side-channel attack allowing extraction of secret information from the cryptographic device. First, the adversary observes a physical variable, such as device power consumption during the cryptographic operation execution. The adversary then makes her guess on the key (or an enumerable part of it), and correlates her hypothetical consumption predictions with the observed physical variable.

B. Idea of attack

To explain the attack, let us examine the signing process.

Process of signing: For document d, random salt s and secret key $SK = (S^{-1}, F, T^{-1})$, we define the signature as a pair (x, s) where

$$y := \text{hash}\left(\text{hash}\left(d\right) || s\right), \tag{6}$$

$$x := T^{-1}\left(F^{-1}\left(S^{-1}(y)\right)\right). \tag{7}$$

The S^{-1} map is applied first, followed by the inverse of central map F, and finally applying the T^{-1} map. Vinegar variables for the first layer are generated randomly at the beginning of the algorithm.

With matrices S^{-1} and T^{-1} known, we can reveal the central map F easily with knowledge of public key $PK = (P)$. Therefore, we aim our attack at the linear parts S and T only. Note that $S = S^{-1}$ and T^{-1} can be computed from T and vice versa. We cannot choose the input of the S matrix multiplication directly due to salting, but we can compute its value as we supply d and know s from the resulting signature.

C. Attack on S map

In the first signing step, a matrix-vector product $\widetilde{x} = S^{-1} \cdot y$ is computed (detailed in subsection II-B). Our attack is aimed at the computation of $\widetilde{x}_{1:32} = S' \cdot y_{33:64} + y_{1:32}$, where y is a known vector and S' is a part of the private key.

1) Attack I: Our CPA attack therefore is row-oriented. Each element i of the final product can be expressed as

$$\widetilde{x}_i = \sum_{j=1}^{32} (S'_{i,j} \cdot y_{j+32}) + y_i. \tag{8}$$

In the reference implementation, vector $\widetilde{x}_i$ is initialized with zeroes, then y is multiplied with S' iteratively, and finally y_i is added due to the identity submatrices in Equation 4. This is one of the differences compared to [8], where y_i is added first, making their attack substantially easier to mount. Our predictions are based on a Hamming weight of the intermediate sum value for $j \in \{2, \ldots, 32\}$. Table I summarizes intermediate values for *Attack I.*

TABLE I
Attack I: Revealing a matrix row.

Target	Intermediate value
$S'_{i,1}$ and $S'_{i,2}$	$S'_{i,1} \cdot y_{33} + S'_{i,2} \cdot y_{34}$
$S'_{i,3}$	$\sum_{j=1}^{2}(S'_{i,j} \cdot y_{j+32}) + S'_{i,3} \cdot y_{35}$
$S'_{i,4}$	$\sum_{j=1}^{3}(S'_{i,j} \cdot y_{j+32}) + S'_{i,4} \cdot y_{36}$
$\vdots$	$\vdots$
$S'_{i,32}$	$\sum_{j=1}^{31}(S'_{i,j} \cdot y_{j+32}) + S'_{i,32} \cdot y_{64}$

In the first step, we target both $S'_{i,1}$ and $S'_{i,2}$ subkeys. Since the attacked 4-bit subkeys can have 16 different values, and there are 32 subkeys in the first matrix column, targeting only $S'_{i,1}$ would not lead to a useful solution. Using these predictions, multiple subkeys are found in the first step since the targeted intermediate value does not correspond to a unique input value. To resolve this, we further process these subkeys independently.

Another problem arises if $S'_{i,2} = 0$. The power predictions would then be the same as targeting only $S'_{i,1}$:

$$S'_{i,1} \cdot y_{33} + S'_{i,2} \cdot y_{34} = S'_{i,1} \cdot y_{33} + 0 \cdot y_{34} = S'_{i,1} \cdot y_{33}. \tag{9}$$

The same problem occurs for each zero element in the row. This problem can be overcome for columns with index $k > 2$. Knowing $S'_{i,j}, j \in \{1, \ldots, k-1\}$, our attack considers only non-zero values, i.e., $S_{i,k} \in \{1, \ldots, 15\}$. If no significant correlation with any of these 15 hypotheses is found, the element is assumed to be a zero.

Furthermore, each row is multiplied by the same vector $y_{33:64}$, and therefore, even if we find the whole row, we are not able to directly distinguish the row's index. We obtain $S'_{i,1:32}$ for some $i \in \{1, \ldots, 32\}$.

