A Security Analysis of Restricted Syndrome Decoding Problems

Ward Beullens[1], Pierre Briaud[2] and Morten Øygarden[2]

[1] IBM Research Europe, Zürich, Switzerland
[2] Simula UiB, Bergen, Norway

Abstract. Restricted syndrome decoding problems (R-SDP and R-SDP(G)) provide an interesting basis for post-quantum cryptography. Indeed, they feature in CROSS, a submission in the ongoing process for standardizing post-quantum signatures. This work improves our understanding of the security of both problems. Firstly, we propose and implement a novel collision attack on R-SDP(G) that provides the best attack under realistic restrictions on memory. Secondly, we derive precise complexity estimates for algebraic attacks on R-SDP that are shown to be accurate by our experiments. We note that neither of these improvements threatens the updated parameters of CROSS.

Keywords: Code-based Cryptography · Restricted Errors · Post-Quantum Cryptography · Cryptanalysis

1 Introduction

It is well-known that large-scale quantum computers will be able to break most of the public-key cryptography in use today. The move to new post-quantum standards for signature and public-key encapsulation mechanisms (KEMs) is well underway. Indeed, the (U.S.) National Institute for Standards and Technology (NIST) has recently concluded a multi-year standardization process for post-quantum algorithms, based on feedback from international academia, industry, and governmental organizations, and the documentation for new standards is being finalized at the time of writing [AAC+22]. The majority of the selected algorithms are based on the computational hardness of problems related to structured lattices, and NIST is currently looking to diversify its portfolio by standardizing schemes based on different hardness assumptions. For KEMs, there are still several candidates from the aforementioned standardization process that are being evaluated, however, there were no remaining viable signature candidates. This prompted NIST to issue a call for additional post-quantum signature schemes, resulting in 40 proposed algorithms that were published in July 2023 for further scrutiny.

One of the main directions in post-quantum cryptography is to base the computational hardness on problems from coding theory. These code-based algorithms have traditionally relied on the Syndrome Decoding Problem (SDP) which is defined as follows.

Problem 1 ((Computational) SDP). *For a given full-rank matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a vector $\mathbf{s} \in \mathbb{F}_q^{n-k}$ and an integer $t \leq n$, find $\mathbf{e} \in \mathbb{F}_q^n$ of weight t satisfying $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$.*

State-of-the-art ISD algorithms for solving this problem usually involve searching for a number of zero entries in $\mathbf{e}$. Thus their computational cost generally worsens as the

E-mail: ward@beullens.com (Ward Beullens), pierre@simula.no (Pierre Briaud), morten.oygarden@simula.no (Morten Øygarden)

weight is increased. This motivated the works of [BBC+20, BBP+23] to look into ways of relaxing the weight restriction by instead limiting the error vector $\mathbf{e}$ to a subset of $\mathbb{F}_q^n$. The culmination of these works is CROSS [BBB+23], a family of signature schemes that was submitted to the ongoing call for additional post-quantum signature standards. The signature schemes are derived from an interactive zero-knowledge identification protocol using Fiat-Shamir transforms. The underlying hard problem in these protocols is either the *restricted syndrome decoding problem* (R-SDP) or a further specialization known as R-SDP(G). The idea of R-SDP is to limit the entries of $\mathbf{e}$ to a multiplicative subgroup $E \subset \mathbb{F}_q^*$ of order $z < q-1$. The restricted syndrome decoding problem with respect to the group E is then defined as

Problem 2 (R-SDP)**.** *Given $g \in \mathbb{F}_q^*$ of order z, $E := \{g^j,\ j \in \{0..z-1\}\}$, a full-rank matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ and a vector $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, find $\boldsymbol{e} \in E^n$ such that $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$.*

The further specialization of R-SDP(G) is achieved by considering errors from a subgroup G of E^n. The set E^n is endowed with the $\star$ operation, which performs the entry-wise multiplication of two vectors. We can then use elements $\mathbf{a}_1, \ldots, \mathbf{a}_m \in E^n$ to generate a subgroup $(G, \star) \subset (E^n, \star)$ as

$$G := \langle \mathbf{a}_1, \ldots, \mathbf{a}_m \rangle := \{\mathbf{a}_1^{u_1} \star \ldots \star \mathbf{a}_m^{u_m} \mid u_i \in \{0, \ldots, z-1\}\}\,.$$

The syndrome decoding problem restricted to G is now defined as follows.

Problem 3 (R-SDP(G))**.** *Given a subgroup $(G, \star) \subset (E^n, \star)$ of order z^m, a full-rank matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ and a vector $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, find $\boldsymbol{e} \in G$ such that $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$.*

The authors of [BBB+23] point to several advantages of using the restricted variants of SDP. Both R-SDP and R-SPD(G) are NP-hard, and generic decoders seem to have a larger computational complexity when compared to a similar instance of SDP. That said, the restricted variants are fairly recent problems, whose concrete security is not as well-studied as that of traditional syndrome decoding. One particular problem is how an attacker may use the structure of G to speed up variants of Stern-Dumer collision attacks on R-SDP(G). Another natural question is whether the restrictions on the error can open up for efficient algebraic attacks, which has recently been shown to be the case for a different decoding problem that relies on structured errors [BØ23].

Contributions. This work explores two directions in the security of restricted syndrome decoding problems. First, we present a new collision attack on R-SDP(G) that is designed with the special structure of G in mind. Note that the initial submission of CROSS (version 1.0) was not able to use G in their security analysis. This was, however, changed in an updated version 1.1, which was made public while we were working on this paper[1]. The new version proposes a collision attack that utilizes G and suggests updated parameters for R-SDP(G) to account for this. We will see that the collision attack presented in this paper is different from that of [BBB+23] and allows for different trade-offs in terms of time and memory complexity. Also, there is no need for assumptions on public key for the attack to work.

The second half of this paper is devoted to algebraic attacks on R-SDP. Note that [BBB+23] provided a heuristic argument that Gröbner basis algorithms will not outperform other attacks, and provided experiments to support this claim. Our analysis goes beyond this by giving estimates for the Hilbert series of the equations of [BBB+23]. These estimates rely on standard arguments and have been shown to be exact in our experiments. The concrete bounds we can obtain from them suggest that the Gröbner basis cost conjectured in [BBB+23] might be slightly overestimated. Still, we agree with their overall conclusion that algebraic attacks will not threaten the CROSS R-SDP parameters.

[1] As of April 2024, the latest release of the CROSS specification is version 1.2. However, the contents of this latter update has no impact on the topics in this paper.

2 A New Combinatorial Attack on R-SDP(G)

In this section we introduce a new combinatorial attack on the R-SDP(G) problem, we report on a proof-of-concept implementation and we compare against the state of the art. We conclude that while our algorithm has a time complexity similar to that of state-of-the-art algorithms, it has much lower memory requirements and works for all keys, as opposed to some subset of weak keys.

