Zero Knowledge Protocols and Signatures from the Restricted Syndrome Decoding Problem

Marco Baldi[1], Sebastian Bitzer[2], Alessio Pavoni[1], Paolo Santini[1], Antonia Wachter-Zeh[2], and Violetta Weger[2]

[1] Polytechnic University of Marche, Italy
[2] Technical University of Munich, Germany

Abstract. The Restricted Syndrome Decoding Problem (R-SDP) corresponds to the Syndrome Decoding Problem (SDP) with the additional constraint that all entries of the solution error vector must live in a fixed subset of the finite field. In this paper, we study how this problem can be applied to the construction of signatures derived from Zero-Knowledge (ZK) protocols. First, we show that R-SDP appears to be well-suited for this type of application: ZK protocols relying on SDP can easily be modified to use R-SDP, resulting in significant reductions in the communication cost. We then introduce and analyze a variant of R-SDP, which we call R-SDP(G), with the property that solution vectors can be represented with a number of bits that is slightly larger than the security parameter (which clearly provides an ultimate lower bound). This enables the design of competitive ZK protocols. We show that existing ZK protocols can greatly benefit from the use of R-SDP, achieving signature sizes in the order of 7 kB, which are smaller than those of several other schemes submitted to NIST's additional call for post-quantum digital signatures.

Keywords: Code-based Cryptography · Post-Quantum Cryptography · Restricted Errors · Signature Scheme · Syndrome Decoding Problem.

1 Introduction

In 2023, the National Institute of Standards and Technology (NIST) reopened the standardization call for post-quantum cryptography, targeting solely signature schemes, preferably not based on structured lattices.[3] Arguably, this additional call has shifted the focus of the cryptographic community to finding new and efficient post-quantum signature schemes.

In particular, over the last years, significant attention has been dedicated to schemes obtained via the Fiat-Shamir transform on a Zero-Knowledge (ZK) interactive protocol. As a matter of fact, signatures derived from this paradigm are now perceived as one of the most promising solutions. This is visible within

[3] See, e.g., the official NIST call `https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf`

the round 1 submissions to NIST's additional call: out of the 40 submitted schemes, 15 candidates are based on such a paradigm.

The Fiat-Shamir transform works by making a ZK protocol non-interactive: The signer simulates an execution of the protocol, binding it to the message-to-be-signed (thanks to the one-wayness of hash functions), and the signature is composed by the *transcript*, i.e., the list of messages that are exchanged during the protocol execution. When the *soundness error* of the considered protocol is too high, a certain number of repetitions is needed to avoid efficient forgeries; in such a case, the number of exchanged messages increases, and the signature size grows, as well.

Historically, the large signature size has been the Achilles' heel of these types of signature schemes. However, the panorama has greatly changed in the past few years thanks to various techniques and optimizations capable of compressing signatures. One of the most popular approaches consists of designing protocols with very low *soundness error*; this reduces the number of repetitions and, consequently, leads to shorter transcripts. In particular, for problems related to decoding (e.g., the Syndrome Decoding Problem (SDP) and the Permuted Kernel Problem (PKP)), popular approaches are the so-called protocol-with-helper [17] and the Multi-Party Computation (MPC) in-the-Head (MPCitH) paradigm [33].

1.1 Our Contribution

In this paper, we study ZK protocols derived from the Restricted Syndrome Problem (R-SDP), introduced in [11][4], which adds to the classical SDP the constraint that each entry must live in a fixed subset $\mathbb{E}$ of the underlying finite field. We first describe the problem in its full generality and then move to the particular version with full Hamming weight and $\mathbb{E}$ being a cyclic subgroup of the multiplicative group.

We show that the restricted setting can be tailored so that many existing ZK protocols receive a significant reduction in the communication cost. This happens because of two phenomena. First, in R-SDP the error can have a much larger Hamming weight, even maximum (that is, no null entry), while still having a unique solution to the problem. This increases the cost of Information Set Decoding (ISD) algorithms, and, as a matter of fact, with R-SDP we can achieve the same security level using smaller codes. Another important improvement is due to the transformations used in ZK protocols. For the classical SDP, they are given by a monomial transformation (a permutation with scaling factors), as these are the transitive linear maps acting on the Hamming sphere. For the new R-SDP, component-wise multiplication with restricted vectors is enough since these linear maps act transitively on the set of restricted vectors. This yields another significant reduction in the communication cost.

Then, we derive a special version of R-SDP, called R-SDP(G), with which the performances of ZK protocols can be further boosted. With R-SDP(G), the

[4] A similar idea was already mentioned in [42], but it was not used in conjunction with a decoding problem.

solution space is a subgroup $G \leq \mathbb{E}^n$, whose size can be tuned to minimize the communication cost. Namely, for a security of λ bits, R-SDP(G) uses a solution space G of size $2^{(1+\alpha)\lambda}$, where $\alpha \geq 0$ is a small constant (say, $\alpha \leq 1$). From a mathematical point of view, G is a group that acts transitively and freely on itself: this implies that we can sample any restricted object (i.e., secret keys and hiding transformations) from G and represent its elements using only $(1+\alpha)\lambda \leq 2\lambda$ bits. The value of α is chosen from a conservative perspective so that existing attacks cannot be sped up considering the knowledge of G.

Finally, we apply R-SDP and R-SDP(G) on modern ZK protocols, namely the GPS scheme [31] and BG scheme for the Permuted Kernel Problem (PKP) [18]. We call the newly derived schemes R-GPS and R-BG, respectively, and show that moving to R-SDP and R-SDP(G) leads to significant reductions in the communication cost and in signature sizes, as well. In fact, for R-GPS, we almost halve the signature sizes, while for R-BG we achieve important savings: using R-SDP(G), we obtain signatures with a size of 7.8 kB (instead of 10.0 kB) for the fast variant, and 7.2 kB (instead of 8.9 kB) for the short variant. We also provide timings for a (non-optimized) Proof of Concept implementation for R-BG, which confirms that the proposed protocols are practical.

The work in this paper lies the foundation of CROSS [10], one of the schemes submitted to the NIST call for additional signatures. CROSS uses R-SDP and R-SDP(G), applied to a basic ZK protocol inspired from [20], with soundness error $\approx 1/2$. Signature sizes are in the same ballpark as those of other ZK/MPCitH schemes, but CROSS is one of the fastest schemes in this category. This is made possible by using R-SDP and R-SDP(G). Indeed, techniques to reduce the soundness error normally come with the price of some significant computational overhead. Since using R-SDP and R-SDP(G) leads to very compact messages, CROSS can use a simple but highly efficient protocol and still achieve sufficiently short signatures.