2) Attack II: After revealing elements of the entire row, the row index must be further identified. This is accomplished by *Attack II*, targeting the final addition of vector $y_{1:32}$. The used power predictions are described in Table II. Using *Attack I* and *Attack II*, we are able to reveal 28 rows[2] out of 32 on average.

TABLE II
ATTACK II: ROW IDENTIFICATION.

Target	Intermediate value
$k \in \{1, \ldots, 32\}$ in y_k	$\sum_{j=1}^{32}(S'_{i,j} \cdot y_{j+32}) + y_k$

3) Attack III: The last step is revealing the remaining rows. To do so, we exploit the word-level parallelism described in subsection II-B. All the matrix rows are partitioned into sets based on this parallelism, i.e., the rows that are processed together are in their respective sets. For each row we attack, the subkey hypotheses are considered together with the other (already known) rows in the set. The power predictions are then based on a Hamming weight of the whole word. We define the aforementioned partitions and a hypothesis H_j^l for every column j and a certain set $l \in \{1, 2, 3, 4\}$ as

$$Set_l \coloneqq \{8 \cdot (l-1) + 1, \ldots, 8 \cdot l\}, \tag{10}$$

$$H_j^l \coloneqq \sum_{i \in Set_l} \mathrm{HW}(\sum_{k=1}^{j}(S'_{i,k} \cdot y_{k+32})). \tag{11}$$

Considering leakage hypotheses of the entire set allows for a more precise prediction based on the processed word. For simplicity, we consider zeroes instead of the unknown elements in S'.

Attack III for i-th row, where l is such that $i \in Set_l$, uses power predictions described in Table III. The predictions are based on a sum of the most probable hypotheses for the other rows determined by previous attacks, and on a hypothesis for the i-th row. In case there are multiple missing rows in the set, the row index must be identified. We accomplish this using the last power prediction in Table III. The attack is then repeated until the entire matrix S' is revealed. *Attack III* is only necessary for revealing the first two row elements $S'_{i,1}$ and $S'_{i,2}$. The following row elements can be revealed using either *Attack III*, or using *Attack I* and *Attack II*.

D. Attack on T map

Matrix T^{-1} has three non-trivial sub-matrices $T^{(1)}$, $T^{(4)}$ and $T^{(3)}$. We attack them separately, but similarly. First, we express the matrix-vector multiplication described in subsection II-B using equations

$$\begin{aligned} \widetilde{y}_{1:32} + T^{(1)} \cdot \widetilde{y}_{33:64} + T^{(4)} \cdot \widetilde{y}_{65:96} &= x_{1:32}, \\ \widetilde{y}_{33:64} + T^{(3)} \cdot \widetilde{y}_{65:96} &= x_{33:64}, \\ \widetilde{y}_{65:96} &= x_{65:96}. \end{aligned} \tag{12}$$

TABLE III
ATTACK III: REVEALING REMAINING ROWS.

Target	Intermediate value I	Power prediction
$S'_{i,1}$ and $S'_{i,2}$	$S'_{i,1} \cdot y_{33} + S'_{i,2} \cdot y_{34}$	$H_2^l + \mathrm{HW}(I)$
$S'_{i,3}$	$\sum_{j=1}^{2}(S'_{i,j} \cdot y_{j+32}) + S'_{i,3} \cdot y_{35}$	$H_3^l + \mathrm{HW}(I)$
$S'_{i,4}$	$\sum_{j=1}^{3}(S'_{i,j} \cdot y_{j+32}) + S'_{i,4} \cdot y_{35}$	$H_4^l + \mathrm{HW}(I)$
$\vdots$	$\vdots$	$\vdots$
$S'_{i,32}$	$\sum_{j=1}^{31}(S'_{i,j} \cdot y_{j+32}) + S'_{i,32} \cdot y_{64}$	$H_{32}^l + \mathrm{HW}(I)$
$k \in Set_l$	$\sum_{j=1}^{32}(S'_{i,j} \cdot y_{j+32}) + y_k$	$\mathrm{HW}(I)$

As this multiplication is performed at the end of the signing, we know the output x, but not the input $\widetilde{y}$. This is the main difference compared to attacking the S matrix.

