Information-Set Decoding with Hints

Anna-Lena Horlemann[1], Sven Puchinger[2], Julian Renner[2], Thomas Schamberger[2], and Antonia Wachter-Zeh[2]

[1] School of Computer Science, University of St. Gallen (HSG), Switzerland
anna-lena.horlemann@unisg.ch
[2] Technical University of Munich (TUM), Munich, Germany
{sven.puchinger, julian.renner, t.schamberger, antonia.wachter-zeh}@tum.de*

Abstract. This paper studies how to incorporate small information leakages (called "hints") into *information-set decoding* (ISD) algorithms. In particular, the influence of these hints on solving the (n, k, t)-*syndrome-decoding problem* (SDP), i.e., generic syndrome decoding of a code of length n, dimension k, and an error of weight t, is analyzed. We motivate all hints by leakages obtainable through realistic side-channel attacks on code-based post-quantum cryptosystems. One class of studied hints consists of partial knowledge of the error or message, which allow to reduce the length, dimension, or error weight using a suitable transformation of the problem. As a second class of hints, we assume that the Hamming weights of subblocks of the error are known, which can be motivated by a template attack. We present adapted ISD algorithms for this type of leakage. For each third-round code-based NIST submission (Classic McEliece, BIKE, HQC), we show how many hints of each type are needed to reduce the work factor below the claimed security level. E.g., for Classic McEliece `mceliece348864`, the work factor is reduced below 2^{128} for 175 known message entries, 9 known error locations, 650 known error-free positions, or known Hamming weights of 29 subblocks of roughly equal size.

1 Introduction

Shortly after the proposal of the first public-key cryptosystem [12], Berlekamp *et al.* proved that decisional decoding in a random linear code is an NP-complete problem [6]. In the same year, McEliece designed the first encryption scheme that relies on the difficulty of the aforementioned problem. However, due to practical issues of the McEliece cryptosystem (i.e., relatively large key sizes), other systems were usually employed, e.g., schemes based on the hardness of factoring large integers and computing discrete logarithms. However, Shor's quantum algorithm [30] will be capable of factoring large integers and computing discrete logarithms efficiently as soon as capable quantum computer exist. Code-based cryptography is believed to resist attacks by quantum computers and thus, the Niederreiter-variant of the McEliece scheme is one of the four finalists in the

* This work was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under Grant No. WA3907/4-1 and SE2989/1-1 and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 801434). This work was partly done while S. Puchinger was with the Technical University of Denmark, Lyngby, Denmark, where he was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 713683.

third round of the *National Institute of Standards and Technology* (NIST) [23] post-quantum security standardization process. In addition, there are two alternative code-based candidates in the third round, namely BIKE and HQC.

Since the schemes in the third round seem to be secure from a theoretical point of view, more and more investigations with respect to their side-channel secure implementation are conducted. With a so called side-channel attack an attacker is able to exploit additional information obtained through observing the execution of a cryptographic algorithm in order to retrieve secret information, e.g., the private key. The two most prominent variants are timing attacks, where an attacker exploits execution time differences dependent on the secret, and power attacks, where the attacker measures the power consumption of the executing device as it is data dependent for CMOS logic. For the original McEliece proposal there are several attacks [36,32,14,21,35], including attacks against implementations that use a non-constant time Patterson decoder. For the submission *Classic McEliece*, a reaction attack utilizing an EM side-channel that leads to a full message recovery is shown in [16]. The submission BIKE is based on a quasi-cyclic code that allows for an optimized multiplication routine for the matrix vector multiplication during the syndrome computation. Published attacks [37,28,33] target this multiplication, where [37] successfully attacks a non-constant time variant and [28,33] target a constant time version of the multiplication proposed in [9]. The third code-based cryptosystem HQC has been attacked in [38,25], where the authors exploit the non-constant time implementation of the used BCH decoder. In [29] a successful power side-channel attack against a constant-time decoder (proposed as a countermeasure in [38]) is shown.

Most of these attacks are able to directly retrieve the entire secret key or plaintext. In [28], a tailored approach for specific partial leakage is given. In this paper, we provide a more general approach and show how arbitrary small information leakage or partial attack results can be incorporated into algorithms that solve the general decoding problem for linear codes and how this affects the security level of code-based cryptosystems. For that, we adapt the theoretical ideas of [11] from lattice-based systems to the code-based setting. More precisely, the authors of [11] propose a framework that can handle four different types of side information (called *hints*) to increase the efficiency of lattice reduction techniques. We propose a similar framework for code-based cryptography, where we translate concepts of these hints to the code-based scenario, introduce additional hints, and show how to transform the decoding problem accordingly.

Some previous works consider partial leakage. In [16,29] it was shown how partial information from attacks can be used to reduce the complexity of the underlying hard mathematical problem (namely, the *syndrome decoding problem* (SDP)). The reason for obtaining only partial information in these attacks is either a limitation on the obtainable side-channel observations as in [16] or certain private keys that can only partially be retrieved with the proposed attack as in [29]. The knowledge of some plaintext bits or some error bits, and its impact on the performance of *information set decoding* (ISD) was already considered, even if only briefly, in [8]. In [31] and [17], side-channel measurements (or timing attacks) are used to determine (some of) the error positions. In the former all error positions are found by measurements, whereas in the latter partial knowledge

of these positions is combined with an ISD algorithm. Furthermore, the latter algorithm is generalized to work with the knowledge of a set of indices containing some errors, or vice versa, the knowledge of error-free positions. In [24] two hints are considered: the error values come from a subset or the entries of the error vector are known, but their positions are not. They adapt an ISD algorithm to incorporate these hints. The cryptosystem set up in [13] uses error vectors whose sub-blocks are from a known set of short vectors. The corresponding generator matrices of the used error-correcting codes have a prescribed triangular block structure. Both these facts are used as partial knowledge in the ISD algorithm for cryptanalyzing this system in [13] and [22]. In [4] a zero-knowledge identification scheme was set up, using errors which live in a subset of the generally used finite field. For analyzing the strength of this scheme the authors set up several ISD algorithms exploiting the fact that the error entries were restricted.

In this paper, we provide a general framework to incorporate partial information leakage (hints) into ISD algorithms. The hints that we consider include general considerations like knowing parts of the message or a measurement of the message, some erroneous or error-free locations, or knowing the Hamming weight of blocks of the error. For each hint, we provide a motivation from the side-channel perspective in the respective chapter. As a general motivation, partial knowledge of the error or message allows an attacker to cope with restrictions on the maximum amount of side-channel observations or to simplify the attack by only allowing the retrieval of partial results in certain special cases. The hint of knowing the Hamming weight for certain error blocks is motivated by practical template attacks [10]. To reduce the amount of required side-channel observations, an attacker might choose to use Hamming weight templates instead of values templates. This might even be a necessity if more than 16-bit value templates have to be created for a successful attack. For each hint, we show how the SDP is transformed into an SDP with smaller parameters which therefore can be solved with smaller complexity. We apply these hints on Classic McEliece, HQC and BIKE and show how much leakage of each type is required to reduce the logarithmic work factor below the claimed security level.

2 Preliminaries

Let $[n] := \{1, 2, \ldots, n\}$ and let $\mathbb{F}_q$ denote the finite field of order q. Matrices and vectors are denoted by $\boldsymbol{A}$ and $\boldsymbol{a}$, respectively, and their entries are denoted by A_{ij} and a_i, respectively, where $i, j \geq 1$. To avoid confusion with the error vectors, $\boldsymbol{\delta}_i$ denotes the i-th unit vector. For a given set of indices I and a matrix $\boldsymbol{A}$, the matrix $\boldsymbol{A}_I$ denotes the submatrix of $\boldsymbol{A}$ containing the rows indexed by I. For a vector $\boldsymbol{x}$, $\boldsymbol{x}_I$ denotes the subvector containing the entries indexed by I.