Exploiting group elements with disjoint support. Recall that the R-SDP(G) problem is equivalent to finding a vector $\boldsymbol{e}$ in the intersection of the affine subspace defined by $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ and of the multiplicative group G. This appears to be a difficult problem because the additive structure of the affine subspace does not interact nicely with the multiplicative structure of G. However, observe that if $\boldsymbol{l}, \boldsymbol{r} \in G$ are two elements of G with multiplicative support in the first and second half respectively, *i.e.*, $\boldsymbol{l} := (\boldsymbol{u}, \boldsymbol{1}) \in G$ and $\boldsymbol{r} := (\boldsymbol{1}, \boldsymbol{v}) \in G$ for $\boldsymbol{u} \in E^{\lceil n/2 \rceil}$ and $\boldsymbol{v} \in E^{\lfloor n/2 \rfloor}$, then multiplication and addition does interact nicely. More precisely, we have

$$(\boldsymbol{l} \star \boldsymbol{r})\boldsymbol{H}^\top = (\boldsymbol{u}, \boldsymbol{v})\boldsymbol{H}^\top = \boldsymbol{u}\boldsymbol{H}_L^\top + \boldsymbol{v}\boldsymbol{H}_R^\top, \tag{1}$$

where $\boldsymbol{H} = (\boldsymbol{H}_L\, \boldsymbol{H}_R)$, $\boldsymbol{H}_L \in \mathbb{F}_q^{(n-k)\times\lceil n/2 \rceil}$, $\boldsymbol{H}_R \in \mathbb{F}_q^{(n-k)\times\lfloor n/2 \rfloor}$. We exploit this property to do a collision attack. Let $L, R \subset G$ be the subgroups of G with support in the left and right half respectively. Then we try to find a collision

$$\boldsymbol{u}\boldsymbol{H}_L^\top = \boldsymbol{s} - \boldsymbol{v}\boldsymbol{H}_R^\top,$$

for $(\boldsymbol{u}, \boldsymbol{v})$ such that $\boldsymbol{l} = (\boldsymbol{u}, \boldsymbol{1}) \in L$ and $\boldsymbol{r} = (\boldsymbol{u}, \boldsymbol{1}) \in R$. If we find such a collision, then $\boldsymbol{e} := \boldsymbol{l} \star \boldsymbol{r}$ is a solution to the R-SDP(G) problem. The attack only works if there exists a solution in the subgroup $L \star R := \{\boldsymbol{l} \star \boldsymbol{r} \mid \boldsymbol{l} \in L \text{ and } \boldsymbol{r} \in R\}$, which in general only happens with a small probability. Therefore, we modify the attack to search for a solution in any coset $L \star R \star \boldsymbol{f}$ and we run the attack for all $\boldsymbol{f}$ in $G/(L \star R)$ until a solution is found.

Stern-Dumer-like optimization. To reduce the cost of the attack, we use a standard idea inspired by the Stern-Dumer decoder [Dum91, Ste89]. More precisely, we put the matrix $(\boldsymbol{H}\boldsymbol{s}^\top)$ in row-reduced echelon form and discard the ℓ top rows. What remains is a new matrix $\boldsymbol{H}' \in \mathbb{F}_q^{(n-k-\ell)\times n}$ and a new syndrome $\boldsymbol{s}' \in \mathbb{F}_q^{n-k-\ell}$. The idea is to sample solutions $\boldsymbol{e}'$ to the smaller system $\boldsymbol{e}'\boldsymbol{H}'^\top = \boldsymbol{s}'$ until we find a solution that also satisfies the original system $\boldsymbol{e}'\boldsymbol{H}^\top = \boldsymbol{s}$. For appropriately chosen values of ℓ this is more efficient than the direct approach because the cost of checking false positives is much smaller than the savings we get from attacking the smaller instance.

2.1 Description and Analysis of the Attack

A complete description of our attack can be found in Algorithm 1.

Algorithm 1: Algorithm for the R-SDP(G) problem

Input : Parity-check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$, syndrome $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$, and subgroup $G \subset E^n$ of rank m, parameter ℓ such that $n-\ell$ is even.

Output : Solution $\boldsymbol{e} \in G$ such that $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$, if it exists. Otherwise output $\perp$.

1 Using Gaussian Elimination mod z, compute the subgroups of G:

$$L := \{\boldsymbol{l} \in G,\ l_i = 1 \text{ for all } i \text{ such that } \ell + (n-\ell)/2 < i \leq n\ \},$$
$$R := \{\boldsymbol{r} \in G,\ r_i = 1 \text{ for all } i \text{ such that } \ell < i \leq \ell + (n-\ell)/2\ \}.$$

2 Using elementary row operations, put the matrix $(\boldsymbol{H}\ \boldsymbol{s}^\top)$ in the form

$$\begin{pmatrix} \boldsymbol{I}_\ell & * & * & * \\ \boldsymbol{0} & \boldsymbol{H}_L & \boldsymbol{H}_R & (\boldsymbol{s}')^\top \end{pmatrix},$$

where $\boldsymbol{I}_\ell$ is the identity matrix of size ℓ, $\boldsymbol{H}_L, \boldsymbol{H}_R \in \mathbb{F}_q^{(n-k-\ell)\times(n-\ell)/2}$, and $\boldsymbol{s}' \in \mathbb{F}_q^{n-k-\ell}$.

3 **for** $\boldsymbol{f} \in G/(L \star R)$ **do**

`/* Search for a solution in the coset` $L \star R \star \boldsymbol{f}$ `*/`

4 Run a collision search to enumerate all pairs $(\boldsymbol{l}, \boldsymbol{r}) \in L \times R$ such that

$$(\boldsymbol{f} \star \boldsymbol{l}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{H}_L^\top \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{s}' - (\boldsymbol{f} \star \boldsymbol{r}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{H}_R^\top \end{pmatrix}.$$

5 For every collision $(\boldsymbol{l}, \boldsymbol{r})$ check if $(\boldsymbol{f} \star \boldsymbol{l} \star \boldsymbol{r})\boldsymbol{H}^\top = \boldsymbol{s}$. If this is the case, output the solution $\boldsymbol{e} := \boldsymbol{f} \star \boldsymbol{l} \star \boldsymbol{r}$.

6 Output $\perp$.

Correctness. If there exists a solution $\boldsymbol{e} \in G$ such that $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$, then the algorithm is guaranteed to output a solution. Indeed, the for-loop on line 3 iterates over all elements of $G/(L \star R)$, which means that the algorithm will eventually reach an iteration with an $\boldsymbol{f}$-vector such that there exists a solution of the form $\boldsymbol{e} := \boldsymbol{l} \star \boldsymbol{r} \star \boldsymbol{f}$, $(\boldsymbol{l}, \boldsymbol{r}) \in L \times R$. Since $\boldsymbol{e}$ is a solution we have

$$(\boldsymbol{l} \star \boldsymbol{r} \star \boldsymbol{f}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{H}_L^\top \\ \boldsymbol{H}_R^\top \end{pmatrix} = \boldsymbol{s}',$$

which can be rewritten as

$$(\boldsymbol{l} \star \boldsymbol{f}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{H}_L^\top \\ \boldsymbol{0} \end{pmatrix} = \boldsymbol{s}' - (\boldsymbol{r} \star \boldsymbol{f}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{H}_R^\top \end{pmatrix}. \tag{2}$$

This means that $(\boldsymbol{l}, \boldsymbol{r})$ is one of the collisions found by the collision search on line 4. The solution $\boldsymbol{e}$ will pass the check of line 5 and be output by the algorithm.

Cost analysis. We now analyze the expected cost of our attack. We assume that the ℓ parameter is chosen such that $n \leq 2m + \ell \leq 2n$. Then, generically, L and R will have rank $r = m + (\ell - n)/2$ and $L \star R$ will have rank $2r$. We ignore the cost of computing these subgroups and doing the partial Gaussian elimination on $(\boldsymbol{H}\ \boldsymbol{s}^\top)$ because these operations can be done in polynomial time. The corresponding cost will in particular be negligible compared to the complexity of the rest of the attack for cryptographically interesting parameters.

