Punctured Syndrome Decoding Problem

Vincent Grosso, Pierre-Louis Cayrel, Brice Colombier, Vlad-Florin Drăgoi

Punctured Syndrome Decoding Problem
Efficient Side-Channel Attacks Against *Classic McEliece*

Vincent Grosso[1][0000−0002−3874−7527], Pierre-Louis Cayrel[1][0000−0002−6708−868X], Brice Colombier[1][0000−0002−6028−3028], and Vlad-Florin Drăgoi[23][0000−0002−8673−9097]

[1] Université Jean Monnet Saint-Etienne, CNRS, Institut d Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
[2] Faculty of Exact Sciences, Aurel Vlaicu University, Arad, Romania
[3] LITIS, University of Rouen Normandie, Saint-Etienne du Rouvray, France

Abstract. Among the fourth round finalists of the NIST post-quantum cryptography standardization process for public-key encryption algorithms and key encapsulation mechanisms, three rely on hard problems from coding theory. Key encapsulation mechanisms are frequently used in hybrid cryptographic systems: a public-key algorithm for key exchange and a secret key algorithm for communication. A major point is thus the initial key exchange that is performed thanks to a key encapsulation mechanism. In this paper, we analyze side-channel vulnerabilities of the key encapsulation mechanism implemented by the *Classic McEliece* cryptosystem, whose security is based on the syndrome decoding problem. We use side-channel leakages to reduce the complexity of the syndrome decoding problem by reducing the length of the code considered. The columns punctured from the original code reduce the complexity of a hard problem from coding theory. This approach leads to efficient profiled side-channel attacks that recover the session key with high success rates, even in noisy scenarios.

Keywords: Post-quantum cryptography · Code-based cryptography · Side-channel attacks.

1 Introduction

Recent developments in quantum computing threaten classical public key cryptography. Indeed, Shor's algorithm [22] could be used to break public key schemes such as RSA or Diffie-Hellman. Therefore, to prepare security in the quantum computing era, in 2016, NIST launched a standardization process for post-quantum cryptography standards to replace current public-key standards which are vulnerable to quantum computing. In July 2022, the fourth round of the standardization process started. Among the four remaining candidates for public key encryption algorithms and key encapsulation mechanisms, three are code-based solutions: *Classic McEliece* [2], BIKE [3], and HQC [1].

All three proposals implement a solution for IND-CCA secure key exchange using a Key Encapsulation Mechanism (KEM) [15]. KEMs are used to exchange

private session key over an insecure channel using public cryptography scheme. To avoid short message and padding issues while using public-key encryption schemes, a key derivation function is used allowing to generate the message sent in the right domain and using most of the time a hash function to derive a uniform random looking secret key, assuming enough entropy in the original message. The security of the private communication relies on the security of the KEM, assuming that secure symmetric algorithms are used. Moreover KEM can be seen as a key-exchange protocol in which only a single message is transmitted, if one of the two parties knows the public key of the second party.

For both *Classic McEliece* and BIKE, the security of the KEM relies on the hardness of the binary Syndrome Decoding Problem (SDP). Conversely, the security of HQC essentially relies on the hardness of decoding a general linear code. The binary SDP is an $\mathcal{NP}$-hard problem stating the following. Knowing a matrix $\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$, an integer $t \leq n$ and a vector $\boldsymbol{s}^* \in \mathbb{F}_2^{n-k}$, it is difficult to recover $\boldsymbol{e} \in \mathbb{F}_2^n$ such that $\boldsymbol{He} = \boldsymbol{s}^*$ and $\mathrm{HW}(\boldsymbol{e}) = t$. The vector $\boldsymbol{s}^*$ is usually referred to as the *syndrome*. In a KEM, the vector $\boldsymbol{s}^*$ is sent, and the secret data $\boldsymbol{e}$ is reconstructed by the recipient. Thus the encapsulation algorithm consists of a matrix-vector multiplication. The difficulty of the problem depends on the weight t of $\boldsymbol{e}$. The problem is difficult when $\boldsymbol{e}$ is of sufficiently "low weight".

Some of the best solutions to solve the binary SDP make use of the so-called "information set decoding" strategy (ISD) [19,23,14,16,4]. The key idea is to exploit the "low weight" property of $\boldsymbol{e}$ by selecting a sufficient number of columns that do not operate in the computation of $\boldsymbol{s}^*$. Afterwards, Gaussian elimination can be performed on the other columns. However, selecting the columns is a challenging phase.

A consequence of the NIST post-quantum cryptography standardization process is to accelerate the development of implementations of code-based cryptography algorithms [21,7,18,8]. In particular, *Classic McEliece* has been implemented on 32-bit microprocessor ARM Cortex-M4 [7], with the limitation that the public key must be stored in the flash memory, and on a Xlilinx Artix-7 FPGA [8]. Implementations on constrained platforms, such as micro-controllers or FPGAs, also lead to physical attacks against different algorithms of code-based cryptography [13,5,9,12]. For example, it has been shown that the session key can be recovered by side-channel attacks with multiple observations during the decapsulation process [13] or with a single observation during the encapsulation process [9]. Colombier *et al.* demonstrated the effectiveness of their method against an implementation on the Chipwhisperer platform, which is known to allow for low-noise side-channel measurements. The efficiency of the proposed method in a more noisy setting was later analyzed in [10].

The focus of this article is on *Classic McEliece*, in particular the matrix-vector multiplication over $\mathbb{F}_2$ used in the syndrome computation. Conversely, BIKE uses polynomial multiplication. Adaptations are needed to apply the attack against other finalists but this is out of the scope of this article.

Contribution This article exposes in details the inherent limitations of previously proposed side-channel attacks against *Classic McEliece* presented in [9]. In par-

ticular, we explain the performance degradation of the existing approach when large noise levels are considered. Besides the intrinsic uncertainty of the Hamming weight classifier, we show that, overall, it is mainly due to an accumulation of errors in the way the integer syndrome is computed, as required by the attack setting and explained below. We then present a new, more efficient method that achieves better resistance against noise present in side-channel traces by resorting to a more traditional divide-and-conquer approach.

This new method is a profiled side-channel attack against Niederreiter-like constructions using packed matrix-vector multiplications, as used in the round four finalist of the NIST standardization process *Classic McEliece*. Moreover, we also study the feasibility of the proposed attacks against implementations that use a larger register size, which is a clear trend in embedded software implementations.

Organization This article is organized as follows. Section 2 describes existing message-recovery attacks on the packed matrix-vector multiplication as used for the syndrome computation in the *Classic McEliece* cryptosystem. The inherent limitations of these attacks, in particular when it comes to error propagation, are detailed in Section 3. In Section 4, we introduce a divide-and-conquer strategy that efficiently limits the propagation of errors. Experimental results are given in Section 5 and we conclude in Section 6.

2 Message-recovery attacks on the packed matrix-vector multiplication

This section introduces code-based KEMs and the target algorithm of the proposed side-channel attack: *Classic McEliece*. In particular, we focus on the matrix-vector multiplication performed during the encapsulation step. We also present previous side-channel attacks that recover the shared session key.

Notations The following notations are used in this article. A finite field is denoted by $\mathbb{F}$. Matrices and vectors are written in bold capital, respectively small letters, *e.g.* a vector of length n is $\boldsymbol{c} = (c_1, \ldots, c_n)$ and a $k \times n$ matrix is $\boldsymbol{H} = (h_{i,j})_{(i,j)\in\mathbb{N}_k^* \times \mathbb{N}_n^*}$. Let $\boldsymbol{H}_{i,(j-1)w+1:jw}$ be the j^{th} block of size w of the i^{th} row of the $\boldsymbol{H}$ matrix. The concatenation of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is written as $\boldsymbol{a} \parallel \boldsymbol{b}$. The Hamming weight of a binary vector $\mathrm{HW}(\boldsymbol{e})$ is the number of its non-zero coordinates. The Hamming distance between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is written as $\mathrm{HD}(\boldsymbol{a}, \boldsymbol{b})$.

2.1 *Classic McEliece* encapsulation

Like others KEMs, *Classic McEliece* includes three operations: key generation, encapsulation and decapsulation. We focus on the encapsulation step in this work. This is detailed in Algorithm 1, where the target operation of the proposed attack is annotated. This target operation performs a matrix-vector multiplication over $\mathbb{F}_2$ and its implementation is detailed in the next subsection.

Algorithm 1 *Classic McEliece* encapsulation

1: **function** ENCAP($\boldsymbol{H}$)
2: Generate a uniform random vector $\boldsymbol{e} \in \mathbb{F}_2^n$ with $\mathrm{HW}(\boldsymbol{e}) = t$.
3: Compute $C \leftarrow \boldsymbol{H}\boldsymbol{e}$ ▷ target operation
4: Compute $K \leftarrow \mathsf{H}(1 \parallel \boldsymbol{e} \parallel C)$ ▷ session key
5: **return** (C, K)

2.2 Packed matrix-vector multiplication

Algorithm 2 shows the pseudo code of a software implementation of the matrix-vector multiplication over $\mathbb{F}_2$. This implementation is referred to as "packed" since multiple bits are stored together in the same machine word. The size of the machine word w is a parameter in this algorithm. In the reference implementation of the *Classic McEliece* submission [2], $w = 8$. In the ARM Cortex-M4 implementation by Chen and Chou [7], $w = 32$. In the vectorized implementation of the *Classic McEliece* submission [2], $w = 64$. Boolean instructions then operate over the full machine word to perform operations in parallel. That is the key operation of the encapsulation step in the *Classic McEliece* KEM. As shown in [9], the strongest side-channel leakage occurs for line 5, when the intermediate variable b is updated by repeatedly adding the logical AND of a matrix entry and a vector entry. To be able to refer to specific intermediate values later, we write these intermediate variables as if they were stored in a matrix: $\boldsymbol{b}_{i,j}$. In actual implementations, a single machine word is used.