The Hamming weight $\mathrm{wt_H}$ of a vector $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ is defined by $\mathrm{wt_H}(\boldsymbol{x}) = |\{x_i \neq 0 \mid i \in [n]\}|$. The parameters of a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of length n, dimension k and minimum distance d are denoted by $[n, k, d]_q$. Let $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ and $\boldsymbol{H} \in \mathbb{F}_q^{(n-k) \times n}$ be a generator matrix and a parity-check matrix, respectively, of an $[n, k, d]_q$-code $\mathcal{C}$. If $\boldsymbol{m} \in \mathbb{F}_q^k$ denotes a message and $\boldsymbol{e} \in \mathbb{F}_q^n$ an error vector, then $\boldsymbol{c} = \boldsymbol{m}\boldsymbol{G} \in \mathcal{C}$ is the corresponding codeword and $\boldsymbol{r} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$ is the corresponding received word. The corresponding syndrome $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$ with respect to $\boldsymbol{H}$ is given by $\boldsymbol{s} = \boldsymbol{r}\boldsymbol{H}^\top = \boldsymbol{e}\boldsymbol{H}^\top$. The goal of syndrome decoding is to

find the lowest-weight error vector $\boldsymbol{e}$ that fulfills the above syndrome equation. The generic (and decisional) version of this *syndrome decoding problem* is known to be NP-complete [6] and is the basis of most code-based cryptosystem. The computational version with prescribed weight t can be formulated as:

Definition 1 ((n,k,t)-Syndrome Decoding Problem (SDP)). *Given* $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$, $t \in \mathbb{N}$ *and* $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, *find* $\boldsymbol{e} \in \mathbb{F}_q^n$ *such that* $\mathrm{wt}_\mathrm{H}(\boldsymbol{e}) = t$ *and* $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$.

With the notation above, we can state the problem also in generator matrix form: Given $\boldsymbol{G}$ and $\boldsymbol{r}$, find $\boldsymbol{e}$ of weight t such that $\boldsymbol{r} - \boldsymbol{e} \in \mathcal{C}$. We can switch between generator and parity-check matrix via linear algebra, and can compute $\boldsymbol{s}$ from $\boldsymbol{r}$, and vice versa if we are only interested in $\boldsymbol{e}$ and not a specific codeword $\boldsymbol{r} - \boldsymbol{e}$. We need to be careful about the latter restriction in Section 3, where we assume that we have partial knowledge about $\boldsymbol{r} - \boldsymbol{e}$.

Throughout this paper, we exemplify our results with different parameter sets of all code-based submissions to the third round of the NIST standardization: Classic McEliece, BIKE, and HQC [2,3,1]. The security of Classic McEliece and BIKE can be reduced to an (n,k,t)-SDP through a message attack. For HQC, one can perform a key attack based on an algorithm that solves the (n,k,t)-SDP with additional information: A solution of the (n,k,t)-SDP has Hamming weights exactly $t/2$ in $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$, respectively, where $\boldsymbol{e}_1, \boldsymbol{e}_2 \in \mathbb{F}_2^{n/2}$ are two equally-sized blocks of the error $\boldsymbol{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2]$. The parameters (n,k,t) of the considered instances of SDP are as follows.

Parameter Set	n	k	t	Security Level	Used for
`mceliece348864`	3488	2720	64	NIST Cat. 1 (128)	Message attack
`mceliece6688128`	6688	5024	128	NIST Cat. 5 (256)	Message attack
`BIKE-Level-1`	24646	12323	134	NIST Cat. 1 (128)	Message attack
`BIKE-Level-5`	81946	40973	264	NIST Cat. 5 (256)	Message attack.
`hqc-128`	35338	17669	132	NIST Cat. 1 (128)	Key attack
`hqc-256`	115274	57637	262	NIST Cat. 5 (256)	Key attack

The table also contains the claimed security levels. Note that the parameters were designed with a security margin. E.g., Stern's information-set decoder (see below) has a work factor of $\approx 2^{146}$ for `mceliece348864`, which is ca. 18 bits above the claimed security level.

The idea of *information-set decoding* (ISD) was first introduced by Prange [26]. In this algorithm, in each iteration one chooses a random information set, reconstructs a message from the corresponding entries, then re-encodes this message and checks if the resulting codeword has Hamming distance at most t to the received word. The computational complexity is roughly $\binom{n}{t}/\binom{n-k}{t}(n-k)^2(n+1)$ binary operations. A more advanced variant is the one by Lee and Brickell [18], where some errors in the information set are possible. It partitions the errors in the information set and the error vector into two parts of prescribed weight each (moreover, one can also define a zero window in the error vector). Even more advanced, based on the birthday paradox, is Stern's ISD algorithm [34], speed up with intermediate sums and early abort techniques, which we will use to attack the various cryptosystems with the hints described in the following

sections. We remark that other generic algorithms like MMT [19] of BJMM [5] might be slightly faster than Stern, however, we will restrict ourselves to Stern.

3 Hint-Based Parameter Reduction in the SDP

We introduce a class of hints and present, for all hints, generic transformations that reduce the SDP parameters. Hence, any ISD algorithm can be applied to the transformed instance to reduce the work factor. We exemplify the impact of these hints on the NIST proposals in Figure 1 below and determine the number of hints needed to reduce the security levels below the claimed level. In Appendix B, we show how different hints can be combined in any order, analogous to the hints considered in the lattice setting in [11].

3.1 Known Error Locations or Error-free Locations

First, we treat the setting where we gain some knowledge about the error vector: (i) either we know that a given position contains an error (and here we distinguish again if we also know the error value or not), or (ii) that the position is error-free. The case of knowing an error location in the binary case and its implication on Classic McEliece was already studied in [17]; however, for comparison we include it in this section. These types of hints can be obtained from attacks that directly target the error. In [16], an attack on Classic McEliece is shown for which an EM side-channel oracle is constructed in order to distinguish if decoding is successful. Using this oracle the authors successively add an error to each position of the ciphertext and observe the oracle for each decryption. If the position was error-free, an additional error is induced, which results in an error during decoding indicated by the oracle. The authors provide a trade-off between required attack measurements and remaining attack complexity by utilizing the partial information (only a part of the error-locations are known) in an ISD algorithm. A similar attack against HQC is shown in [29], where the authors retrieve the private key by observing the decoding result in the power side-channel for specially crafted ciphertexts. In the case of a successful attack, the exact error positions of the secret key are known, but the overall attack success is dependent on the support of error. The authors therefore provide an attack that is able to retrieve the exact error positions of only a large part of possible keys. For the remaining keys, the error positions can only be partially retrieved. Nevertheless, these error positions can again be interpreted as hints.

Hint Type 1 *Consider an (n, k, t)-SDP with given parity check matrix $\boldsymbol{H}$ and syndrome $\boldsymbol{s}$, which has a solution $\boldsymbol{e}$ such that $\mathrm{wt}_{\mathrm{H}}(\boldsymbol{e}) = t$ and $\boldsymbol{e}\boldsymbol{H}^{\top} = \boldsymbol{s}$.*
Given: *An error entry $e_j \in \mathbb{F}_q \backslash \{0\}$.*

Theorem 1. *Using Hint Type 1, any (n, k, t)-SDP can be transformed into an $(n-1, k-1, t-1)$-SDP or an $(n-1, k, t-1)$-SDP. For a random code the former happens with high probability.*

Proof. If we know the error value $\boldsymbol{e}_j$, then we can shorten the code in the j-th position and solve the new syndrome equation $\boldsymbol{e}_{[n]\backslash j}(\boldsymbol{H}_{[n]\backslash j})^{\top} = \boldsymbol{s} - \boldsymbol{e}_j \boldsymbol{H}_j^{\top}$, where $\boldsymbol{e}_{[n]\backslash j}$ has Hamming weight $t - 1$. For a random code and large n, the corresponding shortened code has dimension $k - 1$ with high probability, and we hence have a $(n - 1, k - 1, t - 1)$-SDP; otherwise the shortened code has dimension k, and we hence have a $(n - 1, k, t - 1)$-SDP. □

Hint Type 2 *Consider an (n,k,t)-SDP with given parity check matrix $\boldsymbol{H}$ and received word $\boldsymbol{r}$, which has a solution $\boldsymbol{e}$ such that $\mathrm{wt_H}(\boldsymbol{e}) = t$ and $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{r}\boldsymbol{H}^\top$.*
Given: *An error location j with $e_j \neq 0$.*

Theorem 2. *Using Hint Type 2, any (n,k,t)-SDP can be transformed into an $(n-1,k,t-1)$-SDP or an $(n-1,k-1,t-1)$-SDP. For a random code the former happens with high probability.*

Proof. If we know an error location j, but not the corresponding error value, then we can puncture the code in the j-th position, compute the corresponding parity check matrix $\boldsymbol{H}' \in \mathbb{F}_q^{(n-k-1)\times(n-1)}$ and solve the new syndrome equation $\boldsymbol{e}_{[n]\setminus j}(\boldsymbol{H}')^\top = \boldsymbol{r}_{[n]\setminus j}(\boldsymbol{H}')^\top$, where $\boldsymbol{e}_{[n]\setminus j}$ has Hamming weight $t-1$. As above, this leads to an $(n-1,k,t-1)$-SDP or an $(n-1,k-1,t-1)$-SDP. Then, the value e_j can be found by simple erasure decoding, e.g., by Gaussian elimination. □

If we know several, say ϵ many, error locations, indexed by the set $I \subset [n]$, we can extend the above ideas by shortening or puncturing the code in I.