1.2 Paper Organization

Section 2 settles the notation we use and gives (minimal) preliminaries about linear codes and ZK protocols. In Section 3, we formally introduce R-SDP, show how it can be solved using Information Set Decoding (ISD), and show that R-SDP can be much harder than SDP. We then move to the special case of full-weight vectors and $\mathbb{E}$ being a subgroup of $\mathbb{F}_q^*$, describing generic decoders tailored to this setting. In Section 4, we show how R-SDP can be applied to ZK protocols, using the well-known example of CVE [20], and argue why this leads to very promising schemes. In Section 5, we introduce another variant of R-SDP, called R-SDP(G), and analyze its security.

In Section 6, we apply R-SDP and R-SDP(G) to modern protocols, namely GPS [31] and BG [18], resulting in the schemes called R-GPS and R-BG. We compare the two schemes to existing ones in Section 7 and show their competitiveness. Finally, Section 8 concludes the paper.

2 Notation and Preliminaries

We use $[a; b]$ to denote the set of all reals $x \in \mathbb{R}$ such that $a \leq x \leq b$. For a finite set A, the expression $a \xleftarrow{\$} A$ means that a is chosen uniformly at random from A. In addition, we denote by $|A|$ the cardinality of A, by A^C its complement and by $A_0 = A \cup \{0\}$. As usual, for q being a positive integer, we denote by $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ the ring of integers modulo q. For a prime power q, we denote by $\mathbb{F}_q$ the finite field of order q and by $\mathbb{F}_q^*$ its multiplicative group. For $g \in \mathbb{F}_q^*$, we denote by $\mathrm{ord}(g)$ its multiplicative order.

We use uppercase (resp. lowercase) letters to indicate matrices (resp. vectors). If J is a set, we use $\mathbf{A}_J$ to denote the matrix formed by the columns of $\mathbf{A}$ indexed by J; analogous notation will be used for vectors. The identity matrix of size m is denoted as $\mathbf{I}_m$. We use $\mathbf{0}$ to denote the null matrix or the null vector without specifying dimensions (which will always be clear from the context). We denote by S_n the symmetric group of order n. Finally, we denote by $h_q(x) = x \log_q(q-1) - x \log_q(x) - (1-x) \log_q(1-x)$ the q-ary entropy function.

2.1 Cryptographic Tools

Throughout the paper, we adopt conventional cryptographic notations, e.g., λ denotes the security parameter. Standard functions are always implicitly defined, e.g., Hash indicates a secure hash function with digests of size 2λ. We focus on Zero-Knowledge (ZK) protocols, that is, interactive protocols in which a *prover* aims to convince a *verifier* that she knows a secret that verifies some public statement. Informally, a protocol achieves the ZK property when the interaction between the two parties reveals no information about the specific secret held by the prover. We say that a protocol has *soundness error* ε if a cheating prover, i.e., someone that does not know the secret, can convince the honest verifier with probability ε. When t parallel repetitions of a (public coin) ZK protocol with soundness error ε are considered, and the verifier only accepts if each of the repetitions is accepted, we obtain a new protocol with soundness error ε^t. Due to lack of space, we do not provide formal definitions for such properties, as they are standard and can be found in the literature (e.g., [28] nicely recaps all the necessary background).

ZK protocols can be turned into signature schemes with the Fiat-Shamir transform [30]. For 5-pass protocols (i.e., the number of messages that are exchanged in each execution is 5), the number of parallel executions shall be chosen taking into account the attack in [35]. Namely, setting $\varepsilon^t < 2^{-\lambda}$ may not be enough to protect against forgery attacks. The authors of [35] describe how to properly choose t, and when needed, we use their formula (see Equation (9)).

2.2 Linear Codes

A *linear code* $\mathscr{C}$ over the finite field $\mathbb{F}_q$ with length n and dimension k is a k-dimensional linear subspace of $\mathbb{F}_q^n$. We say that a code of length n and dimension k has *rate* $R = \frac{k}{n}$ and *redundancy* $r = n - k$.

A compact representation for a code is a *generator matrix*, that is, a full-rank $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ such that $\mathscr{C} = \{\mathbf{uG} \mid \mathbf{u} \in \mathbb{F}_q^k\}$. Equivalently, one can represent a code through a full-rank $\mathbf{H} \in \mathbb{F}_q^{r\times n}$, called *parity-check matrix*, such that $\mathscr{C} = \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{cH}^\top = \mathbf{0}\}$. The *syndrome* of some $\mathbf{x} \in \mathbb{F}_q^n$ is the length-r vector $\mathbf{s} = \mathbf{xH}^\top$. A set $J \subseteq \{1, \ldots, n\}$ of size k is called *information set* for $\mathscr{C}$ if $|\mathscr{C}_J| = q^k$, where $\mathscr{C}_J = \{\mathbf{c}_J \mid \mathbf{c} \in \mathscr{C}\}$. It directly follows that $\mathbf{G}_J$ and $\mathbf{H}_{J^C}$ are invertible matrices. We say that a generator matrix, respectively, a parity-check matrix, is in systematic form (with respect to the information set J), if $\mathbf{G}_J = \mathbf{I}_k$, respectively $\mathbf{H}_{J^C} = \mathbf{I}_r$.

We endow the vector space $\mathbb{F}_q^n$ with the *Hamming metric*: given $\mathbf{x} \in \mathbb{F}_q^n$, its Hamming weight $\mathrm{wt}(\mathbf{x})$ is the number of non-zero entries. The *minimum distance* of a linear code is given by $d(\mathscr{C}) = \min\{\mathrm{wt}(\mathbf{c}) \mid \mathbf{c} \in \mathscr{C}, \mathbf{c} \neq \mathbf{0}\}$. The *relative minimum distance* of a code is then denoted by $\delta = d(\mathscr{C})/n$. It is well known that random codes with sufficiently large length n attain the Gilbert-Varshamov (GV) bound: for a random code, we may assume $\delta = h_q^{-1}(1 - R)$.

Code-based cryptography usually relies on the following NP-complete problem [13, 16].

Problem 1 Syndrome Decoding Problem (SDP)
Given $\mathbf{H} \in \mathbb{F}_q^{r\times n}, t \in \mathbb{N}, \mathbf{s} \in \mathbb{F}_q^r$, *decide if there exists* $\mathbf{e} \in \mathbb{F}_q^n$, *such that* $wt(\mathbf{e}) \leq t$ *and* $\mathbf{eH}^\top = \mathbf{s}$?