Algorithm 2 Packed matrix-vector multiplication over $\mathbb{F}_2$

Require: A binary $(n, n-k)$ matrix $\boldsymbol{H}$, and a binary vector $\boldsymbol{e}$ of n elements, the register size w (should be a power of 2)
Ensure: A binary vector $\boldsymbol{s}^* = \boldsymbol{H}\boldsymbol{e}$

1: $\boldsymbol{s}^* \leftarrow 0$
2: **for** $i \leftarrow 1$ to $(n-k)$ **do**
3: $\quad \boldsymbol{b}_{i,0} \leftarrow 0$
4: $\quad$ **for** $j \leftarrow 1$ to $\frac{n}{w}$ **do**
5: $\quad\quad \boldsymbol{b}_{i,j} \leftarrow \boldsymbol{b}_{i,j-1} \oplus \boldsymbol{H}_{i,(j-1)w+1:jw} \wedge \boldsymbol{e}_j$
6: $\quad t \leftarrow \frac{w}{2}$
7: $\quad$ **while** $t > 0$ **do**
8: $\quad\quad \boldsymbol{b}_{i,\frac{n}{w}} \leftarrow \boldsymbol{b}_{i,\frac{n}{w}} \oplus (\boldsymbol{b}_{i,\frac{n}{w}} \gg t)$
9: $\quad\quad t \leftarrow \frac{t}{2}$
10: $\quad \boldsymbol{s}^*_{\lfloor \frac{i}{w} \rfloor} \leftarrow \boldsymbol{s}^*_{\lfloor \frac{i}{w} \rfloor} \vee \left(\left(\boldsymbol{b}_{i,\frac{n}{w}} \wedge 1 \right) \ll (i \bmod w) \right)$
11: **return** $\boldsymbol{s}^*$

2.3 Message recovery attack

We describe the method introduced in [9], to recover session keys on cryptosystems based on the binary syndrome decoding problem. This attack uses side-channel information obtained during the encapsulation step. The message recovery attack is composed of four steps, which we describe hereafter.

1. *Side-channel analysis:* the goal of this first step is to estimate the Hamming weight of the successive intermediate values of $\boldsymbol{b}$ during the matrix-vector multiplication, as shown on line 5 in Algorithm 2. For the loop index i, j, we denote by $\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j})$ the best guess for the Hamming weight of $\boldsymbol{b}_{i,j}$. In [9], authors use a random-forest classifiers for this step, but other classifiers can be used.
2. *Derivation of the integer syndrome:* with the Hamming weight information obtained in the first step, the attacker may estimate the values of the syndrome $\boldsymbol{s}$ in $\mathbb{N}$, in addition to the binary syndrome $\boldsymbol{s}^*$ which is public. This is done by summing the differences of the maximum of each value found in the previous step, as detailed in Equation (1).

$$1 \leq i \leq (n-k) \quad \widetilde{s_i} = \sum_{j=1}^{\frac{n}{w}} \left| \widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j}) - \widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j-1}) \right| \tag{1}$$

   This computation requires a good estimation of the Hamming weight of the intermediate values. In addition, it only works under additional conditions between $\boldsymbol{b}_{i,j}$ and $\boldsymbol{b}_{i,j-1}$. If those conditions are not met, it can lead to derive an erroneous value for the integer syndrome. We discuss these issues in more details in Section 3.
3. *Sort columns:* the next step is to separate the columns into two sets. The first set consists of the columns whose indexes are in the support of $\boldsymbol{e}$. The second set consists of the other columns. However, this separation is a difficult task. In [9], authors compute a score for each column and sort columns according to this score. The score for the column j, based on the work of Feige and Lellouche [11], is defined in Equation (2).

$$\forall j \in [\![1, n]\!], \quad \psi_j(\widetilde{\boldsymbol{s}}) = \boldsymbol{H}_{.,j} \cdot \widetilde{\boldsymbol{s}} + \overline{\boldsymbol{H}}_{.,j} \cdot \overline{\widetilde{\boldsymbol{s}}} \tag{2}$$

   where $\overline{\boldsymbol{H}}$ is the complementary of the matrix $\boldsymbol{H}$and $\overline{\widetilde{\boldsymbol{s}}} = t - \widetilde{\boldsymbol{s}}$, where t is the weight of $\boldsymbol{e}$ as dictated by the security parameters. In [10], the efficiency of this score function is analyzed in the presence of errors.
4. *Information Set Decoding:* as shown in [10] the score function allows to efficiently discriminate most of the columns in the support of $\boldsymbol{e}$ from other columns even in the presence of noise. However, a few columns may still be wrongly classified. In that case, the score function is used to provide a "good" initial permutation for ISD methods.

This method achieves a good success rate in a realistic scenario with measurements on a ChipWhisperer platform [17] for various sets of parameters.

3 Limitation of the CDCG method

In this section, we present errors that can appear in the method of [9] and reduce the efficiency of the message recovery. We concentrate on side-channel and recombination errors that lead to an incorrect syndrome in $\mathbb{N}$, *i.e.*, the two first steps presented in section 2.3. Eventually, we discuss the impact of such errors on scores output by the ψ score function [11].

3.1 Side-channel analysis error

We first try to identify how side-channel analysis errors alter the estimation of the syndrome $\boldsymbol{s}$ in $\mathbb{N}$. Due to their nature and noise in measurements, side-channel attacks can output guesses that are not the targeted sensitive information used in the implementation.

We say that the side-channel distinguisher makes an error if the highest guess score does not correspond to the Hamming weight of the actual computation: $\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j}) \neq \mathrm{HW}(\boldsymbol{b}_{i,j})$. We may rewrite the faulty guess as:

$$\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j}) = \mathrm{HW}(\boldsymbol{b}_{i,j}) + \varepsilon_{i,j},$$

with $\varepsilon_{i,j} \neq 0$.

Due to the leakage model, the error is generally small: $\varepsilon_{i,j} \in \{-1, 1\}$ The Hamming weight guess is the real Hamming weight plus or minus one. In practice, we observe on real traces that the error is small for template and random forests when used as side-channel distinguishers.

If the side-channel distinguisher made some errors for the value in row i and column j then the estimated syndrome in $\mathbb{N}$ will be flawed in the i^{th} position. This is clear when considering how the i^{th} component of $\boldsymbol{s}$ is derived from the Hamming weight of the intermediate values:

$$\begin{aligned} \widetilde{s}_i &= \sum_{j=1}^{\frac{n}{w}} |\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j-1}) - \widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j})| \\ &= \sum_{j=1}^{\frac{n}{w}} |\mathrm{HW}(\boldsymbol{b}_{i,j-1}) - \mathrm{HW}(\boldsymbol{b}_{i,j})| + \varepsilon_{s_i}, \end{aligned} \tag{3}$$

where ε_{s_i} comes from the side-channel error on the i^{th} syndrome entry.

This $\varepsilon_{i,j}$ value actually appears in two Hamming distances: for j and $j-1$. As a consequence, the recombination step given in Equation (3) *amplifies* the side channel noise.

Remark : $\widetilde{s}_i$ corresponds to the number of transitions between $\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j-1})$ and $\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,j})$ for j going from 1 to n/w.

Example 1. For a given row i, let $\mathrm{HW}(\boldsymbol{b}_{i,.}) = (0, 0, 1, 1, 1, 2, 1, 1)$ be the error-free sequence of Hamming weights of the intermediate values. Then, the estimation

part should give a guess value of $\widetilde{s}_i = 3$. Indeed, there are 3 transitions $0 \to 1, 1 \to 2$ and $2 \to 1$.

Depending on where the error $\varepsilon_{i,j}$ appears, the consequence on the $\widetilde{s}_i$ value differs.

- Let's assume we observe $\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,.}) = (0, 1, 1, 1, 1, 2, 1, 1)$, $\varepsilon_{i,1} = +1$ affects $\mathrm{HW}(\boldsymbol{b}_{i,1})$. We derive $\widetilde{s}_i = 3$ and therefore $\varepsilon_{s_i} = 0$
- Let's assume we observe $\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,.}) = (0, 0, 1, 1, 1, 1, 1, 1)$, $\varepsilon_{i,5} = -1$ affects $\mathrm{HW}(\boldsymbol{b}_{i,5})$. We derive $\widetilde{s}_i = 1$ and therefore $\varepsilon_{s_i} = -2$
- Let's assume we observe $\widetilde{\mathrm{HW}}(\boldsymbol{b}_{i,.}) = (0, 0, 1, 1, 1, 2, 1, 2)$, $\varepsilon_{i,7} = +1$ affects $\mathrm{HW}(\boldsymbol{b}_{i,7})$. We derive $\widetilde{s}_i = 4$ and therefore $\varepsilon_{s_i} = 1$

As shown in the example, a negative or null impact on the estimation of the integer syndrome entry can happen. However, these cases occur with low probability.