Corollary 1. *With 9 known error locations (i.e., about 14%) the security level of `mceliece348864` reduces to less than 128 bits. For `mceliece6688128` the security level reduces to less than 256 bits with 4 known error locations (i.e., about 3%).*

Hint Type 3 *Consider an (n,k,t)-SDP with given parity check matrix $\boldsymbol{H}$ and syndrome $\boldsymbol{s}$, which has a solution $\boldsymbol{e}$ such that $\mathrm{wt_H}(\boldsymbol{e}) = t$ and $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}^\top$.*
Given: *An error-free location, i.e., an index j such that $e_j = 0$.*

Theorem 3. *Using Hint Type 3, any (n,k,t)-SDP can be transformed into an $(n-1,k-1,t)$-SDP or an $(n-1,k,t)$-SDP. For a random code the former happens with high probability.*

Proof. The proof is analogous to the one of Theorem 1, with $e_j = 0$. □

Corollary 2. *With 652 known error-free locations (i.e., about 19%) the security level of `mceliece348864` reduces to less than 128 bits security, using the reduction from Theorem 3. For `mceliece6688128` the security level reduces to less than 256 bits with 249 known error-free locations (i.e., about 4%).*

The reduction from Theorem 4 can also be applied to known error-free locations. For *Classic McEliece* this provides a stronger reduction of the security level.

3.2 Measurement of the Error

Any linear operation on the error vector entries that can be observed exactly (i.e., the exact result of the operation can be obtained), can be in principle used for this hint. For instance, if we can obtain intermediate results of a chunk-based syndrome computation in Niederreiter variants of the McEliece cryptosystem (as in the reference implementation of Classic McEliece [2]), we obtain measurements of the error directly. See Appendix D for more details.

Hint Type 4 *Consider an (n,k,t)-SDP with given parity check matrix $\boldsymbol{H}$ and syndrome $\boldsymbol{s}$), which has a solution $\boldsymbol{e}$ such that $\mathrm{wt_H}(\boldsymbol{e}) = t$ and $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}^\top$.*
Given: *A* measurement vector $\boldsymbol{v} \in \mathbb{F}_q^n \setminus \{\boldsymbol{0}\}$, *s. t.* $\boldsymbol{v} \notin \mathcal{C}^\perp$, *and* $\boldsymbol{e}\cdot\boldsymbol{v}^\top = \sigma \in \mathbb{F}_q$.

Recall that the kernel of $\boldsymbol{H}$ is the code $\mathcal{C}$. Hence, the condition $\boldsymbol{v} \notin \mathcal{C}^\perp$ is equivalent to the condition that $\boldsymbol{v}$ and the rows of $\boldsymbol{H}$ are linearly independent.

Theorem 4. *Using Hint Type 4, any (n, k, t)-SDP can be transformed into an $(n, k-1, t)$-SDP.*

Proof. Assume a *measurement vector* $\boldsymbol{v} \in \mathbb{F}_q^n$ is given such that $\boldsymbol{v} \notin \mathcal{C}^\perp$ and $\boldsymbol{e} \cdot \boldsymbol{v}^\top = \sigma$, then we can extend the original syndrome equation to $\bar{\boldsymbol{s}} := [\boldsymbol{s} \;\; \sigma] = \boldsymbol{e}[\boldsymbol{H}^\top \;\; \boldsymbol{v}] = \boldsymbol{e}\bar{\boldsymbol{H}}^\top$. Hence, we obtain a new parity-check matrix $\bar{\boldsymbol{H}}^\top \in \mathbb{F}_q^{(n-k+1)\times n}$ and a corresponding syndrome $\bar{\boldsymbol{s}} \in \mathbb{F}_q^{n-k+1}$. Since $\boldsymbol{v} \notin \mathcal{C}^\perp$, we have $\operatorname{rank}(\bar{\boldsymbol{H}}) = n-k+1$, so $\bar{\boldsymbol{H}}$ is a parity-check matrix of a subcode of $\mathcal{C}$ of dimension $k-1$. □

When combining several measurements for the error, one needs to check that all of them, together with the rows of $\boldsymbol{H}$, are linearly independent. Otherwise, not all of them will reduce the SDP to a smaller instance.

Corollary 3. *With* 175 *known linearly independent error measurements, the security level of* `mceliece348864` *reduces to less than* 128 *bits security. For* `mceliece6688128` *the security level reduces to less than* 256 *bits with* 64 *linearly independent error measurements.*

A special instance of an error measurement is the knowledge of an entry of the error vector, by setting $\boldsymbol{v}$ as the unit vector with the only non-zero entry in the error location, and σ as the error value. It depends on the code parameters and the generic decoder used, which reduction is more useful, see also Figure 1.

3.3 Known Partial Message

This hint is only applicable for the SDP given in its generator matrix form, where we consider $\boldsymbol{G}$ instead of $\boldsymbol{H}$ and $\boldsymbol{r}$ instead of $\boldsymbol{s}$ as, e.g., in the original paper of the McEliece cryptosystem [20], where the ciphertext is of the form $\boldsymbol{r} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$. In this system an attacker may obtain information about the message $\boldsymbol{m}$ by observing the encryption. A possible attack vector is a template attack on the load operations of the different parts of the message. E.g., the correct retrieval of an 8-bit part of the message gives a total of eight hints.

Hint Type 5 *Consider an (n, k, t)-SDP in generator matrix form (given $\boldsymbol{G}$ and $\boldsymbol{r} = \boldsymbol{m}\boldsymbol{G} + \boldsymbol{e}$), which has a solution $\boldsymbol{e}$ with $\mathrm{wt}_\mathrm{H}(\boldsymbol{e}) = t$.*
Given: *A message entry $m_j \in \mathbb{F}_q$.*

Theorem 5. *Using Hint Type 5, any (n, k, t)-SDP can be transformed into an $(n, k-1, t)$-SDP.*

Proof. If an entry m_j of the message $\boldsymbol{m}$ is known, we can reduce the (n, k, t)-SDP to an $(n, k-1, t)$-SDP. This reduction can be done by taking the subcode corresponding to the unknown message bits, computing the corresponding parity check matrix of size $(n-k+1) \times n$ and syndrome-decoding the modified received word $\boldsymbol{r}' := \boldsymbol{r} - m_j\boldsymbol{G}_j$ with respect to the $(n-k+1) \times n$ parity-check matrix. □

Note that, although coming from different types of hints, the reductions in Theorem 4 and Theorem 5 are mathematically the same.

Corollary 4. *With* 175 *known message entries (about* 6.5%*) the security level of* `mceliece348864` *reduces to less than* 128 *bits. For* `mceliece6688128` *the security level reduces to less than* 256 *bits with* 64 *known message entries (about* 1%*).*

3.4 Measurement of the Message

The motivation of this hint works analog to the measurement of the error message described in Subsection 3.2. This means that the knowledge of every exact result of a linear operation on the message vector entries can be used for this hint. If the matrix vector operation $\boldsymbol{mG}$ is again implemented as a chunk-based multiplication, a successful template attack on the final multiplication result of one chunk can be used as this hint.