The main cost of the algorithm is due to the loop on line 3. Heuristically, the solutions of the R-SDP(G) problem will be distributed uniformly over the cosets $G/(L \star R)$. If there are S such solutions, this means that on average we need to consider $|G/(L \star R)|/(S+1) = z^{n-\ell-m}/(S+1)$ cosets before finding one. The cost of each iteration is then the sum of the cost of collision search on line 4 and of the cost of checking for false positives on line 5.

Cost of false positives. On average, there are approximately $|L \star R| q^{k+\ell-n}$ false positives in each iteration, and dispelling a false positive $\boldsymbol{e}$ requires checking on average $q/(q-1) \approx 1$ entries of $\boldsymbol{e}\boldsymbol{H}^\intercal = \boldsymbol{s}$, which can be done with roughly $n-\ell$ field multiplications. So the total cost of checking false positives in one iteration is approximately $(n-\ell) z^{2r} q^{k+\ell-n}$ field multiplications.

Cost of the collision search. The cost of collision search depends on the strategy that is used to find the collisions. The naive method would be to build the list

$$\boldsymbol{\Gamma}_{\boldsymbol{f}} := \left\{ (\boldsymbol{l}, (\boldsymbol{l} \star \boldsymbol{f}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{H}_L^\intercal \\ \boldsymbol{0} \end{pmatrix}) \text{ for } \boldsymbol{l} \in L \right\},$$

and then for every $\boldsymbol{r} \in R$ compute the right hand side of Equation (2) and check if it occurs in $\boldsymbol{\Gamma}_{\boldsymbol{f}}$. An advantage of this approach is that the cost can be amortized over many $\boldsymbol{f}$. Indeed, after one $\boldsymbol{\Gamma}_{\boldsymbol{f}}$ is built, it will be cheaper to compute $\boldsymbol{\Gamma}_{\boldsymbol{f}'}$ for $\boldsymbol{f}' := \boldsymbol{f} \star \boldsymbol{f}''$ where $\boldsymbol{f}''$ is a low-weight codeword of G. Similarly, updating the set of right-hand sides is more efficient than recomputing it from scratch. However, because of the large memory requirement and the cost of accessing the huge list, we expect the naïve approach to be much more expensive than more memory-friendly alternatives in "realistic" cost models. Therefore, we analyze the cost of our attack using the van Oorschot-Wiener collision search [VOW99] to enumerate the collisions.

Given two functions $f_1 : X_1 \to Y$, $f_2 : X_2 \to Y$, with $N = |X_1| = |X_2| < |Y|/2$ and enough memory to store M elements of $X_1 \sqcup X_2$, the van Oorschot-Wiener algorithm allows to enumerate a large fraction of all collisions $(x_1, x_2) \in X_1 \times X_2$ such that $f_1(x_1) = f_2(x_2)$ with a runtime cost dominated by $3.5N^{1.5}/M^{0.5}$ evaluations of f_1 and f_2. (Note: if $M < N$, then this compares favorably a the naïve collision search which takes N^2/M function evaluations. Moreover, van Oorschot-Wiener parallellizes much better than the naïve approach.) Applying the van Oorschot-Wiener algorithm to our application, we have

$$N = |L| = |R| = z^r = z^{m+(\ell-n)/2},$$

$$f_1 : L \to \mathbb{F}_q^{n-k-\ell} : \boldsymbol{l} \mapsto (\boldsymbol{l} \star \boldsymbol{f}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{H}_L^\intercal \\ \boldsymbol{0} \end{pmatrix}, \text{ and}$$

$$f_2 : R \to \mathbb{F}_q^{n-k-\ell} : \boldsymbol{r} \mapsto \boldsymbol{s}' - (\boldsymbol{r} \star \boldsymbol{f}) \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{H}_R^\intercal \end{pmatrix}.$$

An evaluation of f_1 or f_2 requires roughly $(n-\ell)/2 \times (n-k-\ell)$ field multiplications, so the overall cost of the collision search can be estimated as the cost of $3.5(n-\ell)(n-k-\ell) z^{1.5m+0.75(\ell-n)}/M^{0.5}$ field multiplications.

Table 1: Performance of our attack against the SL1 parameter set of CROSS, as a function of the amount of memory that is used.

Memory	fraction of distinguished points	F evals per second	partial solutions per second
128 KB	2^{-10}	34 M	2.0 K
512 KB	2^{-9}	34 M	4.1 K
2 MB	2^{-8}	33 M	8.1 K
8 MB	2^{-7}	33 M	16 K
32 MB	2^{-6}	32 M	31 K
128 MB	2^{-6}	31 M	56 K
512 MB	2^{-5}	29 M	104 K
2 GB	2^{-4}	24 M	167 K
8 GB	2^{-3}	19 M	246 K

Total cost. Putting everything together, we estimate the total cost of the attack in field multiplications by

$$\underbrace{\frac{z^{n-\ell-m}}{S+1}}_{\text{iterations}} \left(\underbrace{3.5(n-\ell)(n-k-\ell)\frac{z^{1.5m+0.75(\ell-n)}}{M^{0.5}}}_{\text{cost of collision search}} + \underbrace{(n-\ell)\frac{z^{2r}}{q^{n-k-\ell}}}_{\text{cost of false positives}} \right). \tag{3}$$

2.2 Proof-of-Concept Implementation

We implemented our attack for the SL1 parameters of CROSS. Our analysis suggests that $\ell = 15$ gives the best attack performance, resulting in groups L and R of rank 5, and $\boldsymbol{H}_L$ and $\boldsymbol{H}_R$ having 4 rows. Therefore, the naive collision search would require storing z^5 elements of $\log(p^4)$ bits each, which amounts to roughly 138 GB of memory. While this is not a prohibitively large amount of memory, we expect that the cost of frequently accessing such a large amount of memory makes the naïve collision search less efficient than the van Oorschot-Wiener method, so we used the latter in our implementation of the attack. The vOW method repeatedly evaluates the function $F : \mathbb{F}_p^5 \to \mathbb{F}_p^5 : \boldsymbol{v} \mapsto f_{H_1(\boldsymbol{v})(H_{2,H_1(\boldsymbol{v})}(\boldsymbol{v})}$, where $H_1 : \mathbb{F}_p^5 \to \{1,2\}$, $H_{2,1} : \mathbb{F}_p^5 \to L$, and $H_{2,2} : \mathbb{F}_p^5 \to R$ are hash functions. Using a single core of an Intel i9-1088H CPU, our preliminary implementation can do up to roughly 34 million evaluations of F per second. Table 1 reports the amount of partial solutions that our implementation finds per evaluation of F (including the cost of checking if the partial solution is a solution to the full R-SDP(G) problem), as a function of how much memory the attack is allowed to use. The table shows that, as the amount of memory increases, fewer evaluations of F are required per partial solution. However, when the amount of memory is large, the number of F-evaluations per second decreases, because the attack is starting to get bottlenecked by the memory accesses, which get more expensive and more frequent (because of the higher fraction of distinguished points). The fraction of distinguished points is chosen to be the power of $1/2$ that maximizes the number of partial solutions checked per second (last column). Since an expected number of $q^l \approx 2^{134}$ partial solutions needs to be checked before a real solution is found, we do not claim that our attack breaks the security level of the new parameter set. E.g. using 8 GB of memory, the attack should take $2^{134}/246000$ core-seconds, which is much longer than a key search against AES-128 would take on the same hardware.