We usually assume that the instance of the SDP is chosen uniformly at random, thus also that the code with parity-check matrix $\mathbf{H}$ attains the GV bound. If the target weight t is less than the minimum distance δn of the GV bound, we expect to have on average a unique solution, since the average number of solutions is given by $q^{n(h_q(\delta)-1+R)} \leq 1$.

3 The Restricted Syndrome Decoding Problem

Let us consider some subset $\mathbb{E}$ of $\mathbb{F}_q^*$, denote by $\mathbb{E}_0 = \mathbb{E} \cup \{0\}$ and by

$$\mathcal{S}_w^{\mathbb{E}} \coloneqq \{\mathbf{x} \in \mathbb{E}_0^n \mid \mathrm{wt}(\mathbf{x}) = w\}$$

the *Hamming sphere with radius* w *and restriction* $\mathbb{E}$. Clearly, for $\mathbb{E}$ of size z, we have $\left|\mathcal{S}_w^{\mathbb{E}}\right| = \binom{n}{w} z^w$. The Restricted Syndrome Decoding Problem (R-SDP), introduced in [11], reads as follows.

Problem 2 Restricted Syndrome Decoding Problem (R-SDP)
Given $\mathbf{H} \in \mathbb{F}_q^{r\times n}$, $\mathbf{s} \in \mathbb{F}_q^r$ *and* $w \in \mathbb{N}$, *decide if there exists* $\mathbf{e} \in \mathcal{S}_w^{\mathbb{E}}$, *such that* $\mathbf{eH}^\top = \mathbf{s}$.

It is easy to see that R-SDP is strongly related to other well-known hard problems. For instance, when $\mathbb{E} = \mathbb{F}_q^*$, the R-SDP corresponds to the classical SDP and if $\mathbb{E} = \{1\}$, the R-SDP is similar to the Subset Sum Problem (SSP) over finite fields.

Consequently, it is unsurprising that R-SDP is NP-complete for any choice of $\mathbb{E}$. The proof is essentially the same as in [11], where the authors focus on the case $\mathbb{E} = \{\pm x_1, \pm x_2, \ldots, \pm x_a\}$. Another proof can be immediately obtained from [43], whenever $1 \in \mathbb{E}$.

We always consider that the R-SDP instance is chosen uniformly at random. We expect to have on average (at most) a unique solution if w is such that

$$\binom{n}{w} z^w q^{k-n} \leq 1. \tag{1}$$

Let $W = w/n \in [0;1]$; since $\binom{n}{w} = 2^{n \cdot h_2(W) \cdot \left(1+o(1)\right)}$, we rewrite Equation (1) as

$$2^{n(h_2(W) + W\log_2(z) - (1-R)\log_2(q))} \leq 1.$$

Let W^* be the maximum value of W for which a random instance of R-SDP is expected to have a unique solution, that is

$$W^* = \max\left\{W \in [0;1] \mid h_2(W) + W\log_2(z) - (1-R)\log_2(q) \leq 0\right\}. \tag{2}$$

Comparing this to the GV bound, we can see that with the R-SDP, we can choose a much larger weight w and still guarantee the uniqueness of the solution. This is a crucial difference with SDP, since a high Hamming weight corresponds to an exponentially large number of solutions [24]. Note that if $\log_2(z) \leq (1 - R)\log_2(q)$, we even have uniqueness for full-weight vectors.

3.1 Solving R-SDP

To compare the computational complexity of R-SDP with classical SDP, we provide an adaption of the Stern/Dumer algorithm [26,40], which works for any choice of $\mathbb{E}$. Notice that, depending on the structure of $\mathbb{E}$, this algorithm can be improved using *representations*. For this, we refer to Appendix B.

Although the Stern/Dumer algorithm is well-known, we will provide full details in the following for the sake of completeness. As a first step, we choose a set $J \subset \{1, \ldots, n\}$ of size $k + \ell$ which contains an information set and perform a Partial Gaussian Elimination (PGE) on $\mathbf{H}$ in the columns indexed by J, obtaining $\widetilde{\mathbf{H}}$, and perform the same operations on the syndrome. For simplicity, let us assume that J is chosen in the first $k + \ell$ positions, thus

$$\mathbf{e}\widetilde{\mathbf{H}}^\top = (\mathbf{e}_1, \mathbf{e}_2) \begin{pmatrix} \mathbf{H}_1 & \mathbf{I}_{r-\ell} \\ \mathbf{H}_2 & \mathbf{0} \end{pmatrix}^\top = (\mathbf{s}_1, \mathbf{s}_2),$$

where $\mathbf{e}_1 \in \mathbb{E}_0^{k+\ell}, \mathbf{e}_2 \in \mathbb{E}_0^{r-\ell}, \mathbf{H}_1 \in \mathbb{F}_q^{(r-\ell)\times(k+\ell)}, \mathbf{H}_2 \in \mathbb{F}_q^{\ell\times(k+\ell)}, \mathbf{s}_1 \in \mathbb{F}_q^{r-\ell}$ and $\mathbf{s}_2 \in \mathbb{F}_q^{\ell}$. Thus, we get two syndrome equations, being

$$\mathbf{e}_1\mathbf{H}_1^\top + \mathbf{e}_2 = \mathbf{s}_1, \quad \mathbf{e}_1\mathbf{H}_2^\top = \mathbf{s}_2.$$

We solve these equations by requiring that $\mathbf{e}_1$ has weight v; for each such $\mathbf{e}_1$, it is enough to check that $\mathbf{s}_1 - \mathbf{e}_1\mathbf{H}_1^\top$ has weight $w-v$ and entries in $\mathbb{E}_0$. To solve the smaller instance given by $\mathbf{H}_2, \mathbf{s}_2$ and v, we use a collision search.