1) Attack on $T^{(3)}$: To reveal $T^{(3)}$, we use the *Attacks I, II* as described for S, since we know the vector used in multiplication:

$$\underbrace{T^{(3)}}_{\text{secret key}} \cdot \underbrace{x_{65:96}}_{\text{known}} + \underbrace{\widetilde{y}_{33:64}}_{\text{unknown}} = \underbrace{x_{33:64}}_{\text{known}}. \tag{13}$$

Unfortunately, we cannot use the final addition to determine the row indices in this case. Instead, we use the right side of the Equation 13 and the fact that

$$\sum_{j=1}^{32}(T_{i,j}^{(3)} \cdot x_{j+65}) + x_{i+32} = \widetilde{y}_{i+32}. \tag{14}$$

Similarly to the S matrix attack, we use *Attack III* to reveal elements from rows beginning with zeroes. When $T^{(3)}$ is found, we can compute

$$\widetilde{y}_{33:64} = T^{(3)} \cdot x_{65:96} + x_{33:64}. \tag{15}$$

2) Attack on $T^{(4)}$: Attacking $T^{(4)}$ is performed in the same manner as attacking $T^{(3)}$, using equation

$$\underbrace{\widetilde{y}_{1:32} + T^{(1)} \cdot \widetilde{y}_{33:64}}_{\text{unknown}} + \underbrace{T^{(4)}}_{\text{secret key}} \cdot \underbrace{x_{65:96}}_{\text{known}} = \underbrace{x_{1:32}}_{\text{known}}. \tag{16}$$

The submatrix $T^{(4)}$ is multiplied by a known vector $x_{65:96}$. We use the right side $x_{1:32}$ of the Equation 16 for the row index determination.

3) Attack on $T^{(1)}$: Attacking $T^{(1)}$ is performed in the same fashion as attacking $T^{(3)}$ and $T^{(4)}$:

$$\begin{aligned} \widetilde{y}_{1:32} + T^{(1)} \cdot \widetilde{y}_{33:64} + T^{(4)} \cdot x_{65:96} &= x_{1:32}, \\ \underbrace{\widetilde{y}_{1:32}}_{\text{unknown}} + \underbrace{T^{(1)}}_{\text{secret key}} \cdot \underbrace{\widetilde{y}_{33:64}}_{\text{known}} &= \underbrace{x_{1:32} + T^{(4)} \cdot x_{65:96}}_{\text{known}}. \end{aligned} \tag{17}$$

The secret submatrix is multiplied by a known vector $\widetilde{y}_{33:64}$ computed using Equation 15. The right side of the Equation 17 is used for row index determination.

E. Extraction of the central map F

There are two possible approaches to extraction of the central map F with knowledge of S and T. The first one is extracting the central map F by eliminating T and S maps from the public key P. The second approach find the central

[2] Probability of non-zero values in the first two columns is $(15/16)^2$, the average number of these rows in a submatrix such as S' is $\mathrm{Mean}(\mathrm{BinomialDistribution}(32, (15/16)^2)) = 28.125$.

map F via known Rainbow inputs and outputs, using a system of linear equations. In this case, knowledge of the public key is not needed. Enough input and output data should be obtained while attacking the S and T maps.

IV. Experimental evaluation

We evaluate the proposed attack on a ChipWhisperer-Lite platform with a 32-bit STM32F303 microcontroller based on ARM Cortex-M4 core as a target. ChipWhisperer-Lite features an integrated 10-bit ADC with 105MS/s sampling rate and uses a synchronous sampling technique [10] for measurements of the target power consumption. We are attacking the implementation proposed in the NIST competition second round, with random data generated by a controlling PC.

For side-channel attack evaluation, we use the success rate [11], i.e., the expected probability of attack successfully distinguishing the correct subkey. The presented results are based on 33 independent experiments and further averaged over 32 random subkey elements/rows.

Attack I targets a single 4-bit subkey and is directly applicable to non-zero subkeys only as described in subsection III-C. Its success rate is therefore based on attacking non-zero subkeys only. *Attack II* is then used to distinguish between up to 32 matrix rows revealed by *Attack I*. Subkeys which *Attacks I and II* fail to reveal are then discovered using *Attack III*, which makes more precise predictions of a processed 32-bit word using previously discovered subkeys.

Attacks I and II (both targeting 4-bit value) have a success rate of 0.75/0.95 using approx. 280/475 power traces. *Attack III* has a success rate of 0.75/0.95 with one, three, or seven other known subkeys (i.e., targeting eight, 16, or 32 bits) using approx. 150/240, 90/140, or 40/70 power traces, respectively.