The side-channel error is directly linked to the accuracy of the side-channel distinguisher. Indeed, the accuracy corresponds to the probability of a correct guess. We see that, with high probability, any wrong guess of the side-channel distinguisher will lead to an overestimation of the syndrome entry.

3.2 "Double-cancellation" error

Another error that can appear, as already discussed in [9], was called the "double cancellation" issue. We recall the problem briefly. We are interested in the successive Hamming weights of the partial matrix-vector product. However, the observations we get are the successive Hamming weight of the $\boldsymbol{b}$ value in line 5 of Algorithm 2. Thus, in the CDCG method, the values we are interested in are estimated with the following approximation of the Hamming distance from the Hamming weight:

$$\mathrm{HD}(\boldsymbol{a}, \boldsymbol{b}) \simeq |\mathrm{HW}(\boldsymbol{a}) - \mathrm{HW}(\boldsymbol{b})|.$$

With this approximation, the $2\mathrm{HW}(\boldsymbol{b} \wedge \neg(\boldsymbol{a}))$ part of the Hamming distance computation is omitted. In our case, we know that both vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ are close due to the low weight of the input vector. Indeed, if we look at Line 5 in Algorithm 2, we can notice that in our case, we can consider one vector $\boldsymbol{a}$ to be random, but the second is of the form $\boldsymbol{b} = \boldsymbol{a} \oplus \boldsymbol{c}$, with $\boldsymbol{c}$ of low weight. Indeed $\boldsymbol{c}$ corresponds to the bitwise AND between a vector that looks random, a sub-group of columns of a line of the matrix $\boldsymbol{H}$, that is indistinguishable from a random matrix, and a subpart of the vector $\boldsymbol{e}$ of low weight $\boldsymbol{c} = \boldsymbol{H}_{i,j} \wedge \boldsymbol{e}_j$. In particular, $\mathrm{HW}(\boldsymbol{c}) \geq 2$ implies $\mathrm{HW}(\boldsymbol{e}) \geq 2$.

The following theorem gives the weight distribution of the blocks.

Theorem 1. *Let n, t, w be strictly positive integers with $t < n$ and w divides n. Let X_i be a discrete random variable denoting the number of blocks of weight i of a binary string of length n and Hamming weight t, where each block has length*

w. For any $2 \leq j \leq w$ *let* $\alpha_j \in \{0, \ldots, t\}$ *satisfying* $\sum_{\ell=1}^{j} \ell\alpha_\ell = t$. *Then*

$$\Pr(X_j = \alpha_j, \ldots, X_2 = \alpha_2, X_1 = \alpha_1) = \frac{\binom{\frac{n}{w}}{\alpha_1,\ldots,\alpha_j}}{\binom{n}{t}} \prod_{\ell=1}^{j} \binom{w}{\ell}^{\alpha_\ell}, \tag{4}$$

where $\binom{\frac{n}{w}}{\alpha_1,\ldots,\alpha_j}$ *denotes the multinomial coefficient.*

Corollary 1. *The probability that the maximum weight is 1 equals* $\frac{w^t\binom{\frac{n}{w}}{t}}{\binom{n}{t}}$.

Moreover, for $t = o(n)$ *when* $n \to \infty$ *the probability that the weights of the blocks are at most 1 can be approximated by*

$$e^{-\frac{(w-1)t^2}{2n}\left(1+\frac{(w+1)t}{3n}+\frac{(w^2+w+1)t^2}{6n^2}+O\left(\frac{t^5}{n^4}\right)\right)}.$$

In particular, using only the first term in the exponent for block sizes $w \in \{8, 32, 64\}$ *gives* $e^{-\frac{7t^2}{2n}}$, $e^{-\frac{31t^2}{2n}}$ *and* $e^{-\frac{63t^2}{2n}}$.

Remark 1. In the case of *Classic McEliece*, we have $t = \mathcal{O}(\frac{n}{\log_2 n})$, which implies that the probability of having only weights 0 and 1 blocks is roughly $e^{-c\frac{n}{\log_2^2 n}}$, where c is a constant related to the block size w.

One can deduce that the probability of having at least one block of weight 2 is extremely high. This result implies that the CDCG method has a very high probability of underestimating the Hamming weights. One can notice, as shown in Table 1a, that for block sizes greater or equal to 32, the probability of having only blocks of weight 0 and 1 is extremely small. For $w = 8$ the weight of the blocks is with high probability at most 2. This shows that it is highly probable that at least one word of the $\boldsymbol{e}$ vector will lead to a recombination error, which will affect the estimated syndrome. We also know that in such a case, all wrong estimated values are underestimated.

Having many blocks of weight strictly greater than 1 increases the estimation error. Therefore, determining the expected number of such blocks would be useful.

In Table 1b, we compute a lower bound on the expected value of the number of such blocks. Notice that for $w = 8$ the number of blocks having weight larger than or equal to 2 is indeed, extremely small. Hence in such a scenario, the syndrome estimation should be rather close to the exact value. On the opposite, for $w = 64$ around 30% of the blocks are of Hamming weight greater than or equal to 2. As we shall see, large values of w have a devastating impact on the success probability of the CDCG method.

3.3 Dependent error

In Section 3.2, we showed that it is highly probable that we have a block of the vector with Hamming weight greater than 1. These blocks are problematic since

Table 1: Weight of blocks $\boldsymbol{e}_j$ for *Classic McEliece* parameters: probability of the maximum weight and lower bound on the average number of blocks with weights larger than or equal to 2.

(a) $\Pr(\max(\mathrm{HW}(\boldsymbol{e}_j)))$

w	max	(3488,64)	(4608,96)	(6688,128)
8	$=1$	0.01378	0.00061	0.00012
	≤ 2	0.873	0.767	0.739
	≤ 3	0.997	0.993	0.992
	≤ 4	0.999	0.999	0.999
	≤ 5	0.999	0.999	0.999
32	$=1$	8.69×10^{-11}	7.59×10^{-19}	3.1×10^{-22}
	≤ 2	0.0804	0.0077	0.0038
	≤ 3	0.753	0.543	0.519
	≤ 4	0.974	0.936	0.936
	≤ 5	0.997	0.994	0.994
64	$=1$	0	0	0
	≤ 2	5.66×10^{-5}	4.21×10^{-9}	3.80×10^{-10}
	≤ 3	0.159	0.021	0.015
	≤ 4	0.715	0.455	0.444
	≤ 5	0.947	0.865	0.869

(b) $|\{j \mid \mathrm{HW}(\boldsymbol{e}_j) \geq 2\}|/(n/w)$

w	(3488,64)	(4608,96)	(6688, 128)
8	3.8/436	6.4/576	7.9/836
32	12.6/109	20.0/144	25.9/209
64	17.1/54.5	24.2/72	31.7/105

they will impact approximately one-fourth of the Hamming weight estimation for the considered block.

Indeed, if $\mathrm{HW}(\boldsymbol{e}_j) = 2$ and the $\boldsymbol{H}_{i,j}$ are random words, then approximately one-fourth of the product for this word column has weight 2. Among this quarter, half of them are underestimated with the approximation used in the CDCG method if we consider $\boldsymbol{a}$ to be random. Hence, the double cancelation error will impact several results and the error induces by a word of weight higher than 1 will lead to dependent errors on the different syndrome estimations.

3.4 Impact of the error on the score computation

After the estimation step, the side-channel analysis error is increased. This error is then propagated with the evaluation of the column score with the ψ function from [11], as used in [9]. The score for the column j is defined as:

$$\forall j \in [\![1, n]\!], \quad \psi_j(\widetilde{\boldsymbol{s}}) = \boldsymbol{H}_{.,j} \cdot \widetilde{\boldsymbol{s}} + \overline{\boldsymbol{H}}_{.,j} \cdot \overline{\widetilde{\boldsymbol{s}}}. \tag{5}$$

Thus if, the i^{th} coordinate of $\widetilde{\boldsymbol{s}}$ is incorrect, it will modify the score of the whole column. If $\boldsymbol{H}_{i,j} = 1$, then the left part $(\boldsymbol{H}_{.,j} \cdot \widetilde{\boldsymbol{s}})$ is affected. In that case, the score computed by ψ for the column i is the error-free score plus ε_j. If $\boldsymbol{H}_{i,j} = 0$, then the left part $(\overline{\boldsymbol{H}}_{.,j} \cdot \overline{\widetilde{\boldsymbol{s}}})$ is affected. In that case, the score computed by ψ for the column i is the error-free score minus ε_j.

Therefore, any incorrect estimation during the side-channel analysis will influence all the results and affect them differently depending on the value of the bit in the $\boldsymbol{H}$ matrix. On average, half of the columns score will be over-evaluated while the other half will be under-evaluated.

4 Error propagation limitation

This section presents a different message recovery attack against Niederreiter-like schemes, that make use of a matrix-vector multiplication in $\mathbb{F}_2$. Our new method does not require estimating the syndrome in $\mathbb{N}$, as previously done in [5,9]. Moreover, it just looks at side-channel results locally and does not propagate the error discussed in Section 3.

4.1 Punctured matrices

In order to cope with the error propagation issue, we propose to use both the incorrect and correct Hamming weight estimations to distinguish between blocks of size w in the error vector where $\boldsymbol{e}_j = \mathbf{0}$ and $\boldsymbol{e}_j \neq \mathbf{0}$. We recall that the attacker has access to the estimations of the Hamming weight $\widetilde{\mathrm{HW}}(b_{i,j})$.