To improve readability, we drop any rounding operations and implicitly assume that all parameters can be chosen as integers. For this, we write $\mathbf{e}_1 = (\mathbf{x}_1, \mathbf{x}_2)$ with $\mathbf{x}_i$ of length $(k+\ell)/2$ and weight $v/2$. Thus, we also split $\mathbf{H}_2 = (\mathbf{A}_1, \mathbf{A}_2)$ and construct the two lists

$$\mathscr{L}_1 \coloneqq \{(\mathbf{x}_1, \mathbf{x}_1\mathbf{A}_1^\top) \mid \mathbf{x}_1 \in \mathbb{E}_0^{(k+\ell)/2}, \mathrm{wt}(\mathbf{x}_1) = v/2\},$$
$$\mathscr{L}_2 \coloneqq \{(\mathbf{x}_2, \mathbf{s}_2 - \mathbf{x}_2\mathbf{A}_2^\top) \mid \mathbf{x}_2 \in \mathbb{E}_0^{(k+\ell)/2}, \mathrm{wt}(\mathbf{x}_2) = v/2\}.$$

We then check for collisions, that is, all pairs $(\mathbf{x}_1, \mathbf{a}) \in \mathscr{L}_1$, $(\mathbf{x}_2, \mathbf{a}) \in \mathscr{L}_2$. The sought-after error vector is then given by $\mathbf{e}_1 = (\mathbf{x}_1, \mathbf{x}_2)$ and $\mathbf{e}_2 = \mathbf{s}_1 - \mathbf{e}_1\mathbf{H}^\top$.

Proposition 1. *The cost of the restricted Stern/Dumer algorithm given in Algorithm 1 is in*

$$\mathcal{O}\left(\binom{n}{w}\binom{(k+\ell)/2}{v/2}^{-2}\binom{r-\ell}{w-v}^{-1} \cdot \left(\binom{(k+\ell)/2}{v/2} z^{v/2} + \binom{k+\ell}{v} z^v q^{-\ell}\right)\right).$$

Proof. The two lists are of size $L = \binom{(k+\ell)/2}{v/2} z^{v/2}$ and the collision search costs approximately $L^2 q^{-\ell} = \binom{k+\ell}{v} z^v q^{-\ell}$, as the probability of a random vector having a fixed syndrome of length ℓ is $q^{-\ell}$. The number of iterations required, is given by the inverse of the success probability of one iteration, namely that the error vector is such that $\mathbf{e}_1 = (\mathbf{x}_1, \mathbf{x}_2)$ with $\mathbf{x}_i$ of weight $v/2$. Thus, the probability is $\binom{(k+\ell)/2}{v/2}^2 \binom{r-\ell}{w-v}\binom{n}{w}^{-1}$. The cost of this restricted Stern/Dumer algorithm is then given by the number of expected iterations, times the cost of one iteration, which consists of constructing the lists $\mathscr{L}_i$ and the collision search. □

Remark 1. Let $L = \ell/n$, then in the case of $w = n$, the optimized cost of Stern's algorithm is in $O\big(2^{F(R,q,z)n}\big)$, where

$$F(R,q,z) = \min_{0 \le L \le 1-R} \left\{ \left(\tfrac{R+L}{2}\right) \log_2(z),\ (R+L)\log_2(z) - L\log_2(q) \right\}.$$

In Figure 1, we give the cost of Stern's algorithm for random R-SDP instances, where we choose $W = W^*$, i.e., the maximal weight that guarantees uniqueness. Note that the cost at the point $z = q-1$ corresponds to the cost of Stern on a random SDP instance and thus, we can see that R-SDP with $z < q-1$ has a much larger cost than the SDP with the same parameters q, n, R.

The security of R-SDP highly depends on the exact shape of $\mathbb{E}$. There are, indeed, several choices that lead to a somewhat easier problem. For instance, one can choose an extension field $\mathbb{F}_{p^m}$, for some prime p and integer m and $\mathbb{E} \subset \mathbb{F}_{p^m}^*$. In this case, several choices of $\mathbb{E}$ lead to an easier problem, e.g. $\mathbb{E} = \mathbb{F}_p^*$. To avoid this possibility, we directly restrict our considerations to prime fields.

Algorithm 1: Restricted Stern/Dumer algorithm

Input $\mathbf{H} \in \mathbb{F}_q^{r\times n}, \mathbf{s} \in \mathbb{F}_q^r, v < w \in \mathbb{N}$.
Output $\mathbf{e} \in \mathbb{E}_0^n$ such that $\mathrm{wt}(\mathbf{e}) = w$, $\mathbf{s} = \mathbf{e}\mathbf{H}^\top$.

1: Choose a set $J \subset \{1, \dots, n\}$ of size $k + \ell$.
2: Find $\mathbf{U} \in \mathbb{F}_q^{r\times r}$ such that $(\mathbf{U}\mathbf{H})_{J^C} = \begin{pmatrix} \mathbf{I}_{r-\ell} \\ \mathbf{0} \end{pmatrix}$ and $(\mathbf{U}\mathbf{H})_J = \begin{pmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{pmatrix}$.
3: Compute $\mathbf{s}\mathbf{U}^\top = \begin{pmatrix} \mathbf{s}_1 & \mathbf{s}_2 \end{pmatrix}$, where $\mathbf{s}_1$ is of size $r - \ell$ and $\mathbf{s}_2$ is of size ℓ.
4: Set $\mathbf{H}_2 = \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{pmatrix}$, for $\mathbf{A}_i$ of size $\ell \times (k + \ell)/2$.
5: Set $\mathscr{L}_1 = \{(\mathbf{x}_1, \mathbf{x}_1\mathbf{A}_1^\top) \mid \mathbf{x}_1 \in \mathbb{E}_0^{(k+\ell)/2}, \mathrm{wt}(\mathbf{x}_1) = v/2\}$.
6: Set $\mathscr{L}_2 = \{(\mathbf{x}_2, \mathbf{s}_1 - \mathbf{x}_2\mathbf{A}_2^\top) \mid \mathbf{x}_2 \in \mathbb{E}_0^{(k+\ell)/2}, \mathrm{wt}(\mathbf{x}_2) = v/2\}$.
7: **for** $((\mathbf{x}_1, \mathbf{a}), (\mathbf{x}_2, \mathbf{a})) \in \mathscr{L}_1 \times \mathscr{L}_2$ **do**
  if $wt(\mathbf{s}_1 - (\mathbf{x}_1, \mathbf{x}_2)\mathbf{H}_1^\top) = w - v$ **then**
    Return $\mathbf{e}_J = (\mathbf{x}_1, \mathbf{x}_2)$ and $\mathbf{e}_{J^C} = \mathbf{s}_1 - (\mathbf{x}_1, \mathbf{x}_2)\mathbf{H}_1^\top$.
8: Start over with Step 1 and a new selection of J.