Hint Type 6 *Consider an (n, k, t)-SDP in generator matrix form (given $\boldsymbol{G}$ and $\boldsymbol{r} = \boldsymbol{mG} + \boldsymbol{e}$), which has a solution $\boldsymbol{e}$ with $\mathrm{wt_H}(\boldsymbol{e}) = t$.*
Given: *A measurement vector $\boldsymbol{v} \in \mathbb{F}_q^k \backslash \{\boldsymbol{0}\}$ such that $\boldsymbol{m} \cdot \boldsymbol{v}^\top = \sigma \in \mathbb{F}_q$.*

Theorem 6. *Using Hint Type 6, any (n, k, t)-SDP can be transformed into an $(n, k-1, t)$-SDP.*

Proof. If we have $\boldsymbol{m} \cdot \boldsymbol{v}^\top = \sigma$, we can use the fact that there is some $\boldsymbol{A} \in GL_k$ such that $\boldsymbol{Av}^\top = \boldsymbol{\delta}_k$, which gives $\boldsymbol{mA}^{-1} \cdot \boldsymbol{Av}^\top = \bar{\boldsymbol{m}} \cdot \boldsymbol{\delta}_k^\top = \bar{\boldsymbol{m}}_k = \sigma$, where $\bar{\boldsymbol{m}} := \boldsymbol{mA}^{-1}$. Then, we get $\boldsymbol{mG} = \bar{\boldsymbol{m}}(\boldsymbol{AG})$ and $\bar{\boldsymbol{m}}_k = \sigma$. We only need to find $\bar{\boldsymbol{m}}_{[k-1]}$ in the code $\bar{\mathcal{C}}$ generated by $\bar{\boldsymbol{G}} := (\boldsymbol{AG})_{[k-1]}$ w.r.t. the transformed received word $\bar{\boldsymbol{r}} := \boldsymbol{r} - \sigma(\boldsymbol{AG})_k$. This gives an $(n, k-1, t)$-SDP. □

Note that the reduction from Theorem 6 is mathematically again the same as the one in Theorem 4 and Theorem 5. When several measurement vectors are given that span an ϵ-dimensional subspace, the (n, k, t)-SDP can be reduced to an $(n, k-\epsilon, t)$-SDP, with the same techniques as above.

Corollary 5. *With 175 linearly independent message measurements, the security level of `mceliece348864` reduces to less than 128 bits security. For `mceliece6688128`, the security level reduces to less than 256 bits with 64 linearly independent message measurements.*

Note that measurement of the message vector is a generalization of knowing a message symbol. Here, the two hints lead to the same reduction, in contrast to the analog measurement/known entry of the error.

Figure 1 exemplifies the work factor reduction of Stern's algorithm (work factor formula as in [15]) for all discussed hints, for Classic McEliece and BIKE.

4 Hamming Weights of Error Blocks Known

We mathematically model this type of leakage as follows: Partition the set $\mathcal{W} = \{1, \ldots, n\}$ into subsets (called blocks) $\mathcal{W}_i$ of cardinality η_i for $i = 1, \ldots, \ell$, and write $\boldsymbol{\eta} := [\eta_1, \ldots, \eta_\ell]$. We assume in the following that we know the exact weight decomposition $\boldsymbol{t}$ of the errors into the sets $\mathcal{W}_i$, i.e., the Hamming weights $t_i := \mathrm{wt_H}(\boldsymbol{e}_{\mathcal{W}_i})$ of the errors restricted to the sets $\mathcal{W}_i$, for all $i = 1, \ldots, \ell$. These types of hints are motivated by template attacks on operations containing the error, e.g., load and store operations of continuous bits (blocks) of the error vector. If the size of simultaneously processed error bits becomes to large (a conservative estimate would be larger 16 bit), the profiling step for all possibilities of values is not practical anymore. In this case, and also to reduce the attack complexity for smaller template sizes, an attacker can opt to use only Hamming weight templates. In this case, a successful template matching reveals the Hamming

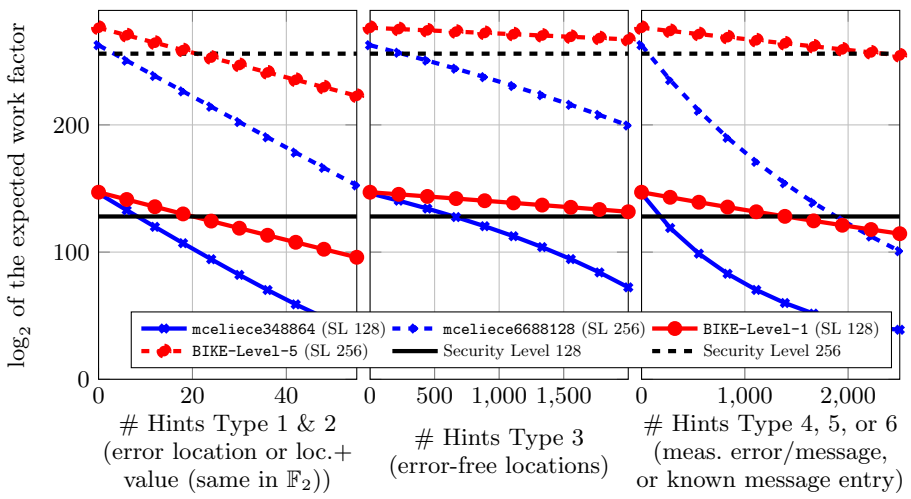

Fig. 1. Influence of hints on work factor of Stern's ISD algorithm, exemplified for two parameter sets each of the NIST round 3 submissions Classic McEliece and BIKE.

weight of the targeted block. An example of such an attack can be given with the HQC cryptosystem. The first step during the HQC decryption consists of the subtraction $\boldsymbol{v} - \boldsymbol{u}\boldsymbol{y}$, where $\boldsymbol{v}$ and $\boldsymbol{u}$ define the ciphertext and $\boldsymbol{y}$ is a very spare error vector and the private key. By fixing $\boldsymbol{v} = [0 \dots 0]$ and $\boldsymbol{u} = [1\ 0 \dots 0]$ the private key is subtracted from a zero vector. A template attack using Hamming weight templates is then able to retrieve the Hamming weight of each subtraction block (depends on the register size of the target platform).

In the following, we adapt well-known ISD algorithms to include this information leakage and describe how this reduces the security level.

4.1 Prange's ISD Algorithm with Known Block Weights

For didactic reasons, we start with Prange's information-set decoder. Based on the knowledge of the Hamming weight of the error blocks, we adapt the strategy of choosing an information set as, thereby (for known weight decomposition $\boldsymbol{t}$) increasing the probability that the information set is error-free. For a given $\boldsymbol{t}$, we fix a vector $\boldsymbol{x} \in \left\{[x_1, \dots, x_\ell] \in \mathbb{Z}^\ell : 0 \le x_i \le \eta_i - t_i, \quad \sum_i x_i = k\right\}$. As described in the following, we have to choose $\boldsymbol{x}$ carefully since the work factor of our decoder will depend on it. Then, in each iteration of the algorithm, we choose independently and uniformly at random a subset $\mathcal{X}_i \subseteq \mathcal{W}_i$ of cardinality x_i for each i, i.e., x_i positions from each block. Since $\sum_i x_i = k$, the union $\mathcal{X} = \cup_i \mathcal{X}_i$ has cardinality k. Then, we proceed as in the original Prange algorithm: We check if $\mathcal{X}$ is an error-free information set.