For simplification, we will denote $w_{i,j} = \widetilde{\mathrm{HW}}(b_{i,j})$ and the matrix of estimated weights $\boldsymbol{W} = (w_{i,j})_{1 \leq i \leq n-k, 1 \leq j \leq \frac{n}{w}}$. The j^{th} column vector of $\boldsymbol{W}$ is $\boldsymbol{w}_j \in \mathbb{N}^{n-k}$, more exactly, $\boldsymbol{w}_j = \left(\widetilde{\mathrm{HW}}(b_{1,j}), \widetilde{\mathrm{HW}}(b_{2,j}), \ldots, \widetilde{\mathrm{HW}}(b_{n-k,j})\right)$. Algorithm 3 below determines for which index j we have $\boldsymbol{e}_j = 0$.

Algorithm 3 ZERO-DISTINGUISHER

Require: $\boldsymbol{W}$: Hamming weight guess for each intermediate value of the $\boldsymbol{b}$ value in Algorithm 2 and a the estimate accuracy computed during profiling phase
Ensure: A set L of blocks to be punctured

$L = \{\emptyset\}$
$\gamma = (n-k)(1 - a^2 - \frac{(1-a)^2}{2}) + \sqrt{(2a^2 + (1-a)^2)(n-k)\log(n-k)}$
if $\mathrm{HW}(\boldsymbol{w}_1) \leq (n-k)(1-a) + \sqrt{2a(n-k)\log(n-k)}$ **then**
    $L \leftarrow L \cup \{1\}$
for $j \leftarrow 2$ to $\frac{n}{w}$ **do**
    if $\mathrm{HW}(\boldsymbol{w}_j - \boldsymbol{w}_{j-1}) \leq \gamma$ **then**
        $L \leftarrow L \cup \{j\}$
return L

If $\boldsymbol{e}_j = 0$ then this implies that, $\boldsymbol{b}_{i,j} = \boldsymbol{b}_{i,j-1}$ for $2 \leq i \leq n-k$. In other words, for the first block, the estimated weight vector $\boldsymbol{w}_1$ should be equal to zero if the estimation is perfect, and if the estimation is not perfect, depending on the accuracy, the value of $\mathrm{HW}(\boldsymbol{w}_1)$ (number of coordinates different from zero) should be rather small. For all the subsequent blocks, the condition $\boldsymbol{e}_j = 0$

Table 2: Distributions of the number of zeros in $\boldsymbol{w}_1$ and $\boldsymbol{w}_j - \boldsymbol{w}_{j-1}$.

	$\mathrm{HW}(\boldsymbol{e}_j) = 0$	$\mathrm{HW}(\boldsymbol{e}_j) = 1$
$n-k-\mathrm{HW}(\boldsymbol{w}_1)$	$\mathcal{B}(n-k, a)$	$\mathcal{B}\left(n-k, \frac{1+a}{4}\right)$
$n-k-\mathrm{HW}(\boldsymbol{w}_j - \boldsymbol{w}_{j-1})$	$\mathcal{B}\left(n-k, a^2 + \frac{(1-a)^2}{2}\right)$	$\mathcal{B}\left(n-k, \frac{1+a^2}{4}\right)$

implies that there should be no difference between $\boldsymbol{w}_j$ and $\boldsymbol{w}_{j-1}$ if the Hamming weight estimation is perfect. In the non-perfect case, the vector $\boldsymbol{w}_j - \boldsymbol{w}_{j-1}$ should have a small Hamming weight, that depends on the classification accuracy. The following theorem gives the necessary conditions on the accuracy a for Algorithm 3 to successfully output a list of valid zero-weight blocks. In order to distinguish between the case $\mathrm{HW}(\boldsymbol{e}_j) = 0$ and $\mathrm{HW}(\boldsymbol{e}_j) = 1$ we will use the following procedure. Denote the random variable $X_i = n - k - \mathrm{HW}(\boldsymbol{w}_j - \boldsymbol{w}_{j-1})$ given $\mathrm{HW}(\boldsymbol{e}_j) = 0$ and $Y_i = n - k - \mathrm{HW}(\boldsymbol{w}_j - \boldsymbol{w}_{j-1})$ given $\mathrm{HW}(\boldsymbol{e}_j) = 1$ for $j \geq 2$ (for $j = 1$ use $X_1 = n-k-\mathrm{HW}(\boldsymbol{w}_1)$). Then we say that one distinghuishes between X_j and Y_j with high probability as long as $\Pr(X_j > Y_j)$ is close to 1. To achieve our goal we will use known results on bounding the tail of binomial distribution and set up a threshold value β^* that acts as an almost perfect separation between the two distributions. More exactly, we will have that $X_j \geq \beta^*$ w.h.p. and $Y_j < \beta^*$ w.h.p. This value β^* will depend on the accuracy parameter a.

Theorem 2. *Assume that the errors are limited to a distance of 1 and overestimation and underestimation are equally probable. Let X_j and Y_j be the random variables as previously defined. Let $a_1 > \frac{1}{3} + \frac{40\log(n-k)}{9(n-k)} + \frac{8\sqrt{2}\sqrt{8\log(n-k)^2 + 3(n-k)\log(n-k)}}{9(n-k)}$ and $a_2 \geq 0.5$ be a solution of the equation:*

$$\sqrt{\frac{n-k}{\log(n-k)}} = \frac{4}{5a^2 - 4a + 1}\left(\sqrt{(3a^2 - 2a + 1)} - \sqrt{\frac{1+a^2}{2}}\right).$$

Then $\Pr(X_j > Y_j) > 1 - \frac{1}{(n-k)} - \frac{1}{e^{\mathcal{O}((3a-1)(n-k))}}$ as long as $a > a_1$ for $j = 1$ and $a > a_2$ for $j \geq 2$.

Moreover, the threshold separation value between the distributions of $n-k-X_j$ and $n-k-Y_j$ equals $(n-k)(1-a) + \sqrt{2a(n-k)\log(n-k)}$ for $j = 1$ and $(n-k)(1-a^2-\frac{(1-a)^2}{2}) + \sqrt{(2a^2+(1-a)^2)(n-k)\log(n-k)}$ for $j \geq 2$.

The proof of Theorem 2 is provided in the appendix. In Table 2, we illustrate the distributions of X_j and Y_j. Some restrictions on the level of accuracy are to be examined in details. For example if $a = 0.4$ the distribution of $n-k-\mathrm{HW}(\boldsymbol{w}_j - \boldsymbol{w}_{j-1})$ is almost identical when $\mathrm{HW}(\boldsymbol{e}_j) = 0$ and $\mathrm{HW}(\boldsymbol{e}_j) = 1$. The larger the difference between the parameters a (respectively $a^2 + \frac{(1-a)^2}{2}$) and $\frac{1+a}{4}$ (respectively $\frac{1+a^2}{4}$) the better for the distinguisher. The one-distance error assumption is based on Hamming weight leakages with Gaussian noise and

assumes univariate attacks. Previous work show that for low-noise setting this assumption can be fulfilled [20,26]

4.2 T-test based score

The method presented in the Section 4.1 is efficient when considering small registers, or equivalently small sub-matrices. However, as the register size increases, the number of columns kept is too high to perform an efficient ISD. For that, we propose a method to select a permutation for the ISD that can be used on the full matrix or its punctured version. Our method is based on a T-test [24]. The T-test is commonly used for leakage assessment to detect if side-channel traces are dependent on a parameter. The traces are separated into two sets according to the known value of a parameter which may have an influence on them. Here, we use the T-test to identify which columns have an impact on the side-channel traces difference.

To achieve this, for all groups of columns (depending on the implementation and parameter w), we separate the rows into two multisets according to the Hamming distance recovered during the side-channel attack (difference of the Hamming weight), the first one S_0 for Hamming distances equal to 0, the second one S_1 for the other cases. In an error-free scenario, two cases occur:

- All rows are in the same multiset (S_0), which means that none of the columns are used in the computation of the syndrome, and thus, the considered coordinates of the vector $\boldsymbol{e}$ are zero.
- The rows are distributed in the two multisets. Hence some coordinates of $\boldsymbol{e}$ are different from zero. We use a statistical test to determine which columns have a different distribution in the two multisets. If the coordinate of $\boldsymbol{e}$ is null, then the distribution should be similar in the two multisets, whereas if the coordinate of $\boldsymbol{e}$ is not null S_0 should contain rows with 0, and S_1 should contain rows with 1.

In order to deal with errors, either from side-channel analysis or recombination, we use a statistical test to deal with the misplacement of rows in the multisets. The method is described in Algorithm 4. The next step is to use the permutation obtained as an initial permutation for ISD-based methods.