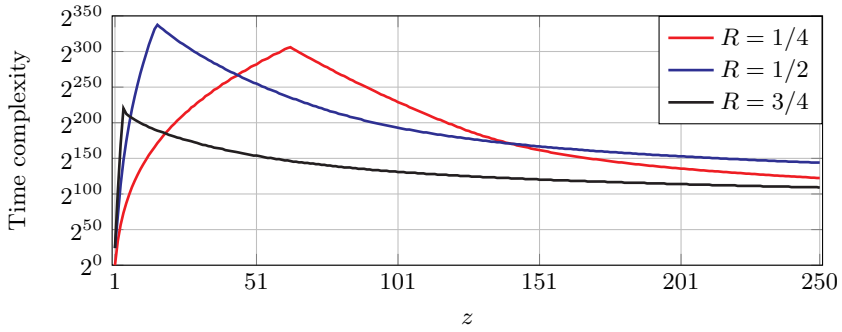

Fig. 1: Cost of Stern's algorithm for random R-SDP instances with $q = 251$, $n = 256$, $W = W^*$ and several rates R.

As another suboptimal choice, one can choose rather large values for q and $\mathbb{E}_0 = \{0, 1\}$. Thus, solvers for subset sum problems may be used [14], where one adds some elements to the search space.

To circumvent possible speedups from such techniques, we restrict ourselves to particular choices of error sets $\mathbb{E}$ of relatively large size. For more details on safe choices of $\mathbb{E}$ and attacks using [14], we refer to Appendix B.

Let us also quickly comment on other solvers for R-SDP. Note that statistical decoding for SDP [19, 25, 34] is based on the bias towards 0 of $\langle \mathbf{e}, \mathbf{h} \rangle$ for sparse vectors $\mathbf{e}$ and $\mathbf{h}$. For R-SDP with full weight vectors, the sought error vector $\mathbf{e}$ is not sparse, and the multiplicative structure of $\mathbb{E}$ is lost in the additions of the inner product. On the other hand, algebraic attacks that exploit the small order of the entries of $\mathbf{e}$ cannot be mounted straightforwardly, as the multiplicative structure of $\mathbb{E}$ is incompatible with the additive linearity of the

syndrome computation. Also, Gröbner bases attacks do not give any speed up over the considered ISD algorithms as observed in [10].

4 Building ZK Protocols from the R-SDP: a Preliminary Analysis

This section describes how standard approaches to building ZK protocols can be converted to use R-SDP.

The protocols we consider achieve zero knowledge thanks to the following fundamental property: there must be a set X and a set of maps $\mathscr{T}$ that act transitively on X, with the property that

$$\forall x \in X,\ \sigma(x) \text{ is uniformly distributed over } X \text{ when } \sigma \xleftarrow{\$} \mathscr{T}. \tag{3}$$

Consequently, when $x \in X$ is the secret key, revealing $y = \sigma(x)$ without revealing σ leaks no information about x.

For schemes based on the SDP, Property (3) is satisfied by choosing X as the Hamming sphere with some radius w and $\mathscr{T}$ as the set of linear isometries, i.e., the set of monomial transformations. A monomial transformation can be described as $(\pi, \mathbf{v})$ with $\pi \in S_n$ a permutation and $\mathbf{v} \in (\mathbb{F}_q^*)^n$. The action of $\sigma = (\pi, \mathbf{v})$ on a vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}_q^n$ corresponds to

$$\sigma(\mathbf{a}) = \left(v_1 a_{\pi^{-1}(1)}, \ldots, v_n a_{\pi^{-1}(n)}\right) = \pi(\mathbf{a}) \star \mathbf{v},$$

where $\star$ denotes component-wise multiplication.

4.1 Zero Knowledge Masking of Restricted Vectors

To use R-SDP, we will make use of the following choices. First, we set

$$\mathbb{E} = \left\{ g^j \mid j \in \{0, 1, \ldots, z-1\} \right\},$$

where $g \in \mathbb{F}_q^*$ has multiplicative order $z < q - 1$. In other words, we choose $\mathbb{E}$ as the cyclic subgroup of $\mathbb{F}_q^*$ which is generated by g and, to have $\mathbb{E} \neq \mathbb{F}_q^*$, we require that g is not primitive. Then, we set $X \coloneqq \mathcal{S}_w^{\mathbb{E}}$, i.e., choose the secret key as an element of the restricted Hamming sphere with radius w. Also, we set $\mathscr{T} \coloneqq S_n \times \mathbb{E}^n$, which contains only the monomial transformations having restricted scaling coefficients. It is easy to see that, with these choices, Property (3) holds. Notice that the action of any $\sigma \coloneqq (\pi, \mathbf{v}) \in \mathscr{T}$ is given by

$$\sigma(\mathbf{a}) = \pi(\mathbf{a}) \star \mathbf{v} = \pi(\mathbf{a}) \star \left(g^{i_1}, \ldots, g^{i_n}\right), \tag{4}$$

with $(i_1, \ldots, i_n) \in \mathbb{Z}_z^n$. We refer to $\mathbb{E}^n$ as *restricted group* and to $\mathscr{T}$ as the group of *restricted maps*. Notice that $(\mathbb{E}^n, \star)$ is an abelian group: we investigate its properties in Section 5.

We observe that, when $w = n$, i.e., we have full weight, we can choose a simpler description for $\mathscr{T}$. Indeed, we have $\mathcal{S}_w^{\mathbb{E}} = \mathbb{E}^n$ and $\mathscr{T} \coloneqq \mathbb{E}^n$, that is, the restricted maps can be represented by restricted full-weight vectors. In fact, for any $\mathbf{e}, \mathbf{e}' \in \mathbb{E}^n$, there exists a unique $\sigma \in \mathscr{T}$ such that $\mathbf{e}' = \sigma(\mathbf{e})$. Moreover, $\sigma \coloneqq \mathbf{v}$ for some $\mathbf{v} \in \mathbb{E}^n$ and $\sigma(\mathbf{e}) = \mathbf{e} \star \mathbf{v}$. More interesting properties of this setting can be found in Section 5. For the moment, it is sufficient to anticipate that this choice is the one that will yield the best performances for ZK protocols.

4.2 The Case Study of CVE with R-SDP

The CVE protocol [20] has been, historically, the first ZK protocol based on non-binary SDP with low Hamming weight. It has been derived from the famous protocol by Stern [41] and Shamir's permuted kernel protocol [39]. Modern solutions, such as [31], are built on CVE. Hence, it makes sense to start by adapting this protocol to the R-SDP setting as a preparatory step. This shows that the most common techniques to build a ZK protocol in the SDP setting also hold for the R-SDP setting. The CVE based on R-SDP is depicted in Figure 2.