Theorem 7. *For a given $\boldsymbol{x}$, Algorithm 1 has expected work factor of $W_{\mathsf{Prange}} = \frac{W_{\mathsf{Prange,Iter}}}{P_{\mathsf{Prange}}}$, where $W_{\mathsf{Prange,Iter}} = (n-k)^2(n+1)$ (cost of one iteration) and $P_{\mathsf{Prange}} = \prod_{i=1}^{\ell} \binom{\eta_i - x_i}{t_i} \binom{\eta_i}{t_i}^{-1}$ (success probability).*

Proof. See Appendix E. □

Algorithm 1: Prange with Known Block Weights

Input: ISD problem (in form $\boldsymbol{r}$ and $\boldsymbol{G}$, cf. discussion below Definition 1) and weight decomposition $\boldsymbol{t}$ of the error, vector $\boldsymbol{x}$
Output: Error $\boldsymbol{e}$
do
  $\mathcal{X} \leftarrow \cup_{i=1}^{\ell} \mathcal{X}_i$, where the $\mathcal{X}_i$ are chosen independently uniformly from the subsets of $\mathcal{W}_i$ of cardinality x_i
  $\boldsymbol{e} \leftarrow \boldsymbol{r} - \boldsymbol{r}_{\mathcal{X}} (\boldsymbol{G}^{\mathcal{X}})^{-1} \boldsymbol{G}$ `// Same as in original Prange alg.`
while $\mathrm{wt_H}(\boldsymbol{e}) \neq \sum_i t_i$
return $\boldsymbol{e}$

Since $\boldsymbol{\eta}$ and $\boldsymbol{t}$ are given, we can influence the work factor of Algorithm 1 by choosing $\boldsymbol{x}$ in a suitable way. The following greedy algorithm maximizes the success probability for given $\boldsymbol{\eta}$ and $\boldsymbol{t}$:

- Initially, choose $x_i = \eta_i - t_i$ for all $i = 1, \ldots, \ell$
- While $\sum_i x_i > k$, decrease the x_j (by one) that increases P_{Prange} the most.

The greedy choice leads to a global maximum of P_{Prange} since the x_i only influence distinct factors of the product, so the increase of P_{Prange} resulting from decreasing one x_j does not influence the relative increase of P_{Prange} of other $x_i \neq x_j$ in the next steps. Note that the algorithm ensures that $x_i \leq \eta_i - t_i$ for all i, i.e., that the success probability P_{Prange} is non-zero. We will see in Section 4.3 that this choice of $\boldsymbol{x}$ reduces the work factor significantly compared to the original Prange algorithm for a growing number of blocks.

4.2 Stern's ISD Algorithm with Known Block Weights

Stern's algorithm uses two parameters p and ν to choose an information set in each round. It allows a fixed number of errors in the information set and additionally restricts the number of errors outside the information set. Stern's algorithm divides the information set into two equal-size subsets $\mathcal{X}$ and $\mathcal{Y}$ and looks for words of weight p at the indices of $\mathcal{X}$ weight p at the indices $\mathcal{Y}$, and weight 0 on a fixed uniform random set $\mathcal{Z}$ of ν positions outside the information set. Hence, we need to choose three sets, $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, at random. Again, we adapt the choice of these sets to the known weight distribution by designing three vectors $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ that indicate how many positions we choose for the three sets, respectively, from each block. A heuristic choice of the vectors is discussed below.

Theorem 8 states the expected work factor of the adapted Stern algorithm. The formula for the success probability in (1) consists of sums whose number of summands may grow exponentially in the parameters p and ℓ. However, we can compute it in polynomial time using dynamic programming similar to [27].

Theorem 8. *For given $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ and parameters p, ν, Algorithm 2 has an expected work factor of $W_{\mathsf{Stern}} := \frac{W_{\mathsf{Stern,Iter}}}{P_{\mathsf{Stern}}}$, where the cost per iteration is given by*

$$W_{\mathsf{Stern,Iter}} = (n-k)^2(n+1) + \nu\big(L(m_x, p) + L(m_y, p) - m_x - m_y + \tbinom{m_y}{p}\big) + \frac{\tbinom{m_x}{p}\tbinom{m_y}{p}}{2^{\nu-1}}(t - 2p + 1)(2p + 1),$$

Algorithm 2: Stern with Known Block Weights

Input: ISD problem and weight decomposition, $\boldsymbol{t}$ of the error, vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$, parameters p, ν (ν must be also the sum of the entries of $\boldsymbol{z}$).
Output: Error $\boldsymbol{e}$
do
  $\mathcal{X} \leftarrow \cup_{i=1}^{\ell} \mathcal{X}_i$, where the $\mathcal{X}_i$ are chosen independently uniformly from the subsets of $\mathcal{W}_i$ of cardinality x_i
  $\mathcal{Y} \leftarrow \cup_{i=1}^{\ell} \mathcal{Y}_i$, where the $\mathcal{Y}_i$ are chosen independently uniformly from the subsets of $\mathcal{W}_i \setminus \mathcal{X}_i$ of cardinality y_i
  $\mathcal{Z} \leftarrow \cup_{i=1}^{\ell} \mathcal{Z}_i$, where the $\mathcal{Z}_i$ are chosen independently uniformly from the subsets of $\mathcal{W}_i \setminus (\mathcal{X}_i \cup \mathcal{Y}_i)$ of cardinality z_i
  $\boldsymbol{e} \leftarrow$ Iteration of original Stern alg. w.r.t. sets $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ and parameters p, ν
while *Stern stopping condition not satisfied for* $\boldsymbol{e}$
return $\boldsymbol{e}$

where $L(x,y) := \sum_{i=1}^{y} \binom{x}{i}$, $m_x := \sum_i x_i$, *and* $m_y := \sum_i y_i$.

The success probability of each iteration is given by

$$P_{\mathsf{Stern}} = \sum_{\substack{\boldsymbol{a} \in \mathbb{Z}^{\ell} \\ 0 \le a_i \le t_i \\ \sum_i a_i = p}} \sum_{\substack{\boldsymbol{b} \in \mathbb{Z}^{\ell} \\ 0 \le b_i \le t_i - a_i \\ \sum_i b_i = p}} \prod_{i=1}^{\ell} \frac{\binom{x_i}{a_i}\binom{y_i}{b_i}\binom{\eta_i - x_i - y_i - z_i}{t_i - a_i - b_i}}{\binom{\eta_i}{t_i}}, \tag{1}$$

and can be computed in polynomial bit complexity (in the parameters n, k, ℓ, p*).*

Proof. See Appendix E. □

Again, the question is how to choose the vectors $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ for given $\boldsymbol{\eta}$ and $\boldsymbol{t}$. We propose the following heuristic:

- Choose a vector $\tilde{\boldsymbol{x}}$ with $\sum_i \tilde{x}_i = k$ as in the heuristic for Prange's decoder. Our goal is to choose as the information set (union of $\mathcal{X}$ and $\mathcal{Y}$) with exactly $\tilde{x}_i$ positions from the i-th block.
- Choose $x_i, y_i \approx \frac{\tilde{x}_i}{2}$, e.g., alternatingly rounded down/up for odd $\tilde{x}_i$ such that we have $\sum_i x_i \approx \sum_i y_i \approx k/2$. The fact that we take roughly the same number of entries from the information subset in the i-th block means that the probability that $\mathcal{X}$ and $\mathcal{Y}$ contain exactly p errors is roughly the same.
- Choose $z_i = 0$ for all $i = 1, \ldots, \ell$. While $\sum_i z_i < \nu$, increase (by one) the z_j that maximizes P_{Stern}.

4.3 Numerical Results and Comparison

We present numerical results for the work factors of the modified Prange, Lee–Brickell (see Appendix F), and Stern algorithms. All plots show logarithmic work factors as a function of the number of blocks ℓ. Figures 2, 3, and 4 (see Appendix A for the latter two) contain the curves for the parameter sets of Classic McEliece, BIKE, and HQC, respectively, which we list in the preliminaries.

We use the presented heuristics to choose $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$, and optimize over the parameters p and ν. If $\ell \nmid n$, we choose $\eta_i \approx n/\ell$ rounded up or down in a suitable ratio. Since the work factors depend heavily on the weight distribution to the blocks, we randomly sample for each parameter set several errors (uniformly at random from the set of errors of weight t) and present realizations as points, plus curves for the log of the mean work factor. The number of realizations for each ℓ and algorithm is roughly 50.