2.3 Comparison with Previous Analysis

This section aims to discuss and compare our new collision attack with the recently introduced collision attack from [BBB+23, Section 7.1.2]. For completeness, we provide a brief overview of this latter attack in Appendix A.1. Our attack shares some similarities with the attack of [BBB+23], both algorithms use the Stern-Dumer-like approach in combination with a collision search. The main difference is that our algorithm performs collision searches between lots of functions with “small” domains, as opposed to the algorithm of [BBB+23] which does a single collision search between two functions with much larger domains. If the collision searches are performed naively, then both attacks have a similar time cost, but our attack has a much lower memory cost. In a realistic scenario where an attacker has a limited amount of memory it is necessary to perform the collision search with a time-memory trade-off such as with the Van Oorschot-Wiener algorithm. In this case, for a fixed amount of memory, our attack will be more time-efficient. A second difference is that our algorithm works for all instances of the RSDP-(G) problem (i.e. all public keys of CROSS), whereas the algorithm of [BBB+23] only works for a small subset of weak public keys. This is because their algorithm searches for and exploits two large subcodes $\mathcal{C}_1, \mathcal{C}_2$ of the dual of $\log(G)$ which have disjoint support, and these codes do not exist for most G. We remark that it is possible to run the attack of [BBB+23] with $\mathcal{C}_1 = \mathcal{C}_2 = \{0\}$, which would result in an attack that works for all public keys, at the cost of only a relatively small loss of efficiency.

2.3.1 Comparing Complexity Estimates.

In Table 2 we showcase three settings of Algorithm 1, under the various assumptions described in Section 2.1. While the settings are limited in memory, we note that this still yields purely theoretical estimates that do not take into account the potential costs of using this (still fairly) large amount of memory, as discussed in Section 2.2. Time complexities are counted in $\log_2$ multiplications in $\mathbb{F}_q$ using (3), and we have used $S = z^m q^{k-n}$ as the expected number of solutions for the R-SDP(G) problem.

Table 2 includes both the old and new parameter sets of CROSS, respectively from versions 1.0 and 1.1 of the CROSS specification. The different parameter sets are denoted SL(n, k, m), where $q = 509$ and $z = 127$ are used in all sets. Recall that the security levels 1, 3 and 5 are based on the security of AES-λ for $\lambda = 128$, 192 and 256, respectively. The three settings of Algorithm 1 differ by restricting $\log_2 M$ to either $\lambda/8$, $\lambda/4$ or $\lambda/2$. Note that these memory bounds sometimes exceed the chosen list size N. To account for this, we have replace the denominator in the collision search cost of Equation (3) with $\mathrm{Max}(N^{0.5}, M^{0.5})$. For the chosen parameters we do not observe any improvements by further increasing the allowed memory.

Table 2: Complexity estimation of Algorithm 1 for solving R-SDP(G) under three different memory restrictions. Time is counted as $\log_2$ multiplications in $\mathbb{F}_q$.

	Algorithm 1 $\log_2 M = \lambda/8$		Algorithm 1 $\log_2 M = \lambda/4$		Algorithm 1 $\log_2 M = \lambda/2$	
	ℓ	Time	ℓ	Time	ℓ	Time
SL1(55, 36, 25)	15	154	15	146	15	145
SL3(79, 48, 40)	25	231	23	223	23	212
SL5(106, 69, 48)	30	292	30	277	30	276
Old SL1(42, 23, 24)	14	134	14	128	12	114
Old SL3(63, 35, 36)	21	196	19	188	17	171
Old SL5(87, 47, 48)	31	261	29	248	27	225

Discussion. For comparison, we briefly recall the time-optimized complexities reported in [BBB+23, Section 8.2]. Note that these timings are given in *bit operations*, under various conservative assumptions (see [BBB+23, Theorem 15] for further details). While memory does affect these time estimates to some extent, there are otherwise no restrictions on the amount of memory used in the attack. For these reasons, the following numbers are meant to provide a qualitative, as opposed to a direct, comparison with Table 2.

We write $\mathfrak{t}$ and $\mathfrak{m}$ for the lower bound on time (in bit operations) and memory (in bits) reported in [BBB+23, Section 8.2]. Furthermore, we let $\mathfrak{f}$ denote a conservative upper bound on the fraction of the public keys affected by this specific attack, computed as $P(d_1, j_1)P(d_2, j_2)$ in accordance with [BBB+23, Section 7.1.2]. For SL1 these numbers are $\mathfrak{t} = 2^{143}$, $\mathfrak{m} = 2^{132}$ and $\mathfrak{f} = 2^{-118}$. For SL3 we have $\mathfrak{t} = 2^{210}$, $\mathfrak{m} = 2^{196}$ and $\mathfrak{f} = 2^{-116}$. Finally, for SL5 the algorithm requires $\mathfrak{t} = 2^{272}$ and $\mathfrak{m} = 2^{259}$ with $\mathfrak{f} = 2^{-209}$. In Table 2 we see that the respective time complexities increase with Algorithm 1 as we add restrictions on memory and ease assumptions on the public key (this is especially true as the table is reported in operations in $\mathbb{F}_q$). Thus the updated SL parameters seem to have a notable security buffer against these collision attacks.

The "Old SL" parameters were, on the other hand, shown to be vulnerable against the collision attack of [BBB+23] (version 1.1), though no attack parameters were specifically suggested. Table 2 shows that these parameters are also vulnerable to Algorithm 1 under certain memory restrictions (and without assumptions on the public key).

3 Gröbner Basis Approach on R-SDP

This section is devoted to the study of algebraic attacks on R-SDP using Gröbner bases. We refer to [CLO13] for the fundamental definitions and theory on Gröbner bases, as well as their role in solving systems of multivariate polynomials. In the following we briefly recall the algebraic analysis from [BBB+23], which proposed the following way to model R-SDP as a system of polynomial equations.

System 1 ([BBB+23], §7.2 p. 43)**.** *Let $R = \mathbb{F}_q[e_1, \ldots, e_n]$, where e_i is the i-th coordinate of the error vector $\boldsymbol{e}$ for $i \in \{1..n\}$. Let $\mathcal{L}$ be the linear system of size $n - k$ corresponding to the parity-check equations $\boldsymbol{e}\boldsymbol{H}^\top = \boldsymbol{s}$ and let*

$$\mathcal{Z} := \{\forall i \in \{1..n\},\ e_i^z - 1\}\,.$$

Finally, let $\mathcal{F} := \mathcal{L} \cup \mathcal{Z}$.

It is easy to see that the solutions to $\mathcal{F}$ exactly correspond to the solutions of the R-SDP problem. The analysis of [BBB+23] assumes that that the complexity of finding a solution to $\mathcal{F}$ is dominated by finding a Gröbner basis for the ideal $\langle\mathcal{F}\rangle$. The authors then estimate the complexity of solving this problem with the Gröbner basis algorithm F_5 [Fau02] terminating in degree d_{reg} by

$$\mathcal{O}\left(\binom{n+d_{\mathrm{reg}}}{n}^{\omega}\right), \tag{4}$$

where $2 \leq \omega \leq 3$ is the linear algebra constant. Predicting the degree d_{reg} is, in general, difficult, but [BBB+23] argues heuristically that its growth will be linear in n. This is also backed by experiments performed with the computer algebra system Magma [BCP97], reported in [BBB+23, Table 2 p. 45]. In turn, the authors argue that such an algebraic approach could not be competitive with combinatorial methods for the proposed parameters in CROSS.

In this section we ultimately reach the same conclusion, but provide a tighter analysis. We also consider hybrid approaches, which seem to give better results in practice. We start with recalling some algebraic preliminaries, which will be needed for our analysis.