Private Key $\quad \mathbf{e} \in \mathcal{S}_w^{\mathbb{E}}$
Public Key $\quad \mathbb{E}, w, \mathbf{H} \in \mathbb{F}_q^{r \times n}, \mathbf{s} = \mathbf{e}\mathbf{H}^\top \in \mathbb{F}_q^r$

PROVER		VERIFIER
Choose $\mathbf{u} \xleftarrow{\$} \mathbb{F}_q^n$, $\sigma \xleftarrow{\$} \mathscr{T}$		
Set $c_0 = \mathsf{Hash}\big(\sigma, \mathbf{u}\mathbf{H}^\top\big)$		
Set $c_1 = \mathsf{Hash}\big(\sigma(\mathbf{u}), \sigma(\mathbf{e})\big)$	$\xrightarrow{\mathsf{Com}=(c_0,c_1)}$	
	$\xleftarrow{\beta}$	Choose $\beta \xleftarrow{\$} \mathbb{F}_q^*$
Set $\mathbf{y} = \sigma(\mathbf{u} + \beta\mathbf{e})$		
	$\xrightarrow{\mathbf{y}}$	
	$\xleftarrow{b}$	Choose $b \xleftarrow{\$} \{0, 1\}$
If $b = 0$, set $\mathsf{Rsp} \coloneqq \sigma$		
If $b = 1$, set $\mathsf{Rsp} \coloneqq \mathbf{e}' = \sigma(\mathbf{e})$	$\xrightarrow{\mathsf{Rsp}}$	If $b = 0$, accept if: $c_0 = \mathsf{Hash}\big(\sigma, \sigma^{-1}(\mathbf{y})\mathbf{H}^\top - \beta\mathbf{s}\big)$ If $b = 1$, accept if: $\mathbf{e}' \in \mathcal{S}_w^{\mathbb{E}}$ and $c_1 = \mathsf{Hash}\big(\mathbf{y} - \beta\mathbf{e}', \mathbf{e}'\big)$

Fig. 2: R-CVE: CVE scheme based on R-SDP

It is easy to see that, as the original CVE scheme, also the R-CVE protocol achieves ZK. Indeed, $\mathbf{u}$ is chosen uniformly at random in $\mathbb{F}_q^n$ and the same holds for $\mathbf{e}' = \sigma(\mathbf{e})$, thanks to Property (3). Also, the soundness error remains the same as in CVE, that is, $\varepsilon = \frac{q}{2(q-1)}$, and an adversary achieving a larger success probability is either able to solve R-SDP or find hash collisions. A rigorous proof of this fact would be identical to the one in [20] and is hence omitted. Finally, using the Fiat-Shamir transform on this $q2$-identification scheme, due to [32] we get EUF-CMA security.

We consider two possible choices for R-SDP.

Table 1: Comparison between communication costs for SDP and R-SDP, for the case of $\lambda = 128$ and $R \approx 0.5$; all sizes are expressed in bytes.

	q	z	n	k	w	Size(σ)	Size($\mathbf{e}'$)	Size($\mathbf{y}$)
SDP	512	511	196	92	84	372.4	118.1	220.0
	256	255	207	93	90	369.3	115.0	207.0
	128	127	220	101	90	367.3	105.0	192.5
R-SDP I	677	26	84	42	73	102.0	48.5	105.0
	379	21	103	52	82	124.8	54.0	108.1
	197	14	103	51	91	117.1	49.6	103.0
R-SDP II	2017	63	70	32	70	52.5	52.5	96.3
	1021	30	79	40	79	49.4	49.4	98.8
	197	14	102	51	102	51.0	51.0	102.0

Choice I Values of z such that $W^* < 1$. We set $w = W^* n < n$ and $\mathscr{T} \coloneqq S_n \times \mathbb{E}^n$. Representing σ and $\mathbf{e}'$ requires

$$\begin{cases} \text{Size}(\sigma) = \lceil \log_2(n!) + n \log_2(z) \rceil, \\ \text{Size}(\mathbf{e}') = \lceil \log_2 \left(\binom{n}{w} \right) + w \log_2(z) \rceil. \end{cases} \tag{Choice I}$$

The sizes are derived considering that σ and $\mathbf{e}'$ are uniformly distributed in two sets with sizes $|\mathscr{T}| = |S_n| \cdot |\mathbb{E}^n| = n! \cdot z^n$ and $|\mathcal{S}_w^{\mathbb{E}}| = \binom{n}{w} z^w$.

Choice II Values of z for which (2) returns $W^* = 1$. Remember that, asymptotically, this is guaranteed when $z \leq q^{1-R}$. In this case, we can choose $\mathscr{T} \coloneqq \mathbb{E}^n$, and consequently have

$$\text{Size}(\sigma) = \text{Size}(\mathbf{e}') = \lceil n \log_2(z) \rceil. \tag{Choice II}$$

When SDP is used, we instead have

$$\begin{cases} \text{Size}(\sigma) = \lceil \log_2(n!) \rceil + n \log_2(q-1) \rceil, \\ \text{Size}(\mathbf{e}') = \lceil \log_2 \left(\binom{n}{w} \right) + w \log_2(q-1) \rceil. \end{cases} \tag{SDP}$$

In Table 1, we have compared how the above sizes behave when targeting a security level of $\lambda = 128$ bits. For SDP, we have used the parameters which are recommended in [31], while for R-SDP we have designed some instances taking into account the attacks described in Appendix B and the attack of [35] (see Equation (9)). Table 1 shows that R-SDP yields much smaller sizes than SDP for the same security level. In particular, Choice II seems to be better suited for ZK protocols. Indeed, we are able to completely avoid the use of permutations and thus reduce the original cost of sending a map to $\lceil n \log_2(z) \rceil$.

We would like to point out that, in modern protocols, several techniques can be applied to reduce the communication cost, the simplest one being sending generating seeds instead of random objects. However, almost every scheme (apart

from those based on MPCitH) makes use, at some point, of messages containing the objects we have considered in Table 1. As the sizes of these objects are significantly smaller for R-SDP, the problem is very promising.

For the remainder of the paper we will only focus on R-SDP Choice II, i.e., we will consider R-SDP with maximum Hamming weight $w = n$, since this allows for greater reductions.

Remark 2. For SDP and R-SDP Case I, we have considered optimal (i.e., as small as possible) sizes for σ and $\mathbf{e}'$. Notice that, to achieve such sizes, one should use encoding/decoding schemes (e.g., the Lehmer code) which require rather involved operations. In certain applications, these schemes may not be applicable, as the resulting protocol would become too slow: in such cases, the sizes for SDP and R-SDP Case I would get larger than those in Table 1. For instance, the standard encoding for permutations is through a list of n integers in the range $[0; n-1]$, thus taking $n \log_2(n)$ bits (instead of the optimal $\log_2(n!) \approx n \log_2(n/e)$ bits).