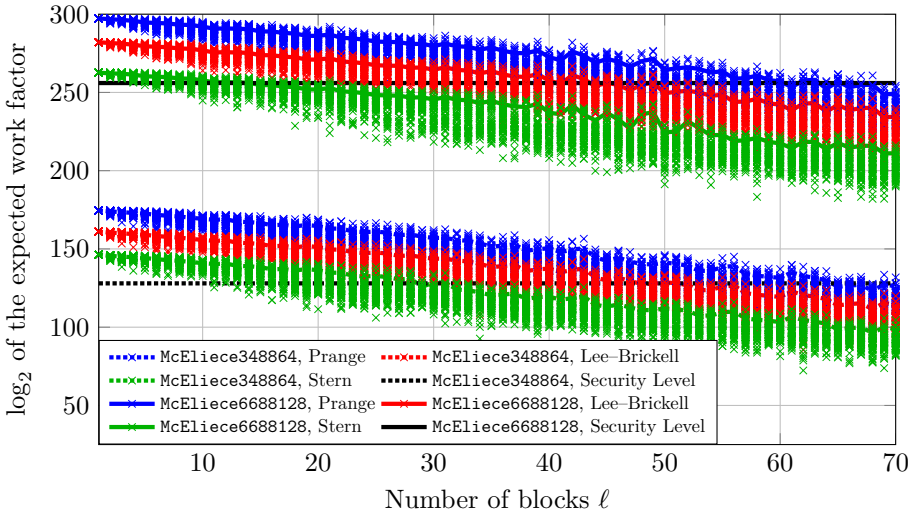

Fig. 2. Work factors of the modified Prange, Lee–Brickell, and Stern algorithms as a function of the number of blocks for which the weight distribution is known, for two Classic McEliece parameter sets. Lines are means, points are realizations.

It can be seen that for all systems, parameter sets, and algorithms, only a few blocks are needed to push the work factor below the claimed security level. For instance, in `mceliece348864`, Stern's algorithm only needs to know the weight decomposition for 11 blocks (of size $\eta_i \approx 317$) to get below the security level for *some* weight decompositions (i.e., realizations of the error), and 29 blocks (of size $\eta_i \approx 120$) to push the *mean* work factor below the claimed security level.

References

1. C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. Bos, J. Deneuville, A. Dion, P. Gaborit, J. Lacan, E. Persichetti, J. Robert, P. Véron, and G. Zémor, "Hamming Quasi-Cyclic (HQC)," *Third round submission to the NIST post-quantum cryptography call*, 2019. [Online]. Available: https://pqc-hqc.org
2. M. R. Albrecht, D. J. Bernstein, T. Chou, C. Cid, J. Gilcher, T. Lange, V. Maram, I. von Maurich, R. Misoczki, R. Niederhagen, K. G. Paterson, E. Persichetti,

C. Peters, P. Schwabe, N. Sendrier, J. Szefer, C. J. Tjhai, M. Tomlinson, and W. Wang, "Classic McEliece," *Third round submission to the NIST post-quantum cryptography call*, 2019. [Online]. Available: https://classic.mceliece.org

3. N. Aragon, P. S. L. M. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Ghosh, S. Gueron, T. Güneysu, C. Aguilar-Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor, "BIKE: Bit Flipping Key Encapsulation," *Third round submission to the NIST post-quantum cryptography call*, 2019. [Online]. Available: https://bikesuite.org/
4. M. Baldi, M. Battaglioni, F. Chiaraluce, A.-L. Horlemann-Trautmann, E. Persichetti, P. Santini, and V. Weger, "A new path to code-based signatures via identification schemes with restricted errors," 2020.
5. A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding," in *Advances in Cryptology - EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds. Springer Verlag, 2012, vol. 7237, pp. 520–536.
6. E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978.
7. D. J. Bernstein, T. Lange, and C. Peters, "Smaller decoding exponents: ball-collision decoding," in *Annual Cryptology Conference*. Springer, 2011, pp. 743–760.
8. A. Canteaut, "A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense bch codes of length 511," *IEEE Transactions on Information Theory*, vol. 44, pp. 367–378, 1998.
9. T. Chou, "QcBits: Constant-time small-key code-based cryptography," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2016, pp. 280–300.
10. M. O. Choudary and M. G. Kuhn, "Efficient, portable template attacks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 490–501, feb 2018.
11. D. Dachman-Soled, L. Ducas, H. Gong, and M. Rossi, "Lwe with side information: Attacks and concrete security estimation," in *Advances in Cryptology – CRYPTO 2020*, D. Micciancio and T. Ristenpart, Eds. Cham: Springer International Publishing, 2020, pp. 329–358.
12. W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.
13. D. Gligoroski, S. Samardjiska, H. Jacobsen, and S. Bezzateev, "Mceliece in the world of escher," Cryptology ePrint Archive, Report 2014/360, 2014.
14. S. Heyse, A. Moradi, and C. Paar, "Practical power analysis attacks on software implementations of McEliece," in *Post-Quantum Cryptography*. Springer Berlin Heidelberg, 2010, pp. 108–125.
15. A.-L. Horlemann-Trautmann and V. Weger, "Information set decoding in the Lee metric with applications to cryptography," *Advances in Mathematics of Communications*, vol. online, 2020.
16. N. Lahr, R. Niederhagen, R. Petri, and S. Samardjiska, "Side channel information set decoding using iterative chunking," in *Advances in Cryptology – ASIACRYPT 2020*. Springer International Publishing, 2020, pp. 881–910.
17. ——, "Side channel information set decoding using iterative chunking," in *Advances in Cryptology – ASIACRYPT 2020*, S. Moriai and H. Wang, Eds. Cham: Springer International Publishing, 2020, pp. 881–910.
18. P. Lee and E. Brickell, "An observation on the security of McEliece's public-key cryptosystem," in *Advances in Cryptology - EUROCRYPT 88*. Springer Verlag, 1988, pp. 275–280.

19. A. May, A. Meurer, and E. Thomae, "Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2011, pp. 107–124.
20. R. J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory," *DSN Progress Report*, vol. 44, pp. 114–116, 1978.
21. H. G. Molter, M. Stöttinger, A. Shoufan, and F. Strenzke, "A simple power analysis attack on a McEliece cryptoprocessor," *Journal of Cryptographic Engineering*, vol. 1, no. 1, pp. 29–36, feb 2011.
22. D. Moody and R. Perlner, "Vulnerabilities of "mceliece in the world of escher"," in *Post-Quantum Cryptography*, T. Takagi, Ed. Cham: Springer International Publishing, 2016, pp. 104–117.
23. National Institute of Standards and Technology (NIST), U.S. Department of Commerce, "Post-quantum cryptography standardization," 2017. [Online]. Available: https://csrc.nist.gov/Projects/post-quantum-cryptography/Post-Quantum-Cryptography-Standardization
24. R. Niebuhr, E. Persichetti, P.-L. Cayrel, S. Bulygin, and J. Buchmann, "On lower bounds for information set decoding over fq and on the effect of partial knowledge," *International journal of information and coding theory*, vol. 4, no. 1, pp. 47–78, 2017.
25. T. B. Paiva and R. Terada, "A timing attack on the HQC encryption scheme," in *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 551–573.
26. E. Prange, "The use of information sets in decoding cyclic codes," *IRE Trans. Inf. Theory*, vol. 8, no. 5, pp. 5–9, Sep. 1962.
27. S. Puchinger, J. Renner, and J. Rosenkilde, "Generic Decoding in the Sum-Rank Metric," in *IEEE International Symposium on Information Theory (ISIT), extended version on arxiv: https://arxiv.org/abs/2001.04812*, 2020, pp. 54–59.
28. M. Rossi, M. Hamburg, M. Hutter, and M. E. Marson, "A side-channel assisted cryptanalytic attack against QcBits," in *Lecture Notes in Computer Science*. Springer International Publishing, 2017, pp. 3–23.
29. T. Schamberger, J. Renner, G. Sigl, and A. Wachter-Zeh, "A power side-channel attack on the cca2-secure hqc kem," Cryptology ePrint Archive, Report 2020/910, 2020, https://eprint.iacr.org/2020/910.
30. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.
31. A. Shoufan, F. Strenzke, H. G. Molter, and M. Stöttinger, "A timing attack against patterson algorithm in the mceliece pkc," in *Information, Security and Cryptology – ICISC 2009*, D. Lee and S. Hong, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 161–175.
32. A. Shoufan, F. Strenzke, H. G. Molter, and M. Stöttinger, "A timing attack against patterson algorithm in the McEliece PKC," in *Information, Security and Cryptology – ICISC 2009*. Springer Berlin Heidelberg, 2010, pp. 161–175.
33. B.-Y. Sim, J. Kwon, K. Y. Choi, J. Cho, A. Park, and D.-G. Han, "Novel side-channel attacks on quasi-cyclic code-based cryptography," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. Volume 2019, pp. Issue 4–, 2019.
34. J. Stern, "A method for finding codewords of small weight," in *International Colloquium on Coding Theory and Applications*. Springer, 1988, pp. 106–113.
35. F. Strenzke, "Timing attacks against the syndrome inversion in code-based cryptosystems," in *Post-Quantum Cryptography*. Springer Berlin Heidelberg, 2013, pp. 217–230.