3.1 Preliminaries

Let R denote a polynomial ring over $\mathbb{F}_q$ in n variables. For a set of polynomials $p_1, \ldots, p_m \in R$, we let $I = \langle p_1, \ldots, p_m \rangle \subset R$ denote its ideal. I is said to be a homogeneous ideal if it can be generated by a set of homogeneous polynomials. Let R_d denote the $\mathbb{F}_q$-vector space generated by the monomials of degree d in R. For a homogeneous ideal I, we have the subspace $I_d := \{p \in I,\ \deg(f) = d\} = I \cap R_d$. The *Hilbert function* of R/I is then defined as

$$\begin{array}{rccc} \mathcal{HF}_{R/I}: & \mathbb{N} & \longrightarrow & \mathbb{N} \\ & d & \longmapsto & \dim_{\mathbb{F}_q}(R_d/I_d) \end{array}$$

and the *Hilbert series* is

Definition 1 (Hilbert series)**.** Let $I \subset R$ be a homogeneous ideal. The Hilbert series of the quotient ring R/I is

$$\mathcal{H}_{R/I}(x) := \sum_{d=0}^{\infty} \mathcal{HF}_{R/I}(d) x^d.$$

Computing the Hilbert series of a general ideal is difficult. There are, however, an important class of polynomial systems, known as *semi-regular sequences*, where an expression for the Hilbert Series is known. Recall that a homogeneous ideal I is said to be zero-dimensional if R/I is a finite dimensional vector space. In this case, the *degree of regularity*, d_{reg}, is the smallest integer d such that $I_d = R_d$.

Definition 2 (Semi-regular sequence, [Bar04])**.** Let $\mathcal{P} := \{p_1, \ldots, p_m\}$ be a sequence of homogeneous polynomials such that $I := \langle \mathcal{P} \rangle$ is zero-dimensional with degree of regularity d_{reg}. The sequence $\mathcal{P}$ is said to be semi-regular if $I \neq R$ and if for $1 \leq i \leq m$, $g_i p_i = 0$ in $R/\langle p_1, \ldots, p_{i-1} \rangle$ with $\deg(g_i p_i) < d_{\text{reg}}$ implies $g_i = 0$ in $R/\langle p_1, \ldots, p_{i-1} \rangle$.

Proposition 1 ([Bar04])**.** *Let $\mathcal{P} := \{p_1, \ldots, p_m\}$ be a homogeneous semi-regular system where $\deg(p_i) = d_i$ for $1 \leq i \leq m$ and let $S_{m,n}(z) = \frac{\prod_{i=1}^{m}(1-x^{d_i})}{(1-x)^n}$. Then the Hilbert Series associated with $\mathcal{P}$ is given by*

$$\mathcal{H}_{R/\langle \mathcal{P} \rangle}(x) = \left[S_{m,n}(x)\right]^{+},$$

where $[\cdot]^{+}$ means truncation after the first non-positive coefficient.

The semi-regularity notion is extended to an affine sequence $\{p_1, \ldots, p_m\}$ by considering $\{p_1^h, \ldots, p_m^h\}$, where p_i^h denotes the homogeneous part of p_i of maximal degree.

3.2 A Conjecture on the Hilbert Series

We consider the affine ideal $\langle \mathcal{F} \rangle \subset R$ and $I := \langle \mathcal{F}^h \rangle$ the homogeneous ideal generated by the highest degree parts. The degree of regularity of I will play a crucial role in our complexity estimates for finding solutions of $\mathcal{F}$ (see Assumption 2 below). To derive this, we formulate a conjecture on the Hilbert series of R/I. Note that the ideal is guaranteed to be zero-dimensional due to the equations in $\mathcal{Z}^h$. In fact, we have a precise description of the Hilbert series of $R/\langle \mathcal{Z}^h \rangle$.

Lemma 1. *The Hilbert series of $S := R/\langle \mathcal{Z}^h \rangle$ is equal to*

$$\mathcal{H}_S(x) = (1 + x + \cdots + x^{z-1})^n = \left(\frac{1 - x^z}{1 - x}\right)^n.$$

Proof. As the equations from $\mathcal{Z}^h$ are univariate and do not depend on the coordinate index, we obtain

$$\mathcal{H}_S(x) = \left(\mathcal{H}_{\mathbb{F}_q[e]/\langle e^z \rangle}(x)\right)^n.$$

The fact that $\mathcal{H}_{\mathbb{F}_q[e]/\langle e^z \rangle}(x) = 1 + x + \cdots + x^{z-1}$ is clear. □

Our conjecture follows a similar strategy to [BØ23] and [CMT23, §5.3]. That is, we assume a generic behaviour of the $\mathcal{L}^h$ equations in $S = R/\langle \mathcal{Z}^h \rangle$. More formally, we can define a regular property by adapting Definition 2 to this quotient.

Definition 3 (Semi-regularity over S)**.** Let $\mathcal{P} := \{p_1, \ldots, p_m\}$ be a sequence of homogeneous polynomials in S such that $I := \langle \mathcal{P} \rangle$ is zero-dimensional with degree of regularity d_{reg}. The sequence $\mathcal{P}$ is said to be semi-regular if $I \neq S$ and if for $1 \leq i \leq m$, $g_i p_i = 0$ in $S/\langle p_1, \ldots, p_{i-1} \rangle$ with $\deg(g_i p_i) < d_{\text{reg}}$ implies $g_i = 0$ in $S/\langle p_1, \ldots, p_{i-1} \rangle$.

Note that while S is identical to the quotient defined by the top degree parts of the *field equations* from the field $\mathbb{F}_z$, the notion of semi-regularity over S is different from semi-regularity over $\mathbb{F}_z$. The main difference is that the coefficients of $\mathcal{P}$ do not belong in $\mathbb{F}_z$. This means that we do not expect "Frobenius-like" cancellations caused by p^z. In particular, Definition 3 is different from semi-regularity over $\mathbb{F}_2$ [Bar04, Definition 3.2.4] when $z = 2$.

Assumption 1. *We assume that the equations of the image of $\mathcal{L}^h$ in S satisfy Definition 3.*

If this assumption holds, we obtain

Proposition 2. *Under Assumption 1, the Hilbert series of $R/\langle \mathcal{F}^h \rangle$ is equal to*

$$\mathcal{H}_{R/\langle \mathcal{F}^h \rangle}(x) = \left[(1-x)^{n-k} \left(\frac{1-x^z}{1-x} \right)^n \right]_+ = \left[\frac{(1-x^z)^n}{(1-x)^k} \right]_+ ,$$

where $[\cdot]_+$ refers to the truncation after the first non-positive coefficient.

Proof. The zero-dimensional character of the ideal $\langle \mathcal{F}^h \rangle$ and Assumption 1 imply the relation

$$\mathcal{H}_{R/\langle \mathcal{F}^h \rangle}(x) = \left[(1-x)^{n-k} \mathcal{H}_S(x) \right]_+ ,$$

where $[\cdot]_+$ is the truncation after the first non-positive coefficient. This follows from a reasoning similar to [Bar04, §3.3.1], or more recently [BØ23]. We conclude by Lemma 1. □

While we are not able to prove Assumption 1, it was found to be valid in our computations of Hilbert series for several parameter sets, see Appendix B. Thus we will, in practice, use Proposition 2 as a conjecture on the full Hilbert series $\mathcal{H}_{R/\langle \mathcal{F}^h \rangle}$.