5 Experimental validation

In this section, we compare our method with the CDCG method in various settings, to evaluate the different approaches in different case studies. In particular, we want to illustrate the limitations of the CDCG method we described before in Section 3. Our experimental validation confirms that our T-test-based approach is better suited than the previous method in low and large noise settings. For that, we consider simulation leakages and optimal template attacks [6] , *i.e.* with perfect modeling. We consider a leakage of the form $\mathcal{L}_{i,j} = \mathrm{HW}(\boldsymbol{b}_{i,j}) + \mathcal{N}(0, \sigma^2)$, where the Hamming weight HW can be on $w = 8$, 32 or 64-bit values and the

Algorithm 4 T-test based attacks

Require: $\boldsymbol{W}$: Hamming weight guesses for each intermediate value of the $\boldsymbol{b}$ value in Algorithm 2 and a binary $(n, n-k)$ matrix $\boldsymbol{H}$.
Ensure: A n-permutation ϕ (of the coordinates of the vector $\boldsymbol{e}$).

for $j \leftarrow 1$ to $\frac{n}{w}$ **do**
  $(S_0, S_1) \leftarrow (\{\emptyset\}, \{\emptyset\})$
  for $i \leftarrow 1$ to $n-k$ **do** ▷ Separate columns according to side-channel analysis
    if $w_{i,j} - w_{i,j-1} = 0$ **then**
      $S_0 \leftarrow S_0 \cup \boldsymbol{H}_{i,(j-1)w+1:jw}$
    else
      $S_1 \leftarrow S_1 \cup \boldsymbol{H}_{i,(j-1)w+1:jw}$
  T-score$[(j-1)w+1 : jw] \leftarrow$T-test$(S_0 \sim S_1)$ ▷ Perform feature selection
return $\phi \leftarrow$ argsort(T-score) ▷ Sort in decreasing order

noise variance σ^2 affects the side-channel distinguisher accuracy. To estimate the accuracy of the template attack, we use the 3-σ rule $a \simeq \text{erf}\left(\frac{1}{2\sqrt{2}\sigma}\right)$, where erf is the Gauss error function [25]. While this estimation may not be true for limit case, *i.e.* $\text{HW}(\boldsymbol{b}_{i,j}) = 0$ or $\text{HW}(\boldsymbol{b}_{i,j}) = w$. We also evaluate the accuracy of the distinguisher and the one observed in experimental results is close to the 3-σ rule one. This is due to the fact that most of the values of $\text{HW}(\boldsymbol{b}_{i,j})$ are close to $\frac{w}{2}$, and we consider a relatively low-noise case. Experiments confirm that the puncturing methods offer better results than the previous method for large registers and/or high noise.

For reproducibility, the source code of the simulation is given in `https://github.com/vingrosso/Side-channel-attacks-Classic-McEliece.git`.

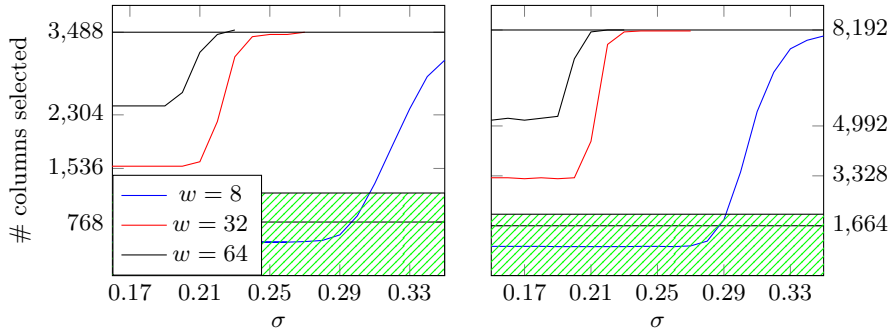
5.1 Punctured matrices

In this experiment, we consider the selection method to reduce the ISD problem via the method presented in Section 4.1. We arbitrary set to 2^{40} binary operations the maximum value of a computationally feasible attack. All the lower values are part of the so-called computationally feasible zone.

The idea is to evaluate the resistance to noise of the selection method for different register sizes. We simulate 100 experiments for the first and last set of parameters of *Classic McEliece* $(n, k, t) = (3488, 2720, 64)$ and $(n, k, t) = (8192, 6528, 128)$. The results are plotted in Figure 1.

As expected for small register size $w = 8$ and high accuracy $a = 0.92$ ($\sigma = 0.26$), the punctured method allows for an effective discrimination of a sufficient number of blocks of columns. Consequently, a simple Gaussian elimination is sufficient to recover the syndrome up to $\sigma < 0.29$. When the noise variance increases, a reduced syndrome decoding problem can be solved. However, the number of columns kept becomes rapidly large, close to all the columns, and requires too much computational power to mount a successful attack.

For large registers ($w = 32$ and $w = 64$), each block kept adds 32 or 64 columns for only one or two selected columns. Hence, even for low noise and



(a) $(n, k, t) = (3488, 2720, 64)$. (b) $(n, k, t) = (8192, 6528, 128)$.

Fig. 1: Median number of columns selected with the punctured method. The hatching zone corresponds to the computationally feasible zone.

high accuracy, the number of columns kept is too large and compromises the success of an ISD attack.

For the set of parameters of *Classic McEliece* the length of the codes are divided into a number of blocks n/w equal to $[436, 576, 836, 1024]$ (for $w = 8$), $[109, 144, 209, 256]$ (for $w = 32$) and $[51, 72, 105, 128]$ (for $w = 64$). As for the codimension of the code we obtain a number of blocks $(n - k)/w$ equal to $[96, 156, 208, 208]$ (for $w = 8$), $[24, 39, 52, 52]$ (for $w = 32$) and $[12, 20, 26, 26]$ (for $w = 64$).

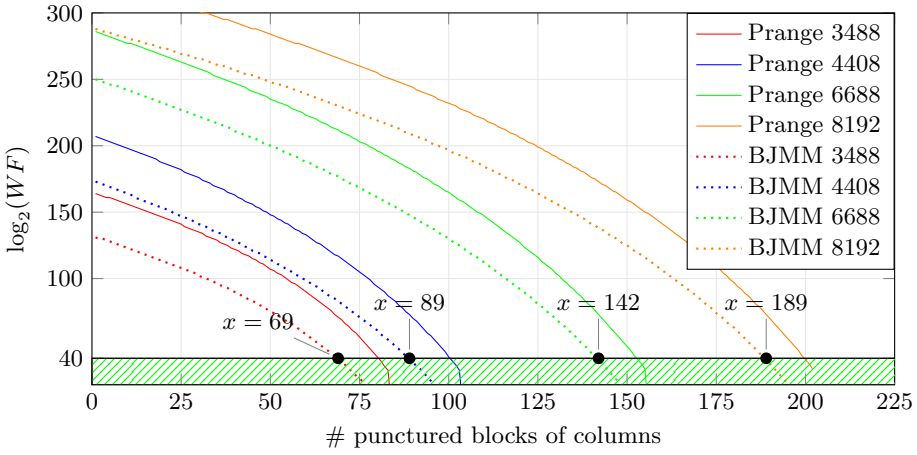


Fig. 2: Two ISD variants on punctured matrices ($w = 32$).

In Figure 2, we represent an estimated complexity of two ISD variants, Prange [19] and BJMM [4], when applied on punctured matrices. At each step, we increase by one the number of blocks of size $w = 32$ that are to be removed. Hence, we decrease the length of the matrix by 32, while keeping the same codimension, *i.e.* $n - k$ is constant. For example, the Prange variant applied on $n = 3488$ with 80 punctured blocks, gives a complexity smaller than 2^{30}. In this case, one has to remove 2560 columns out of the information set which is of size $k = 2720$. The horizontal solid line at $y = 40$ points out a rough limit from where ISD techniques become computationally feasible in practice. Computing the intersection points of this line with the BJMM variants gives a number of blocks to be punctured equal to $[69, 89, 142, 189]$. This implies that one needs to distinguish $[2208, 2848, 4544, 6048]$ columns, *i.e.* to select $[1280, 1760, 2144, 2144]$ columns. Represented as a factor, one has to select $[1.66, 1.41, 1.28, 1.28]$ times $(n - k)$ columns to perform the BJMM attack with a time complexity of 2^{40} binary operations.

5.2 Impact of the side-channel distinguisher accuracy

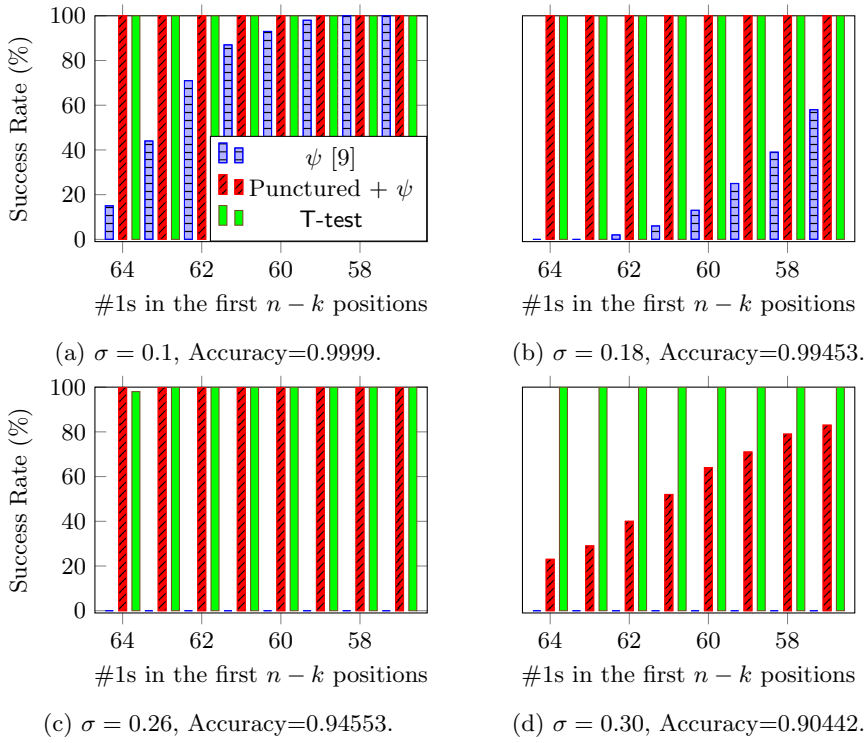
In this experiment, we evaluate the impact of the accuracy of the distinguisher on the success of three methods: CDCG punctured, CDCG and T-test. We consider Hamming weight 8-bit leakages, which means $0 \leq \mathrm{HW}(\boldsymbol{b}_{i,j}) \leq 8$, and we consider different values of noise σ to modify the accuracy. We work with the first set of parameters of *Classic McEliece*: $(n, k, t) = (3488, 2720, 64)$. In Figure 3, we can notice that for every accuracy parameter evaluated, the T-test method achieves a similar success rate, while the success rate of the CDCG method drops rapidly when the accuracy decreases. Applying the score function ψ on the punctured matrix does not help: the limit appears as early as for $\sigma = 0.3$ for punctured matrices as shown in the experiments of Section 5.1.