36. F. Strenzke, E. Tews, H. G. Molter, R. Overbeck, and A. Shoufan, "Side channels in the McEliece PKC," in *Post-Quantum Cryptography*. Springer Berlin Heidelberg, 2008, pp. 216–229.
37. I. von Maurich and T. Güneysu, "Towards side-channel resistant implementations of QC-MDPC McEliece encryption on constrained devices," in *Post-Quantum Cryptography*. Springer International Publishing, 2014, pp. 266–282.
38. G. Wafo-Tapa, S. Bettaieb, L. Bidoux, P. Gaborit, and E. Marcatel, "A practicable timing attack against hqc and its countermeasure," Cryptology ePrint Archive, Report 2019/909, 2019, https://eprint.iacr.org/2019/909.

A Further Numerical Results for Known Block Weights

Figures 3 and 4 present numerical results for known block sizes (cf. Section 4.3), for BIKE and HQC, respectively.

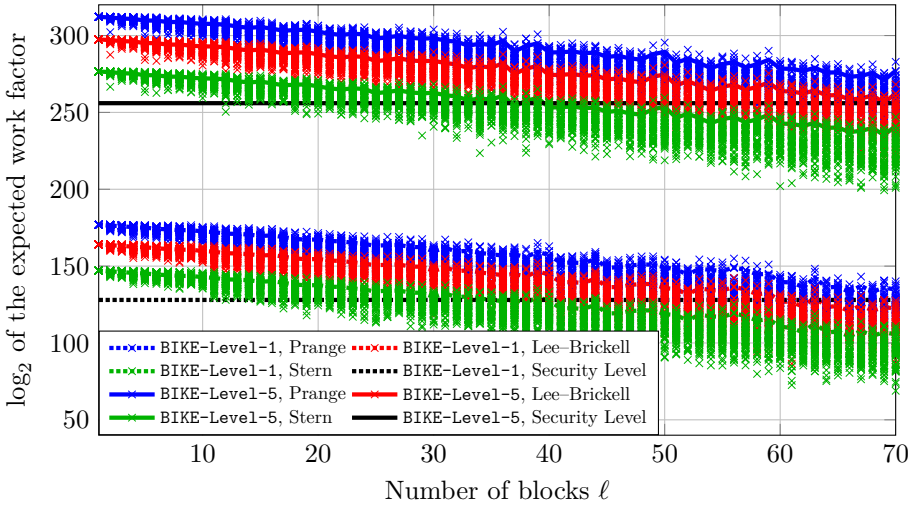

Fig. 3. Work factors of the modified Prange, Lee–Brickell, and Stern algorithms as a function of the number of blocks for which the weight distribution is known, for two BIKE parameter sets. Lines are means, points are realizations.

B Combining Different Types of Hints

The restricted error values from Appendix C can easily be combined with any other hint described previously, since they do not transform the parameters of the SDP, and are only used within the algorithms solving any given SDP. Therefore, we will now focus on how to combine the results from Subsections 3.1 to 3.4.

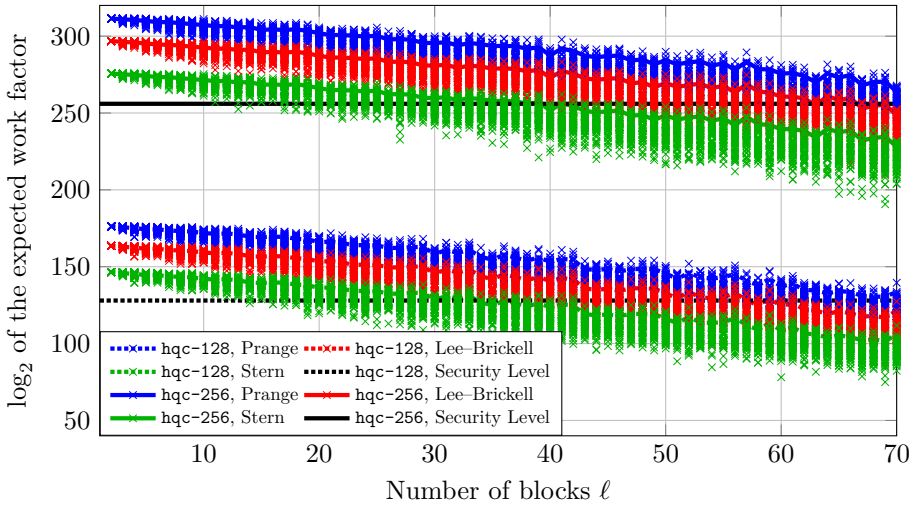

Fig. 4. Work factors of the modified Prange, Lee–Brickell, and Stern algorithms as a function of the number of blocks for which the weight distribution is known, for two HQC parameter sets. Lines are means, points are realizations.

If we know an error measurement (as in 3.2) or a message measurement (as in 3.4), which includes the case of knowing a message entry (as in 3.3), then we transform the original SDP into one where the sought-after codeword lives in a subcode of the original code, with the same error vector, but different received word. Since the error vector is unchanged in this new SDP instance, any hint about it can be used in the usual way. Note however, that when combining message measurements and/or error measurements, it might happen that not all of the hints are linearly independent and that thus some of them do not add any extra information.

If we know a precise error value (as in the first and third case in 3.1), say e_j, then we shorten the code and delete the j-th coordinates in the error vector and the received word (and if $e_j \neq 0$ we decrease the weight of the error vector). The shortening implicitly contains the hint $r_j - e_j = \boldsymbol{m}\boldsymbol{G}(j)$, where $\boldsymbol{G}(j)$ denotes the j-th column of $\boldsymbol{G}$, which is again a measurement of the message. It can hence be combined with other measure measurements (or error measurements) as above, in the shorter code.

If we know only an error location but not the value (as in the second case in 3.1), then we puncture the original code and do not have the implicit measurement which reduces the dimension of the code. That means that $\boldsymbol{m}$ is not changed in the reduction, and hence any hints about the message can be used, also in the reduced $(n-1, k, t-1)$-SDP.

Overall, it follows that all hints can be combined with each other, in arbitrary order. However, it might be that not all of them are linearly independent to prior

hints, and might not provide any extra information, and hence no reduction of the SDP.

C Restricted Error Values

When considering a cryptosystem over $\mathbb{F}_q$ with $q > 2$, and knowing that the error values are from a proper subset $E \subset \mathbb{F}_q$, this can again speed up any decoder. For this the randomly guessed (partial) error vectors in the decoding algorithm are simply chosen from $E^{n'}$ instead of $\mathbb{F}_q^{n'}$ (where n' is the length of the respective partial error vector). This has e.g. been done in [4], where the error vectors live in $\{\pm 1\}^n$.

This idea can straight-forwardly be extended if one only knows that a part of the error vector has restricted error values.

Remark 1. Since Classic McEliece, BIKE and HQC are defined over $\mathbb{F}_2$, this consideration has no impact on their security.