3.3 A Tighter Complexity Bound

The degree of regularity and Equation (4) capture the complexity of computing Gröbner bases for *homogeneous* polynomial systems. In general, a more suited measure of the degree reached for *affine* polynomial systems in a Gröbner basis algorithm like F_4 [Fau99] is the *solving degree* [CG23, Definition 1.1]. The exact relation between d_{reg} and the solving degree depends on the affine polynomials that are associated with non-trivial syzygies in $\langle \mathcal{F}^h \rangle$, and whether they reduce to zero modulo previous polynomials in the Gröbner basis computation. While there are polynomial systems where the solving degree and degree of regularity are different (see, e.g., [CG23, Example 4.3]), they typically coincide for polynomial systems that do not exhibit a particular algebraic structure. The two degrees are indeed found to be the same in all the experiments we have performed with the F_4 algorithm implemented in Magma. This leads to the following assumption, which will justify our use of d_{reg} in complexity estimates.

Assumption 2. *We assume that the solving degree of $\mathcal{F}$ coincides with the degree of regularity of the homogeneous ideal $\langle \mathcal{F}^h \rangle$.*

We have also verified that d_{reg} is equal to the first fall degree [CG23, Definition 1.3] for $\mathcal{F}$ (which is consistent with Assumption 1).

Under Assumption 1 and 2, we now have an explicit way of computing the degree d_{reg} that is used in Equation (4). That is, $d_{\text{reg}} = \deg(\mathcal{H}_{R/I}) + 1$, where $\mathcal{H}_{R/I}$ is the series in Proposition 2. However, we note that the binomial expression in Equation (4) counts all monomials of degree d_{reg} in n variables, which is an overestimate. Indeed, by first performing a reduction step modulo the $e_i^z - 1 = 0$ equations, we may instead consider a matrix whose columns are indexed by monomials whose partial degree is only $\leq z-1$ in each variable, that is the monomials in $S = R/\langle \mathcal{Z}^h \rangle$. Recall that the coefficient of the degree d term in the series of Lemma 1 counts the number of degree d monomials in S. The number of degree $\leq d$ monomials of this form is given by the coefficient of degree d in the following modification of this series.

$$\frac{1}{1-x}\mathcal{H}_S(x) = \frac{1}{1-x}\left(\frac{1-x^z}{1-x}\right)^n. \tag{5}$$

All in all, we can refine the cost estimate for computing a Gröbner basis of $\mathcal{F}$ by

Proposition 3. *Under Assumption 1 and 2, we estimate the complexity of solving $\mathcal{F}$ using Gröbner bases by*

$$\mathcal{O}(M^{\omega}_{z,(d_{n,k,z},n)}), \tag{6}$$

where $d_{n,k,z}$ is the degree of regularity derived from the Hilbert series in Proposition 2, $M_{z,(d_{n,k,z},n)}$ is the coefficient of degree $d_{n,k,z}$ in the series of (5), and $2 \leq \omega \leq 3$ is the linear algebra constant.

3.4 Hybrid Approach

In its plain form, the above attack performs poorly on the CROSS-R-SDP parameters of [BBB+23]. This motivates the study of hybrid techniques to improve the complexity. The standard hybrid approach corresponds to fixing several unknowns in $\mathcal{F}$. As the error vector is random in E^n and $|E| = z$, the success probability is $1/z$ each time we fix a variable. Similarly to previous works, such as [Bet12, Section 4,2] and [BØ23], we adopt the same genericity assumptions as in the plain case regarding specialized systems. More specifically, we fix f of the k last variables, i.e., $e_i,\ i \in \{n-k+1..n\}$. Note that the top degree parts of $\mathcal{P}$ will then be the same as the homogeneous polynomials associated with the parity-checks of the code defined by $\boldsymbol{H}$ shortened at the same f positions. Since Assumption 1 was on the highest degree components, its hybrid adaptation given in Assumption 3 does not depend on the vector of specialization $\boldsymbol{v} \in E^f$, only on f. For any $f \in \{0..k\}$, let $\boldsymbol{e}_f := (e_{f+1}, \ldots, e_n)$, let $\mathcal{Z}_f := \{\forall i \in \{f+1..n\},\ e_i^z - 1\}$ and let $S_f := \mathbb{F}_q[\boldsymbol{e}_f]/\langle \mathcal{Z}_f^h \rangle$.

Assumption 3. *For any $f \in \{0..k\}$ and any $\boldsymbol{v} \in E^f$, we assume that the system $\mathcal{F}_{spec,\boldsymbol{v},f}$ obtained by fixing the last f variables to $\boldsymbol{v}$ is semi-regular in S_f.*

In the same manner as Proposition 2, we can show the following result.

Proposition 4. *Under Assumption 3, the Hilbert series of $R/\langle \mathcal{F}^h_{spec,\boldsymbol{v},f} \rangle$ is equal to*

$$\mathcal{H}_{R/\langle \mathcal{F}^h_{spec,\boldsymbol{v},f} \rangle}(x) = \left[(1-x)^{n-k}\left(\frac{1-x^z}{1-x}\right)^{n-f}\right]_+ = \left[\frac{(1-x^z)^{n-f}}{(1-x)^{k-f}}\right]_+,$$

where $[\cdot]_+$ refers to the truncation after the first non-positive coefficient.

We also adopt an analogue to Assumption 2. Note that the following assumption does depend on the choice of $\boldsymbol{v} \in E^f$.

Assumption 4. *We assume that the solving degree of $\mathcal{F}_{spec,\boldsymbol{v},f}$ coincides with the degree of regularity of the homogeneous ideal $\langle \mathcal{F}^h_{spec,\boldsymbol{v},f} \rangle$.*

Finally, we follow the reasoning of Section 3.3 and obtain

Proposition 5. *Under Assumptions 3 and 4, we estimate the complexity of the standard hybrid approach on $\mathcal{F}$ using Gröbner bases by*

$$\mathcal{O}\left(\min_{\substack{0\leq f\leq k\\ d_{n,k,z,f}\geq z}} \left(z^f M^{\omega}_{z,(d_{n,k,z,f},n-f)}\right)\right), \tag{7}$$

where $d_{n,k,z,f}$ is the degree of regularity derived from the Hilbert series, where $M_{z,(d_{n,k,z,f},n-f)}$ is the coefficient of degree $d_{n,k,z,f}$ in the series $\frac{1}{1-x}\left(\frac{1-x^z}{1-x}\right)^{n-f}$ and where $2 \leq \omega \leq 3$ is the linear algebra constant.

On the CROSS parameters, this approach does not yield the most efficient attack. More precisely, for the 3 security levels, the best strategy using Equation (7) is to fix variables until we can solve at degree $z = 7$. This is the least degree where we can exploit the $\mathcal{Z}$ equations. Concretely, as we fix almost all k variables, the cost of this approach is not competitive with the R-SDP adaptation of Prange's algorithm.

A possible generalization of this approach is to consider univariate equations of degree $d_i \in \{1..z-1\}$ which vanish on the e_i's with probability < 1. These equations correspond to guessing subsets $E_i \subset E$ such that $e_i \in E_i$. (The method described above is then the case $|E_i| = 1$). Note that a similar approach has already been performed on MQ systems, see [Bet12, §4.4 p. 111]. However, testing different values of d_i suggests that this generalization would still not outperform Prange on the CROSS parameters.