4.3 MPCitH based on R-SDP

To reduce the soundness error of a ZK protocol, one can use a $(N-1)$-private MPC. For SDP, the MPCitH paradigm has been first employed in the SDItH scheme [29], and then improved in [4]. In Appendix C we briefly discuss how existing MPCitH schemes may be adapted to use R-SDP. In particular, we argue that the PKP protocol in [27, Section 6] seems to be the best choice. Since PKP is essentially a decoding problem, this is not surprising.

As we show later, we will also adapt the BG protocol, introduced in [18, Figure 3] and based on PKP, to the use of R-SDP. We postpone the presentation of the resulting protocol to Section 6, and continue describing how R-SDP can be made even more powerful with an ad-hoc choice for the set of restricted vectors.

5 R-SDP(G): Using Subgroups of the Restricted Group

In this section, we present a generalization of R-SDP, which allows to represent objects in an even more compact way. The idea consists of identifying a set of restricted maps that i) has small cardinality (but not too small, since this may facilitate attacks), and ii) admits a compact and easy-to-compute representation. We extend this reasoning to restricted vectors, and in the end obtain that, for a security level of λ bits, we can represent any restricted object with $(1+\alpha)\lambda$ bits, with α being a small positive constant. Since we are reducing the space from which secret keys and ephemeral objects are sampled, security issues may arise. Yet, with coding theory arguments, we argue that incorporating this information into existing attacks does not lead to significant speed-ups.

5.1 Properties of the Restricted Group

Let us make some observations on the properties of $\mathbb{E}^n$, seen as a group.

Recall that $\mathbb{E} = \{g^i \mid i \in \{0, \dots, z-1\}\}$ is the cyclic subgroup of $\mathbb{F}_q^*$ generated by g, with order z. We focus on the case $n = w$, so that restricted vectors are given by $\mathbf{e} = (g^{i_1}, \dots, g^{i_n}) \in \mathbb{E}^n$, for $i_j \in \{0, \dots, z-1\}$.

Also this particular case of R-SDP is still NP-hard, see Appendix A.

As we have already shown in the previous section, in such a setting also the restricted maps can be represented by restricted vectors. In fact, any map sending $\mathbf{e} = (g^{i_1}, \dots, g^{i_n})$ to $\mathbf{e}' = (g^{j_1}, \dots, g^{j_n})$ is simply given by componentwise multiplication with $(g^{j_1 - i_1}, \dots, g^{j_n - i_n})$. Indeed, one can simply check that

$$\sigma(\mathbf{e}) = (g^{j_1 - i_1}, \dots, g^{j_n - i_n}) \star (g^{i_1}, \dots, g^{i_n}) = (g^{j_1}, \dots, g^{j_n}) = \mathbf{e}'.$$

We thus use restricted maps σ in $\mathbb{E}^n$ and write $\sigma(\mathbf{e})$ to mean $\sigma \star \mathbf{e}$. Notice that $\mathbb{E}^n$ acts transitively and freely on itself: for any pair $\mathbf{e}, \mathbf{e}' \in \mathbb{E}^n$, there is a unique map $\sigma \in \mathbb{E}^n$ such that $\mathbf{e}' = \sigma(\mathbf{e})$.

There exists a natural bijection $\ell : \mathbb{E}^n \to \mathbb{Z}_z^n$, which allows for a compact representation of the restricted vectors in $\mathbb{E}^n$, as

$$\ell\big((g^{i_1}, \dots, g^{i_n})\big) = (i_1, \dots, i_n).$$

It is easy to see that $(\mathbb{E}^n, \star)$ is isomorphic to $(\mathbb{Z}_z^n, +)$, and that both groups are abelian. This also allows for a more efficient computation of $\sigma(\mathbf{e})$. Indeed, if $\sigma = (g^{s_1}, \dots, g^{s_n}) \in \mathbb{E}^n$ and $\mathbf{e} = (g^{i_1}, \dots, g^{i_n}) \in \mathbb{E}^n$, instead of computing

$$\sigma(\mathbf{e}) = (g^{s_1}, \dots, g^{s_n}) \star (g^{i_1}, \dots, g^{i_n}) = (g^{s_1 + i_1}, \dots, g^{s_n + i_n}),$$

one can simply add the two exponents over $\mathbb{Z}_z$ as

$$\ell(\sigma) + \ell(\mathbf{e}) = (s_1, \dots, s_n) + (i_1, \dots, i_n) = \ell(\sigma(\mathbf{e})).$$

Any restricted vector $\mathbf{e} \in \mathbb{E}^n$ generates a cyclic subgroup $\{\mathbf{e}^i \mid i \in \mathbb{N}\} < \mathbb{E}^n$. Due to the isomorphism to $\mathbb{Z}_z^n$, the order of $\mathbf{e}$ is the same as the order of $\ell(\mathbf{e})$ in $(\mathbb{Z}_z^n, +)$. Recall that $x \in \mathbb{Z}_z$ has order $\frac{z}{\gcd(x,z)}$. Thus, for $\mathbf{e} = (g^{i_1}, \dots, g^{i_n})$, we have that

$$\operatorname{ord}(\mathbf{e}) = \operatorname{lcm}\big(\operatorname{ord}(i_1), \dots, \operatorname{ord}(i_n)\big) = \operatorname{lcm}\left(\tfrac{z}{\gcd(i_1,z)}, \dots, \tfrac{z}{\gcd(i_n,z)}\right),$$

where lcm denotes the least common multiple.

One can easily construct a restricted vector $\mathbf{e}$ with maximum order z, e.g., by taking one of the i_j which is coprime to z.

5.2 Cyclic Subgroups of the Restricted Group

We now consider the subgroup G of $(\mathbb{E}^n, \star)$ which is generated by m many restricted vectors $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{E}^n$. In other words,

$$G = \langle \mathbf{x}_1, \dots, \mathbf{x}_m \rangle = \{\, \mathbf{x}_1^{u_1} \star \cdots \star \mathbf{x}_m^{u_m} \,|\, u_j \in \{0, \dots, z-1\}\}.$$

In the following, we will call G the *restricted subgroup*. To any $\mathbf{a} \in G$, we can associate a vector representation through $\ell_G : G \to \mathbb{Z}_z^m$, as follows

$$\ell_G\left(\mathbf{x}_1^{u_1} \star \cdots \star \mathbf{x}_m^{u_m}\right) = (u_1, \ldots, u_m). \tag{5}$$

Clearly, $(G, \star)$ is a subgroup of $(\mathbb{E}^n, \star)$ and ℓ_G is a group homomorphism. Thus, for any $\mathbf{a}, \mathbf{b} \in G$, we have

$$\ell_G(\mathbf{a} \star \mathbf{b}) = \ell_G(\mathbf{a}) + \ell_G(\mathbf{b}) \mod z.$$

Note that a priori the elements of G do not have a unique representation in $\mathbb{Z}_z^m$. We later give a condition to have a unique $\ell_G(\mathbf{a}) \in \mathbb{Z}_z^m$.