D Partial Measurements of the Error Vector

During encryption of many code-based cryptosystems, one needs to compute the syndrome $\boldsymbol{s}$ of an error $\boldsymbol{e}$ w.r.t. the public parity-check matrix $\boldsymbol{H}$, where all vectors/matrices are in a finite field, typically $\mathbb{F}_2$. This is done using a vector-matrix multiplication $\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^\top$. In practice, this multiplication can be done by grouping entries of the vectors into blocks of a fixed bit size B (e.g., $B = 8$ bits in the reference implementation of Classic McEliece [2]). This means that we in parallel compute, for each row $\boldsymbol{h}_i$ of $\boldsymbol{H}$ and each $j = 0, \ldots, B-1$, the *j-th shifted inner product*

$$b_{i,j} := \sum_{\mu=0}^{n/B-1} e_{j+\mu B} H_{i,j+\mu B}.$$

To get the inner product of $\boldsymbol{e}$ with the i-th row of $\boldsymbol{H}$, we finally need to sum up

$$\langle \boldsymbol{e}, \boldsymbol{h}_i \rangle = \sum_{j=0}^{B-1} b_{i,j}.$$

If we are able to retrieve, for some i, the B-bit vector $\boldsymbol{b}_i = [b_{i,0}, \ldots, b_{i,B-1}]$, then we obtain B measurements of the error, where the measurement vectors are of the form

$$\begin{aligned}
\boldsymbol{v}_{i,0} &:= [H_{i,0}, \underbrace{0, \ldots, 0}_{B-1 \text{ zeros}}, H_{i,B}, 0, \ldots, 0, H_{i,2B}, 0, \ldots, 0, H_{i,n-B}] \\
\boldsymbol{v}_{i,1} &:= [0, H_{i,1}, 0, \ldots, 0, H_{i,B+1}, 0, \ldots, 0, H_{i,2B+1}, 0, \ldots, 0] \\
&\vdots \\
\boldsymbol{v}_{i,B-1} &:= [0, \ldots, H_{i,B-1}, 0, \ldots, 0, H_{i,2B-1}, 0, \ldots, 0, H_{i,n-1}]
\end{aligned}$$

The vector $\boldsymbol{b}_i$, for some i, can be obtained through a template attack on the syndrome computation in the encryption step. Please note that this attack on the reference implementation of Classic McEliece allows for the usage of value templates since B is small.

E Proofs

In this appendix, we present proofs of some of the theorems in the paper.

Proof of Theorem 7 (Modified Prange Algorithm). The cost of one iteration stays the same as in the original Prange decoder and is dominated by matrix inversion. An iteration succeeds iff the chosen positions in $\mathcal{X}$ are error-free.[3] This is again true iff every $\mathcal{X}_i$ is error-free. The fraction $\binom{\eta_i - x_i}{t_i} / \binom{\eta_i}{t_i}$ equals the probability that in block $\mathcal{W}_i$ of length η_i, the randomly chosen x_i positions are error-free, given that exactly t_i errors are contained in $\mathcal{W}_i$, i.e., exactly Prange's success probability restricted to one block. Since the positions $\mathcal{X}_i$ are chosen independently, the claim follows. □

Proof of Theorem 8 (Modified Stern Algorithm). The proof works similar to the adapted Lee–Brickell algorithm in Appendix F. One iteration costs as much as in the original Stern algorithm, see [15] for a detailed analysis. Stern's stopping condition is fulfilled if and only if there are exactly p errors in $\mathcal{X}$, p errors in $\mathcal{Y}$, and 0 errors in $\mathcal{Z}$. The success probability formula then follows by summing over all cases to distribute p errors over the partition $\mathcal{X}_i$ of $\mathcal{X}$ (vector $\boldsymbol{a}$) and to distribute p errors over the partition $\mathcal{Y}_i$ of $\mathcal{Y}$ (vector $\boldsymbol{b}$). Again, the number of summands is exponential in n, k, ℓ, p, but we can compute P_{Stern} in polynomial time using dynamic programming. □

F Modified Lee–Brickell Algorithm for Known Block Weights Hint

Lee and Brickell's algorithm works similarly to Prange's ISD algorithm. The difference is that the algorithm succeeds even if a few errors are contained in the randomly chosen information set. This means that the success probability is significantly increased compared to Prange's algorithm, but a bit more work is needed per iteration. We adapt the choice of the information sets in the Lee–Brickell algorithm: As in our adaptation of Prange's ISD, we choose from each block a given number positions for the information set, and hope that overall at most a few errors are contained in the information set. The number of positions that we choose from each block depends on the distribution of errors. Algorithm 3 summarizes the decoder for blocks.

Theorem 9 states the expected work factor of the adapted Lee–Brickell algorithm. Note that the formula for the success probability in (2) consists of a sum

[3] As in most works on information-set decoding, we neglect the probability that a randomly chosen set is not an information set, since it is for most codes a constant in the same order of magnitude as 1.

Algorithm 3: Lee–Brickell with Known Block Weights

Input: ISD problem and weight decomposition, $\boldsymbol{t}$ of the error, vector $\boldsymbol{x}$, parameter p
Output: Error $\boldsymbol{e}$
do
  $\mathcal{X} \leftarrow \cup_{i=1}^{\ell} \mathcal{X}_i$, where the $\mathcal{X}_i$ are chosen independently uniformly from the subsets of $\mathcal{W}_i$ of cardinality x_i
  $\boldsymbol{e} \leftarrow$ Iteration of original Lee–Brickell alg. w.r.t. inf. set $\mathcal{X}$ and parameter p
while *Lee–Brickell stopping condition not satisfied for* $\boldsymbol{e}$
return $\boldsymbol{e}$

whose number of summands may grow exponentially in the parameters p and ℓ. However, we can compute it in polynomial time using dynamic programming in a similar approach as in [27].

Theorem 9. *For a given $\boldsymbol{x}$ and parameter p, Algorithm 3 has an expected work factor of $W_{\mathsf{LB}} = \frac{W_{\mathsf{LB,Iter}}}{P_{\mathsf{LB}}}$, where the cost per iteration is given by $W_{\mathsf{LB,Iter}} := (n-k)^2(n+1) + (n-k)\sum_{i=1}^{p}\binom{k}{i} + (n-k)\binom{k}{p}$, and the success probability of each iteration is*

$$P_{\mathsf{LB}} = \sum_{\substack{\boldsymbol{a}\in\mathbb{Z}^{\ell} \\ 0\le a_i\le t_i \\ \sum_i a_i = p}} \prod_{i=1}^{\ell} \frac{\binom{x_i}{a_i}\binom{\eta_i - x_i}{t_i - a_i}}{\binom{\eta_i}{t_i}}. \tag{2}$$

The success probability P_{LB} can be computed in polynomial bit complexity (in the parameters n, k, ℓ, p).

Proof. We adapt the original Lee–Brickell algorithm only by changing the selection of the information set. Hence, the cost per iteration $W_{\mathsf{LB,Iter}}$ is the same as in the original algorithm. The cost is as stated if the concept of intermediate sums is used (cf. [7]).

The Lee–Brickell algorithm succeeds if the information set $\mathcal{X}$ is chosen such that it contains exactly p errors, and the remaining positions contain exactly $t - p$ errors. Hence, for P_{LB}, we sum over all possibilities that the p errors are distributed over the information set partitions $\mathcal{X}_i$ in the ℓ blocks. Obviously, there can only be at most t_i errors in the i-th information set. The products of fractions of binomial coefficients count the number of possibilities to choose information sets with $|\mathcal{X}_i| = x_i$ that contain exactly a_i errors, divided by the number of possibilities to distribute the t_i error positions on the η_i positions of a block.

We can compute P_{LB} in polynomial time using dynamic programming, see [27] for an efficient algorithm to compute a similar formula. □

Again, the success probability depends significantly on the choice of $\boldsymbol{x}$ for given $\boldsymbol{\eta}$ and $\boldsymbol{t}$. A possible heuristic is to the same as for Prange's decoder: Choose $x_i = \eta_i - t_i$ and decrease that x_i by one, for which (2) is increased the

most. Since the best choice for p is often a small integer, it appears to be a good enough choice to use exactly the same $\boldsymbol{x}$ as computed for Prange's ISD ($p = 0$), even though p is chosen to be greater than 0.