Attack III exhibits a better success rate than *Attacks I and II*, which is expected thanks to better signal-to-noise ratio given more precise power predictions.

V. Multiplicative Masking Countermeasure

A simple countermeasure masking the matrix-vector multiplication is proposed in [8]. The input value y is randomized via multiplication by a scalar mask $m \in \mathbb{F}$. The correct output can then be obtained by multiplying the result with mask inversion: $m^{-1} \cdot S^{-1}(m \cdot y) = S^{-1}(y) = \widetilde{x}$. Using this approach, internal values are masked and unmasked two times and during computation of the central map F, intermediate values are not masked at all.

The discussed reference implementation uses central map F with zero linear and absolute coefficients. Therefore, a single mask may be used throughout the whole signature process. After the initial hashing, the vector y is multiplied by a squared mask m^2. The output is then unmasked by multiplying it with the inversion of the mask m^{-1}:

$$x = m^{-1} \cdot T^{-1}\left(F^{-1}\left(S^{-1}\left(m^2 \cdot y\right)\right)\right). \tag{18}$$

The linear maps are homogeneous of degree one and the polynomials $f^{(k)}, k \in \{v_1 + 1, \ldots, n\}$, in the central map F are homogeneous of degree two. Assuming more general polynomials for now, the behavior of the masked input x in the quadratic part can be described by

$$\begin{aligned} f^{(k)}(m \cdot x) = & \sum_{i,j \in \{1,\ldots,n\}} c_{i,j}^{(k)} \cdot (m \cdot x_i) \cdot (m \cdot x_j) = \\ & = m^2 \sum_{i,j \in \{1,\ldots,n\}} c_{i,j}^{(k)} \cdot x_i \cdot x_j = m^2 \cdot y_{k-v_1}, \end{aligned} \tag{19}$$

and therefore $m^{-1} \cdot (f^{(k)})^{-1}(m^2 \cdot y_{k-v_1}) = m^{-1} \cdot (m \cdot x) = x$.

VI. Conclusion

In this paper, we presented a side-channel attack on the Rainbow digital signature, NIST's third round candidate for post-quantum standard. We analyzed the 32-bit reference implementation and proposed three combined Correlation Power Analysis attacks allowing extraction of a full secret key. We evaluated the proposed attack on a 32-bit microcontroller with ARM Cortex-M4 core and successfully extracted the secret key. Finally, we proposed an extension of a known masking scheme, that allows randomization of intermediate values used in the signing process, including the non-linear part, with no significant time or memory overhead.

Acknowledgments

The authors would like to thank Jan Onderka for proofreading. This work was supported by the Czech Technical University (CTU) grant No. SGS20/211/OHK3/3T/18; and by the Student Summer Research Program 2020 of FIT CTU in Prague.

References

[1] IEEE Std 1363-2000 Working Group and others, "IEEE 1363: Standard Specifications for Public-Key Cryptography," *IEEE, Inc., USA*, 2000.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[3] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *International Conference on Applied Cryptography and Network Security*. Springer, 2005, pp. 164–175.

[4] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced oil and vinegar signature schemes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 206–222.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[6] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual international cryptology conference*. Springer, 1999, pp. 388–397.

[7] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *International workshop on cryptographic hardware and embedded systems*. Springer, 2004, pp. 16–29.

[8] A. Park, K.-A. Shim, N. Koo, and D.-G. Han, "Side-channel attacks on post-quantum signature schemes based on multivariate quadratic equations," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 500–523, 2018.

[9] A. Petzoldt, S. Bulygin, and J. Buchmann, "Cyclicrainbow–a multivariate signature scheme with a partially cyclic public key," in *International Conference on Cryptology in India*. Springer, 2010, pp. 33–48.

[10] C. O'Flynn and Z. Chen, "Synchronous sampling and clock recovery of internal oscillators for side channel analysis and fault injection," *Journal of Cryptographic Engineering*, vol. 5, no. 1, pp. 53–69, 2015.

[11] F.-X. Standaert, T. G. Malkin, and M. Yung, "A unified framework for the analysis of side-channel key recovery attacks," in *Annual international conference on the theory and applications of cryptographic techniques*. Springer, 2009, pp. 443–461.