As discussed in Section 3.1, this was expected since every side-channel error will have an effect on the syndrome computation, and each incorrect coordinate in the syndrome will have an impact on all the column scores. By contrast, side-channel error in the proposed method will only alter the two columns where the incorrect Hamming weight is used, and, thanks to the large number of rows, we can correct this error efficiently.

In Figure 4, we consider the T-test method only and look at the impact of the accuracy value. As expected, the lower the accuracy, the less efficient the methods. Finally we can notice that the size of the population in the T-test method helps in the columns selection step. Indeed, when considering larger parameter sets, the success rate increases.

5.3 Impact of the register size

In the next experiment, we highlight the recombination error discussed in section 3.2. As discussed in the previous section, the larger the register, the more likely dependent errors are. Hence, we expect the success rate of all methods to drop when larger registers are considered. We fix the accuracy at 0.99822 for a



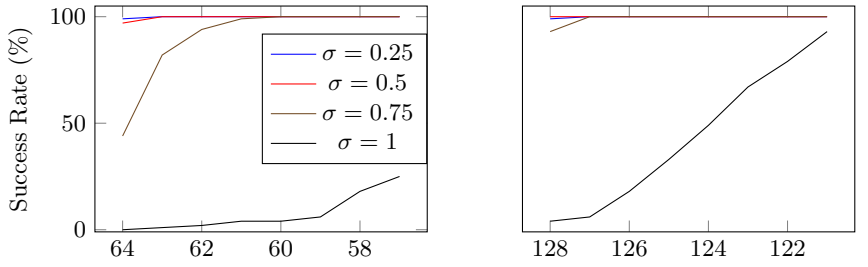
(a) $\sigma = 0.1$, Accuracy=0.9999.

(b) $\sigma = 0.18$, Accuracy=0.99453.

(c) $\sigma = 0.26$, Accuracy=0.94553.

(d) $\sigma = 0.30$, Accuracy=0.90442.

Fig. 3: Success rate of the three methods for 8-bit words and different noise levels.



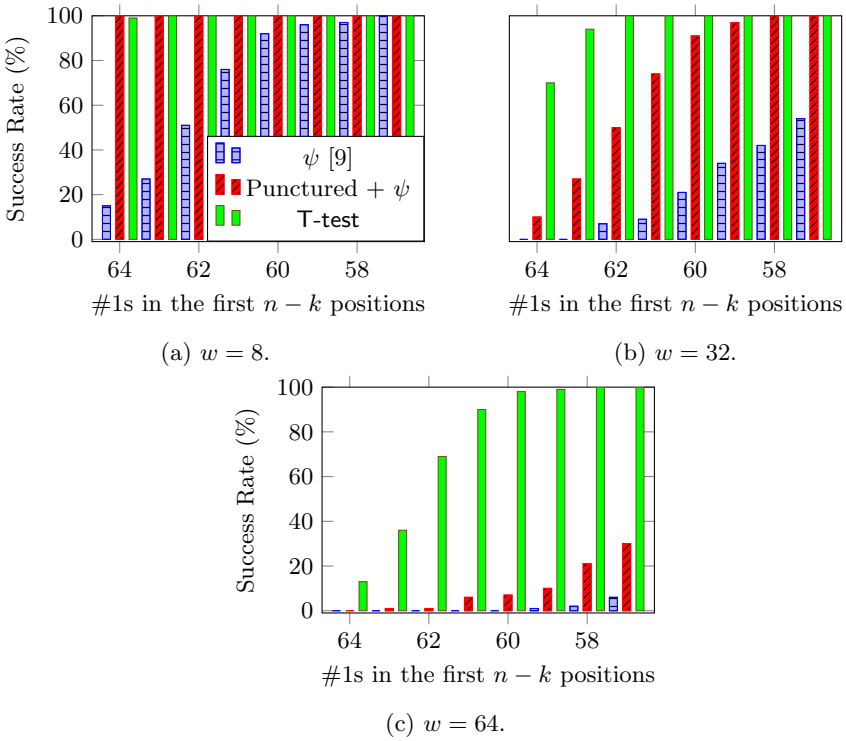
(a) $(n, k, t) = (3488, 2720, 64)$.

(b) $(n, k, t) = (8192, 6528, 128)$.

Fig. 4: Success rate of the T-test method for $w = 8$ and different noise levels for two *Classic McEliece* parameters sets.

noise level of $\sigma = 0.16$ that is close to the accuracy obtained on the real traces used in [9].

In Figure 5, as expected, the success rate of all three methods decreases when larger registers sizes are considered. However, in all cases, the proposed T-test-



(a) $w = 8$.

(b) $w = 32$.

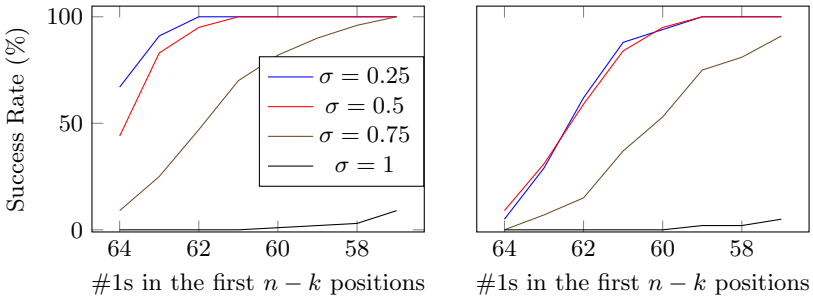
(c) $w = 64$.

Fig. 5: Comparison of the three methods for different register sizes at noise level $\sigma = 0.16$.

based method shows a better success rate than the CDCG method. We refer to Section 3.3 for a more detailed explanation of the impact of the dependent error on the CDCG method. The proposed method is also affected by larger register sizes, especially when noise increase as shown in Figure 6. For the noise levels considered in this figure, the punctured method does not manage to distinguish blocks with $\boldsymbol{e}_j = 0$ and $\boldsymbol{e}_j \neq 0$. Thus the T-test is the only solution when considering large registers and "high" noise scenarios.

6 Conclusion

In this paper, we analyze and develop techniques to solve the syndrome decoding problem with noisy information. In particular, we analyze some weaknesses of the method proposed in [9]. The weaknesses are due to the redefinition of the classical syndrome decoding problem into the integer syndrome decoding problem. We demonstrate that reformulating to integer syndrome decoding problem propagate errors due to side-channel acquisition.



(a) $w = 32$. (b) $w = 64$.

Fig. 6: Success rate of the T-test method for 32-bit and 64-bit words and different noise levels with $(n, k, t) = (3488, 2720, 64)$.

Next, we present two methods based on a divide-and-conquer approach, to avoid the propagation of the error. The methods presented are based on the fact that the distribution of the side-channel observations are different when a block of the vector $\boldsymbol{e}$ is 0 or not. The first method characterizes the distributions of the estimation according to the accuracy and finds the bound on the number of coordinates equal to 0 to distinguish if the block of the vector $\boldsymbol{e}$ is 0 or not. The second solution separate the rows of the matrix according to the side-channel leakages and evaluate if the rows seems to follow a uniform distribution in the two set or follow different distributions in the two sets. The analysis of the behavior of the two distribution is performed with a T-test. This allows us to discriminate inside the block which coordinate is more likely to follow a different distribution, allowing for an even finer analysis than the first method.

We finally validate our approach with various experiments. Both methods presented offer a better success rate than state-of-the-art attacks and the T-test is generally more efficient when considering larger registers or a higher level of noise. Compared with existing attack paths, this method cannot be used when the attacker obtains an integer syndrome only, without partial information, as done in [5]. In [9], the author suggests using masking as a countermeasure. An interesting research direction would be to evaluate the efficiency of the different approaches when masked implementations are considered.

All presented side-channel attack methods on KEM for code-based cryptography so far exploit profiling. An interesting research direction could be to turn these attacks into a non-profiled attack. Another path could be to adapt the technique to different rings or fields rather than the binary field considered. The specific structure of the public key in the BIKE cryptosystem, a quasi-cyclic moderate density parity check matrix, is not exploited in this work and deserves more investigations.

Acknowledgements

This work was funded by a French national grant managed by the Agence Nationale de la Recherche (ANR): project PQ-TLS reference ANR-22-PETQ-0008 through France 2023 program.