3.5 Asymptotic analysis

The goal of this section is to give an asymptotic equivalent of the degree of regularity when the length n tends to infinity assuming a constant code rate $R := 1/\alpha := k/n$. An initial observation is that the conjectured Hilbert series for $\mathcal{F}$ is the same as the one of a semi-regular system containing n equations of degree z in k variables. Thus we can leverage the technical machinery from the work of Bardet, Faugère and Salvy [Bar04, BFS05].

Theorem 1 ([Bar04], Theorem 4.1.3 p. 81)**.** *For any constant $\alpha > 1$, an asymptotic equivalent of the degree of regularity of a semi-regular sequence of $n = \alpha k$ equations with degrees $d_1, \dots, d_n$ in k variables when $n \to +\infty$ is*

$$d_{reg} \sim \phi(x_0)k,$$

where

$$\phi(x) = \frac{x}{1-x} - \frac{1}{k}\sum_{j=1}^{n} \frac{d_i x^{d_i}}{1-x^{d_i}}, \tag{8}$$

and where x_0 is the root of ϕ' such that $\phi(x_0) > 0$ is minimal.

The function ϕ relevant to the setting $d_i = z$ for all $i \in \{1..n\}$ is $\phi(x) = \frac{x}{1-x} - \alpha\frac{zx^z}{1-x^z}$. The consequence for the R-SDP setting of CROSS is

Lemma 2. *Assuming a constant code rate $R := 1/\alpha$, there exists a constant $c_{\alpha,z} > 0$ such that the degree of regularity of $\langle \mathcal{F}^h \rangle$ behaves as*

$$d_{n,k,z} \sim c_{\alpha,z}k.$$

We can precisely give the value of $c_{\alpha,z}$ when z is small. We showcase this for the cases $z = 2$ and 3 For the former case, we may rely on the study of quadratic equations performed in [Bar04].

Lemma 3 ([Bar04], Corollary 4.4.1 p. 95)**.** *When $z = 2$, an equivalent of the degree of regularity d_{reg} when k goes to infinity is $d_{reg} \sim c_{\alpha,2}k$, where*

$$c_{\alpha,2} = -\frac{1}{2} + \alpha - \sqrt{\alpha(\alpha - 1)}.$$

The case $z = 3$ can be obtained by a resultant computation, namely

Lemma 4. *When $z = 3$, an equivalent of the degree of regularity d_{reg} when k goes to infinity is $d_{reg} \sim c_{\alpha,3}k$, where*

$$c_{\alpha,3} = -\frac{1}{2} + \frac{3\alpha}{2} - \frac{\sqrt{81\alpha^2 - 24\sqrt{\alpha} - 54\alpha - 3}}{6}.$$

Proof. We have to study the roots of the derivative ϕ', where

$$\phi(x) = \frac{x}{1-x} - 3\alpha\frac{x^3}{1-x^3}.$$

More precisely, we look for the smallest possible value of $u = \phi(x_0)$ for such a root x_0. Since ϕ and ϕ' are rational fractions, we may rewrite

$$\phi(x_0) - u = 0, \qquad \phi'(x_0) = 0,$$

as a polynomial system

$$P(x_0, u) = 0, \qquad Q(x_0, u) = 0,$$

where the polynomials P and Q can be easily computed. To eliminate x_0, we consider the resultant $T(Y) = \mathrm{Res}_X(P(X,Y), Q(X,Y))$ which is a polynomial of degree 4. Its roots sorted in decreasing order are

$$\begin{cases} -\frac{1}{2} + \frac{3\alpha}{2} + \frac{\sqrt{81\alpha^2 + 24\sqrt{\alpha} - 54\alpha - 3}}{6} \\ -\frac{1}{2} + \frac{3\alpha}{2} + \frac{\sqrt{81\alpha^2 - 24\sqrt{\alpha} - 54\alpha - 3}}{6} \\ -\frac{1}{2} + \frac{3\alpha}{2} - \frac{\sqrt{81\alpha^2 - 24\sqrt{\alpha} - 54\alpha - 3}}{6} \\ -\frac{1}{2} + \frac{3\alpha}{2} - \frac{\sqrt{81\alpha^2 + 24\sqrt{\alpha} - 54\alpha - 3}}{6}. \end{cases}$$

The two greatest roots are positive as $\alpha > 1 > 1/3$. The explicit expression of the resultant also shows that the product of roots is negative. This implies that we eventually keep the value

$$-\frac{1}{2} + \frac{3\alpha}{2} - \frac{\sqrt{81\alpha^2 - 24\sqrt{\alpha} - 54\alpha - 3}}{6}.$$

$\square$

Degrees $z \geq 4$ can still be tackled in the same fashion. The technical difficulty is that we will handle a resultant of larger degree and thus we are no longer guaranteed to have a closed form expression for its roots.

Finally, let us return to the hybrid approach. We note that the conjectured Hilbert series for $R/\langle \mathcal{F}^h_{\mathrm{spec},\boldsymbol{v},\boldsymbol{f}} \rangle$ is also the Hilbert series of a semi-regular system, which means that we can obtain a value for d_{reg} in the same manner as in the plain case. Based on this equivalence, a possible next step could be to derive the best asymptotic trade-off following the approach of [BFP09, Bet12] (see also [BS24, §4.4] for a recent use of this technique).

References

[AAC$^+$22] G Alagic, D Apon, D Cooper, Q Dang, T Dang, J Kelsey, J Lichtinger, Yi-Kai Liu, C Miller, D Moody, R Peralta, R Perlner, A Robinson, and D Smith-Tone. Status report on the third round of the NIST post-quantum cryptography standardization process, 2022. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Interagency or Internal Report (IR) 8413. URL: https://doi.org/10.6028/NIST.IR.8413.

[Bar04] Magali Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université Paris VI, December 2004. http://tel.archives-ouvertes.fr/tel-00449609/en/.

[BBB$^+$23] Marco Baldi, Alessandro Barenghi, Sebastian Bitzer, Patrick Karl, Felice Manganiello, Alessio Pavoni, Gerardo Pelosi, Paolo Santini, Jonas Schupp, Freeman Slaughter, Antonia Wachter-Zeh, and Violetta Weger. CROSS: Codes and Restricted Objects Signature Scheme. Submission to the NIST Post-Quantum Cryptography Standardization Project, 2023.

[BBC$^+$20] Marco Baldi, Massimo Battaglioni, Franco Chiaraluce, Anna-Lena Horlemann-Trautmann, Edoardo Persichetti, Paolo Santini, and Violetta Weger. A new path to code-based signatures via identification schemes with restricted errors. *arXiv preprint arXiv:2008.06403*, 2020.

[BBP$^+$23] Marco Baldi, Sebastian Bitzer, Alessio Pavoni, Paolo Santini, Antonia Wachter-Zeh, and Violetta Weger. Zero knowledge protocols and signatures from the restricted syndrome decoding problem. Cryptology ePrint Archive, Paper 2023/385, 2023. URL: https://eprint.iacr.org/2023/385.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993). URL: http://dx.doi.org/10.1006/jsco.1996.0125, `doi:10.1006/jsco.1996.0125`.

[Bet12] Luk Bettale. *Cryptanalyse algébrique : outils et applications.* PhD thesis, Université Pierre et Marie Curie - Paris 6, 2012.

[BFP09] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2009.

[BFS05] Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. Asymptotic Behaviour of the Index of Regularity of Semi-Regular Quadratic Polynomial Systems. In *MEGA 2005*, pages 1–17, May 2005. URL: https://hal.science/hal-01486845.

[BØ23] Pierre Briaud and Morten Øygarden. A new algebraic approach to the regular syndrome decoding problem and implications for pcg constructions. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 391–422, Cham, 2023. Springer Nature Switzerland.

[BS24] Charles Bouillaguet and Julia Sauvage. Preliminary cryptanalysis of the biscuit signature scheme. Cryptology ePrint Archive, Paper 2024/148, 2024. https://eprint.iacr.org/2024/148. URL: https://eprint.iacr.org/2024/148.

[CG23] Alessio Caminata and Elisa Gorla. Solving degree, last fall degree, and related invariants. *Journal of Symbolic Computation*, 114:322–335, 2023.

[CLO13] David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: an Introduction to Computational Algebraic Geometry and Commutative Algebra.* Springer Science & Business Media, 2013.

[CMT23] Alain Couvreur, Rocco Mora, and Jean-Pierre Tillich. A new approach based on quadratic forms to attack the McEliece cryptosystem. In Ron Steinfeld Jian Guo, editor, *ASIACRYPT 2023*, Guangzhou, China, December 2023. Springer. 68 pages (Long version). URL: https://inria.hal.science/hal-04215135.

[Dum91] Ilya Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.

[Fau99] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.

[Fau02] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, 2002.

[Ste89] Jacques Stern. A method for finding codewords of small weight. In Gérard Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications*, pages 106–113, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.

[VOW99] Paul C Van Oorschot and Michael J Wiener. Parallel collision search with cryptanalytic applications. *Journal of cryptology*, 12:1–28, 1999.

A Details for the Comparison of Collision Attacks

A.1 An Overview of the Collision Attack in [BBB+23]

We start with a brief, high-level overview of the collision attack introduced in version 1.1 of [BBB+23]. Consider the $(n-m)\times n$ parity-check matrix $\mathbf{M}_H$ of $\mathbf{M}_G$, and let $\langle \mathbf{M}_H \rangle$ denote its associated code. For simplicity, let us suppose that there are two subcodes $\mathcal{C}_1, \mathcal{C}_2 \subset \langle \mathbf{M}_H \rangle$ of dimensions d_1, d_2 and disjoint support J_1 and J_2, respectively. Moreover, we write $j_1 := |J_1|$, $j_2 := |J_2|$, and assume for simplicity that the subcodes are chosen such that $\rho := j_1 - d_1 = j_2 - d_2$. We further introduce the notation

$$l := j_1 + j_2 - k, \qquad \tilde{l} := 2\rho - m.$$

After a reordering of the columns, the subcode $\mathcal{C}_1$ is generated by the matrix $\begin{bmatrix}\mathbf{0} & \mathbf{G}_1 & \mathbf{0}\end{bmatrix}$, for a matrix $\mathbf{G}_1 \in \mathbb{F}_z^{d_1\times j_1}$ representing the support J_1. Similarly, we have that $\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{G}_2\end{bmatrix}$ generates $\mathcal{C}_2$ for some $\mathbf{G}_2 \in \mathbb{F}_z^{d_2\times j_2}$. Upon performing suitable linear operations on the matrices $\mathbf{M}_H$ and $\begin{bmatrix}\mathbf{H} & \mathbf{s}^\intercal\end{bmatrix}$ one obtains

$$\mathbf{M}'_H = \begin{pmatrix} * & * & * \\ \mathbf{0} & \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{0} & \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_2 \end{pmatrix}, \qquad \mathbf{H}' = \begin{pmatrix} * & * & * & * \\ \mathbf{0} & \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{s}_2'^{\top} \end{pmatrix} \tag{9}$$

for matrices $\mathbf{M}_1 \in \mathbb{F}_z^{\tilde{l}\times j_1}$, $\mathbf{M}_2 \in \mathbb{F}_z^{\tilde{l}\times j_2}$, $\mathbf{H}_1 \in \mathbb{F}_q^{l\times j_1}$ and $\mathbf{H}_2 \in \mathbb{F}_q^{l\times j_2}$. The idea is now to make lists from the kernel elements of $\mathbf{B}_1^\intercal$ and $\mathbf{B}_2^\intercal$, and use information from both $\mathbf{M}'_H$ and $\mathbf{H}'$ to search for collisions. More precisely, the two lists are created as

$$\begin{gathered} \left\{ (\mathbf{x}_1, \mathbf{x}_1\mathbf{M}_1^\intercal, g^{(\mathbf{0}\,\mathbf{x}_1\,\mathbf{0})} \begin{pmatrix}\mathbf{0} & \mathbf{H}_1 & \mathbf{0}\end{pmatrix}^\intercal) \,\middle|\, \text{for } \mathbf{x}_1 \in \mathrm{Ker}(\mathbf{B}_1^\intercal) \right\}, \\ \left\{ (\mathbf{x}_2, \mathbf{x}_2\mathbf{M}_2^\intercal, \mathbf{s}'_2 - g^{(\mathbf{0}\,\mathbf{0}\,\mathbf{x}_2)} \begin{pmatrix}\mathbf{0} & \mathbf{0} & \mathbf{H}_2\end{pmatrix}^\intercal) \,\middle|\, \text{for } \mathbf{x}_2 \in \mathrm{Ker}(\mathbf{B}_2^\intercal) \right\}. \end{gathered} \tag{10}$$

The latter $\mathbb{F}_z^{\tilde{l}} \times \mathbb{F}_q^{l}$ part of a list element is called the *label*, and is used to find collisions. Indeed, it can be verified that a solution to the R-SDP(G) problem will correspond to a $\mathbb{F}_z$-tuple whose last $j_1 + j_2$ entries form an element $(\mathbf{x}_1, \mathbf{x}_2) \in \mathrm{Ker}(\mathbf{B}_1^\top) \times \mathrm{Ker}(\mathbf{B}_2^\top)$, where

$$\mathbf{x}_1\mathbf{M}_1^\top + \mathbf{x}_2\mathbf{M}_2^\top = 0; \text{ and}$$
$$g^{(\mathbf{0}\,\mathbf{x}_1\,\mathbf{x}_2)} \begin{pmatrix}\mathbf{0} & \mathbf{H}_1 & \mathbf{H}_2\end{pmatrix}^\top = s_2'.$$

B Extra Details on the Algebraic Attack

We ran several experiments in the computer algebra system Magma to verify Assumption 1 and 2. The results are reported in the following.

F_4 algorithm. In Tables 3 and 4, we compare the solving degree d_{solv} of $\mathcal{F}$ to the first degree fall d_{ff}. Both quantities have been obtained from Magma's implementation of F_4. We also indicate the *degree falls* generated at the step in degree d_{ff}. We notice that both degrees coincide in all cases. The computations for the larger instances were fairly time consuming. For instance, the total time to obtain the last rows in Tables 3 and 4 was 441852.309s and 227555.690s respectively.

Table 4: Parameters $q = 127$, $z = 7$ and $k = n/2$.

n	d_{ff}	d_{solv}
8	9 (6:15 7:40 8:5)	9
10	10 (7:54 8:145 9:15)	10
12	11 (8:245 9:490 10:45)	11
14	12 (9:1204 10:1631 11:77)	12
16	12 (9:2660 10:6896 11:3492)	12
18	13 (12:2925 13:12630)	13

n	d_{ff}	d_{solv}
8	13 (10:21 11:84 12:7)	13
10	14 (13:195 14:169)	14
12	16 (14:1833 15:456)	16
14	17 (16:3521 17:4816)	17

Hilbert series. For the same parameters, we computed to Hilbert series of $R/\langle \mathcal{F}^h \rangle$. This series was always in accordance with the one of Proposition 2. The degree of regularity derived from it was also identical to the first degree fall d_{ff} from above.