Appendix

Proof of Theorem 1

We deal here with a classical combinatorial urn process. It can be described as follows. We place t balls into $\frac{n}{w}$ urns, where the urns are labeled with respect to the number of balls contained in the urn. Hence, we can have urns labeled with integers from 0 to w. And we are interested in how many urns are labeled with the integer j, where $1 \leq j \leq w$. The number of possible $(i_2, \ldots, i_j)$ urns labeled with $(2, \ldots, j)$ equals

$$\binom{\frac{n}{w}}{i_2}\binom{\frac{n}{w}-i_2}{i_3}\cdots\binom{\frac{n}{w}-i_2-\cdots-i_{j-1}}{i_j}. \tag{6}$$

As there are $\frac{n}{w}-i_2-\cdots-i_j$ remaining urns, which are either labeled with 0 or with 1, and since there are a total of t balls from which $2i_2+\cdots+ji_j$ where already extracted, we can place the remaining balls in the remaining urns in $\binom{\frac{n}{w}-i_2-\cdots-i_j}{t-2i_2\cdots-ji_j}$ possible ways. This makes a total of

$$\binom{\frac{n}{w}}{i_2}\binom{\frac{n}{w}-i_2}{i_3}\cdots\binom{\frac{n}{w}-i_2-\cdots-i_{j-1}}{i_j}\binom{\frac{n}{w}-i_2-\cdots-i_j}{t-2i_2\cdots-ji_j} = \binom{\frac{n}{w}}{i_1,\ldots,i_j}. \tag{7}$$

with $i_1 = t-2i_2-\ldots ji_j$.

Now each urn labeled with j has $\binom{w}{j}$ possible representatives. Thus, we can deduce the number of positive cases which equal

$$\binom{\frac{n}{w}}{i_1,\ldots,i_j}\prod_{l=1}^{j}\binom{w}{l}^{i_l}.$$

Proof of Theorem 2

To prove Theorem 2 we will proceed step by step. We shall assume that errors are limited to a distance of 1 and overestimation and underestimation are equally probable, and the side-channel distinguisher accuracy equal to a.

Lemma 1. *Given* $\boldsymbol{e}_j = \mathbf{0}$ *we have*

$$\Pr\left(w_{\ell,1}=0\right) = a,\ \forall 1 \leq \ell \leq n-k,$$
$$\Pr\left(w_{\ell,j}-w_{\ell,j-1}=0\right) = a^2+\frac{(1-a)^2}{2}, \forall 1 \leq \ell \leq n-k, \forall 2 \leq j \leq \frac{n}{w}.$$

Proof. By definition of a we have $\Pr(w_{j,1} = 0) = a, \forall 1 \leq \ell \leq n-k$.

For the intermediate blocks $\Pr\left(w_{\ell,j} - w_{b_{\ell,j-1}} = 0\right)$ depends on the estimations at step $j-1$ and j. So, either both estimations are correct, with probability a^2, or both estimations are overestimated (resp. underestimates), with probability $\frac{1-a}{2}$.

Corollary 2. *Given* $\boldsymbol{e}_j = \mathbf{0}$ *we have* $HW(\boldsymbol{w}_1) \sim n-k-\mathcal{B}(n-k,a)$ *and* $HW(\boldsymbol{w}_j - \boldsymbol{w}_{j-1}) \sim n-k-\mathcal{B}\left(n-k, a^2+\frac{(1-a)^2}{2}\right)$.

Lemma 2. *Given* $HW(\boldsymbol{e}_j) = 1$ *we have*

$$\Pr(w_{\ell,1} = 0) = \frac{1+a}{4}, \quad \forall 1 \leq \ell \leq n-k,$$
$$\Pr(w_{\ell,j} - w_{\ell,j-1} = 0) = \frac{1+a^2}{4}, \quad \forall 1 \leq \ell \leq n-k,\ \forall 2 \leq j \leq \frac{n}{w}.$$

Proof. For the first block, without loss of generality, we assume that $\boldsymbol{e}_j(i) = 1$. We have two cases to obtain $w_{\ell,1} = 0$.

1. The i^{th} bit of the word of the matrix is 0, and we correctly estimate $w_{\ell,1}$, the probability is $\frac{a}{2}$.
2. The i^{th} bit of the word of the matrix is 1, and we underestimate $w_{\ell,1}$, probability $\frac{1-a}{4}$.

For the intermediate blocks, without loss of generality, we assume $\boldsymbol{e}_j(i) = 1$. Thus, we have two cases to obtain $w_{\ell,j} - w_{\ell,j-1} = 0$.

1. The i^{th} bit of the word of the matrix is 0, and we made the same error on both evaluations for j and $j-1$. Both correct, with probability $\frac{a^2}{2}$, both underestimated, with probability $\left(\frac{1-a}{2}\right)^2$, similar for both overestimated, with a probability $\left(\frac{1-a}{2}\right)^2$.
2. The i^{th} bit of the word of the matrix is 1.
   (a) The weight increases (resp. decreases), i.e. $\mathrm{HW}(b_{\ell,j}) = \mathrm{HW}(b_{\ell,j-1}) + 1$, we correctly estimate $w_{\ell,j}$ but underestimate (resp. overestimate) $w_{\ell,j-1}$ with a probability $\frac{1}{2}\frac{1-a}{2}a$ (resp. $\frac{1}{2}\frac{1-a}{2}a$).
   (b) Similarly, the error can be on $w_{\ell,j}$ overestimation or underestimation, and the difference will be zero depending on the impact on the weight modification, here also, we have two times probability of $\frac{1}{2}\frac{1}{2}\frac{1-a}{2}a$.

By summing all cases, we obtain the following:

$$\Pr(\widetilde{\mathrm{HW}}(b_{\ell,j}) - \widetilde{\mathrm{HW}}(b_{\ell,j-1}) = 0) = \frac{a^2}{2} + \left(\frac{1-a}{2}\right)^2 + 4\left(\frac{1}{4}\frac{1-a}{2}a\right).$$

Corollary 3. *Given* $HW(\boldsymbol{e}_j) = 1$ *we have* $HW(\boldsymbol{w}_1) \sim n-k-\mathcal{B}(n-k, \frac{1+a}{4})$ *and* $HW(\boldsymbol{w}_j - \boldsymbol{w}_{j-1}) \sim n-k-\mathcal{B}(n-k, \frac{1+a^2}{4})$.

Proposition 1. *Let* $a > \frac{1}{3} + \frac{40\log(n-k)}{9(n-k)} + \frac{8\sqrt{2}\sqrt{8\log(n-k)^2+3(n-k)\log(n-k)}}{9(n-k)}$. *Then,* $\Pr(X_1 > Y_1) \geq 1 - \frac{1}{(n-k)} - \frac{1}{e^{\mathcal{O}((3a-1)(n-k))}}$.

Moreover when $\boldsymbol{e}_1 = \mathbf{0}$ *we have* $HW(\boldsymbol{w}_1) \leq (n-k)(1-a) + \sqrt{2a(n-k)\log(n-k)}$.

Proof. Let us first recall that $X_1 = n - k - \mathrm{HW}(\boldsymbol{w}_1)$ given $\boldsymbol{e}_1 = \mathbf{0}$ and $Y_1 = n - k - \mathrm{HW}(\boldsymbol{w}_1)$ given $\mathrm{HW}(\boldsymbol{e}_1) = 1$. Also, by Lemma 1 $X_1 \sim \mathcal{B}(n-k, a)$ and by Lemma 2 $Y_1 \sim \mathcal{B}(n-k, \frac{1+a}{4})$. Let $\beta^* = (n-k)\frac{1+a}{4} + \beta$. This value will act as the separation between the distributions. More exactly we will require use the fact that

$$\Pr(X_1 > Y_1) \geq \Pr(X_1 > (n-k)\frac{1+a}{4} + \beta)\Pr(Y_1 < (n-k)\frac{1+a}{4} + \beta). \quad (8)$$

First we need to check the existence of such a value. For that we need to determine if such a positive integer β satisfying $(n-k)\frac{1+a}{4} + \sqrt{(n-k)\frac{1+a}{2}\log(n-k)} \leq (n-k)\frac{1+a}{4} + \beta \leq (n-k)a - \sqrt{2a(n-k)\log(n-k)}$ exists. By making the upper bound and lower bound equal, we obtain the wanted condition on a. This also implies that $\beta < (n-k)\frac{3a-1}{4} - \sqrt{2a(n-k)\log(n-k)}$.

Second, we will determine the probability in (8). Using Chernoff one gets

$$\Pr(Y_1 \geq (n-k)\frac{1+a}{4} + \beta) \leq e^{-\frac{\beta^2}{\frac{1+a}{2}(n-k)+\beta\frac{2}{3}}} \quad (9)$$

$$\Pr(X_1 \leq (n-k)\frac{1+a}{4} + \beta) \leq e^{-\frac{((n-k)a-(n-k)\frac{1+a}{4}-\beta)^2}{2a(n-k)}} \quad (10)$$

$$\Pr(X_1 > Y_1) \geq 1 - e^{-\frac{\beta^2}{\frac{1+a}{2}(n-k)+\beta\frac{2}{3}}} - e^{-\frac{((n-k)\frac{3a-1}{4}-\beta)^2}{2a(n-k)}}. \quad (11)$$

Putting $\beta = \frac{3a-1}{4}(n-k) - \sqrt{2a(n-k)\log(n-k)}$ in the previous equation we deduce $\Pr(X_1 > Y_1) \geq 1 - \frac{1}{e^{\log(n-k)}} - \frac{1}{e^{\mathcal{O}((3a-1)(n-k))}}$. The threshold value equals, $\beta^* = (n-k)a - \sqrt{2a(n-k)\log(n-k)}$.

Proposition 2. *Let* $a \geq 0.5$ *be a solution of the equation*

$$\sqrt{\frac{n-k}{\log(n-k)}} = \frac{4}{5a^2-4a+1}\left(\sqrt{(3a^2-2a+1)} - \sqrt{\frac{1+a^2}{2}}\right).$$

Then, $\Pr(X_1 > Y_1) \geq 1 - \frac{1}{(n-k)} - \frac{1}{e^{\mathcal{O}((3a-1)(n-k))}}$. *Moreover, when* $\boldsymbol{e}_j = \mathbf{0}$ *we have* $HW(\boldsymbol{w}_i - \boldsymbol{w}_{i-1}) \leq (n-k)(1-a^2-\frac{(1-a)^2}{2}) + \sqrt{(2a^2+(1-a)^2)(n-k)\log(n-k)}$.

The proof of this Proposition is identical to the previous one.

References

1. Aguilar Melchor, C., Aragon, N., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Persichetti, E., Zémor, G., Bos, J., Dion, A., Lacan, J., Robert, J.M., Veron, P.: HQC. Tech. rep., National Institute of Standards and Technology (2022), available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions`

2. Albrecht, M.R., Bernstein, D.J., Chou, T., Cid, C., Gilcher, J., Lange, T., Maram, V., von Maurich, I., Misoczki, R., Niederhagen, R., Paterson, K.G., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., Szefer, J., Tjhai, C.J., Tomlinson, M., Wang, W.: Classic McEliece. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-4-submissions
3. Aragon, N., Barreto, P., Bettaieb, S., Bidoux, L., Blazy, O., Deneuville, J.C., Gaborit, P., Gueron, S., Guneysu, T., Aguilar Melchor, C., Misoczki, R., Persichetti, E., Sendrier, N., Tillich, J.P., Zémor, G., Vasseur, V., Ghosh, S., Richter-Brokmann, J.: BIKE. Tech. rep., National Institute of Standards and Technology (2022), available at https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions
4. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, pp. 520–536. Springer (2012). https://doi.org/10.1007/978-3-642-29011-4_31, https://doi.org/10.1007/978-3-642-29011-4_31
5. Cayrel, P., Colombier, B., Dragoi, V., Menu, A., Bossuet, L.: Message-recovery laser fault injection attack on the Classic McEliece cryptosystem. In: Canteaut, A., Standaert, F. (eds.) Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12697, pp. 438–467. Springer (2021). https://doi.org/10.1007/978-3-030-77886-6_15, https://doi.org/10.1007/978-3-030-77886-6_15
6. Chari, S., Rao, J.R., Rohatgi, P.: Template attacks. In: Jr., B.S.K., Koç, Ç.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers. Lecture Notes in Computer Science, vol. 2523, pp. 13–28. Springer (2002). https://doi.org/10.1007/3-540-36400-5_3, https://doi.org/10.1007/3-540-36400-5_3
7. Chen, M., Chou, T.: Classic McEliece on the ARM cortex-M4. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2021**(3), 125–148 (2021). https://doi.org/10.46586/tches.v2021.i3.125-148, https://doi.org/10.46586/tches.v2021.i3.125-148
8. Chen, P., Chou, T., Deshpande, S., Lahr, N., Niederhagen, R., Szefer, J., Wang, W.: Complete and improved FPGA implementation of Classic McEliece. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022**(3), 71–113 (2022). https://doi.org/10.46586/tches.v2022.i3.71-113, https://doi.org/10.46586/tches.v2022.i3.71-113
9. Colombier, B., Dragoi, V., Cayrel, P., Grosso, V.: Profiled side-channel attack on cryptosystems based on the binary syndrome decoding problem. IEEE Trans. Inf. Forensics Secur. **17**, 3407–3420 (2022). https://doi.org/10.1109/TIFS.2022.3198277, https://doi.org/10.1109/TIFS.2022.3198277
10. Dragoi, V., Colombier, B., Cayrel, P., Grosso, V.: Integer syndrome decoding in the presence of noise. IACR Cryptol. ePrint Arch. p. 636 (2022), https://eprint.iacr.org/2022/636

11. Feige, U., Lellouche, A.: Quantitative group testing and the rank of random matrices. CoRR **abs/2006.09074** (2020), https://arxiv.org/abs/2006.09074
12. Guo, Q., Johansson, A., Johansson, T.: A key-recovery side-channel attack on Classic McEliece implementations. IACR Trans. Cryptogr. Hardw. Embed. Syst. **2022**(4), 800–827 (2022). https://doi.org/10.46586/tches.v2022.i4.800-827, https://doi.org/10.46586/tches.v2022.i4.800-827
13. Lahr, N., Niederhagen, R., Petri, R., Samardjiska, S.: Side channel information set decoding using iterative chunking - plaintext recovery from the "Classic McEliece" hardware reference implementation. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12491, pp. 881–910. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_29, https://doi.org/10.1007/978-3-030-64837-4_29
14. Lee, P.J., Brickell, E.F.: An observation on the security of McEliece's public-key cryptosystem. In: Günther, C.G. (ed.) Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings. Lecture Notes in Computer Science, vol. 330, pp. 275–280. Springer (1988). https://doi.org/10.1007/3-540-45961-8_25, https://doi.org/10.1007/3-540-45961-8_25
15. Lucks, S.: A variant of the Cramer-Shoup cryptosystem for groups of unknown order. In: Zheng, Y. (ed.) Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2501, pp. 27–45. Springer (2002). https://doi.org/10.1007/3-540-36178-2_2, https://doi.org/10.1007/3-540-36178-2_2
16. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 107–124. Springer (2011). https://doi.org/10.1007/978-3-642-25385-0_6, https://doi.org/10.1007/978-3-642-25385-0_6
17. O'Flynn, C., Chen, Z.D.: Chipwhisperer: An open-source platform for hardware embedded security research. In: Prouff, E. (ed.) Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers. Lecture Notes in Computer Science, vol. 8622, pp. 243–260. Springer (2014). https://doi.org/10.1007/978-3-319-10175-0_17, https://doi.org/10.1007/978-3-319-10175-0_17
18. Pircher, S., Geier, J., Zeh, A., Mueller-Gritschneder, D.: Exploring the RISC-V vector extension for the Classic McEliece post-quantum cryptosystem. In: 22nd International Symposium on Quality Electronic Design, ISQED 2021, Santa Clara, CA, USA, April 7-9, 2021. pp. 401–407. IEEE (2021). https://doi.org/10.1109/ISQED51717.2021.9424273, https://doi.org/10.1109/ISQED51717.2021.9424273
19. Prange, E.: The use of information sets in decoding cyclic codes. IRE Trans. Inf. Theory **8**(5), 5–9 (1962). https://doi.org/10.1109/TIT.1962.1057777, https://doi.org/10.1109/TIT.1962.1057777
20. Renauld, M., Standaert, F.: Algebraic side-channel attacks. In: Bao, F., Yung, M., Lin, D., Jing, J. (eds.) Information Security and Cryptology - 5th Inter-

national Conference, Inscrypt 2009, Beijing, China, December 12-15, 2009. Revised Selected Papers. Lecture Notes in Computer Science, vol. 6151, pp. 393–410. Springer (2009). https://doi.org/10.1007/978-3-642-16342-5_29, https://doi.org/10.1007/978-3-642-16342-5_29
21. Roth, J., Karatsiolis, E.G., Krämer, J.: Classic McEliece implementation with low memory footprint. In: Liardet, P., Mentens, N. (eds.) Smart Card Research and Advanced Applications - 19th International Conference, CARDIS 2020, Virtual Event, November 18-19, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12609, pp. 34–49. Springer (2020). https://doi.org/10.1007/978-3-030-68487-7_3, https://doi.org/10.1007/978-3-030-68487-7_3
22. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994. pp. 124–134. IEEE Computer Society (1994). https://doi.org/10.1109/SFCS.1994.365700, https://doi.org/10.1109/SFCS.1994.365700
23. Stern, J.: A method for finding codewords of small weight. In: Cohen, G.D., Wolfmann, J. (eds.) Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings. Lecture Notes in Computer Science, vol. 388, pp. 106–113. Springer (1988). https://doi.org/10.1007/BFb0019850, https://doi.org/10.1007/BFb0019850
24. Welch, B.L.: The generalization of 'STUDENT'S' problem when several different population variances are involved. Biometrika **34**(1-2), 28–35 (1947)
25. Winters, R.: Practical Predictive Analytics. Packt Publishing, Birmingham, England (2017), http://www.scholarvox.com/book/88842906
26. Zhang, Q., Wang, A., Niu, Y., Shang, N., Xu, R., Zhang, G., Zhu, L.: Side-channel attacks and countermeasures for identity-based cryptographic algorithm SM9. Secur. Commun. Networks **2018**, 9701756:1–9701756:14 (2018). https://doi.org/10.1155/2018/9701756, https://doi.org/10.1155/2018/9701756