In the following proposition we show the connection between ℓ and ℓ_G.

Proposition 2. *Let $\mathbf{M}_G \in \mathbb{Z}_z^{m \times n}$ be the matrix whose j-th row is $\ell(\mathbf{x}_j)$, and $\mathscr{B} = \{\mathbf{u}\mathbf{M}_G \mid \mathbf{u} \in \mathbb{Z}_z^m\}$. Then, $\ell(\mathbf{a}) = \ell_G(\mathbf{a})\mathbf{M}_G \mod z$, for any $\mathbf{a} \in G$ and $|\mathscr{B}| = |G|$.*

Proof. Let $\mathbf{x}_j = \left(g^{i_1^{(j)}}, \ldots, g^{i_n^{(j)}}\right)$, hence $\ell(\mathbf{x}_j) = \left(i_1^{(j)}, \ldots, i_n^{(j)}\right)$, and $\mathbf{a} \in G$. Then, it holds that

$$\mathbf{a} = \star_{j=1}^m \left(g^{u_j i_1^{(j)}}, \ldots, g^{u_j i_n^{(j)}}\right) = \left(g^{\sum_{j=1}^m u_j i_1^{(j)}}, \ldots, g^{\sum_{j=1}^m u_j i_n^{(j)}}\right).$$

By construction, the element in the j-th row and v-th column of $\mathbf{M}_G$ is $i_v^{(j)}$. Hence, for $\mathbf{u} = \ell_G(\mathbf{a}) = (u_1, \ldots, u_m) \in \mathbb{Z}_z^m$ we get

$$\ell(\mathbf{a}) = \left(\sum_{j=1}^m u_j i_1^{(j)}, \ldots, \sum_{j=1}^m u_j i_n^{(j)}\right) = \mathbf{u}\mathbf{M}_G \in \mathbb{Z}_z^n.$$

The second claim follows, since $\ell : \mathbb{E}^n \mapsto \mathbb{Z}_z^n$ is a bijection. □

Remark 3. When z is a prime number, we can easily construct a G of maximal order z^m, by taking $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{E}^n$ such that $\{\ell(\mathbf{x}_1), \ldots, \ell(\mathbf{x}_m)\}$ are linearly independent. This is equivalent to asking for a full-rank matrix $\mathbf{M}_G$.

Note that $\mathbf{M}_G$ acts like the generator matrix of G and for $\ell(\mathbf{a}) = \mathbf{u}\mathbf{M}_G$ we have that $\mathbf{u} = \ell_G(\mathbf{a})$ is the information vector. Thus, if we require z to be prime and $\mathbf{M}_G \in \mathbb{F}_z^{m \times n}$ to have full rank m, then for any $\mathbf{a} \in G$ we have a unique $\ell_G(\mathbf{a}) = \mathbf{u}$, which is such that $\mathbf{u}\mathbf{M}_G = \ell(\mathbf{a}) \in \mathbb{F}_z^n$. We thus from now on focus on prime order z.

To summarize, in order to represent a vector $\mathbf{e} \in G$, respectively a transformation $\sigma \in G$, it is enough to use $\ell_G(\mathbf{e})$, respectively $\ell_G(\sigma)$. Given the matrix $\mathbf{M}_G$, containing all the $\ell(\mathbf{x}_i)$ of the generators, the $\ell_G(\mathbf{e})$ is the length-m vector of coefficients in $\mathbb{F}_z$ required to generate $\ell(\mathbf{e}) \in \mathbb{F}_z^n$.

This will also have an impact on the sizes of restricted vectors and restricted transformations, as now elements of G are represented with an element of $\mathbb{Z}_z^m$, of size $m\lceil \log_2(z) \rceil$.

5.3 Solving R-SDP with Restricted Subgroup

In this section, we focus only on restrictions $\mathbb{E} = \{g^i \mid i \in \{0, \ldots, z-1\}\}$ such that z is prime, $w = n$ and restricted subgroups with $|G| = z^m$. We now introduce R-SDP with the additional constraint that the solution is an element of G.

Problem 3 R-SDP(G): SDP with Restricted Diagonal Subgroup G
Let $G = \langle \mathbf{x}_1, \ldots, \mathbf{x}_m \rangle$, $\mathbf{H} \in \mathbb{F}_q^{r \times n}$ and $\mathbf{s} \in \mathbb{F}_q^r$. Does there exist a vector $\mathbf{e} \in G$ with $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$?

Note that as R-SDP(G) includes R-SDP as a special case (for $G = \mathbb{E}^n$) also R-SDP(G) is NP-complete. Since $|G| = z^m$, the criterion to have (on average) a unique solution gets modified as follows

$$m \log_2(z) \leq (1 - R) n \log_2(q).$$

Since the subgroup G possesses some additional structure and introduces a smaller solution space, we argue in the remainder of this section on how to choose parameters properly, such that attacks exploiting this structure can be ruled out. First, it is obvious that G must have a sufficiently large order, i.e., $|G| \geq 2^\lambda$. Indeed, if $|G|$ is too small, then R-SDP(G) can be solved with a trivial brute-force attack over G, taking time $O\big(|G|\big)$.

On the other hand, one can also disregard G, find all candidate solutions $\mathbf{e} \in \mathbb{E}^n$ and check their validity, i.e., whether $\mathbf{e} \in G$, afterwards. Such attacks can be thwarted by choosing instances which have more than 2^λ solutions in $\mathbb{E}^n$. In this case, checking for all the candidate solutions whether $\mathbf{e} \in G$ guarantees the security level. Notice that this choice is rather conservative: we are neglecting the cost of actually finding these solutions, using the exponential solvers, e.g. restricted Stern/Dumer. It follows that any efficient solver for R-SDP(G) has to directly take G into account. Since this is not possible for the algorithm presented in Appendix B, we only consider a solver based on restricted Stern in the following.

ISD attacks for R-SDP(G) An improved collision search requires a method to enumerate parts of the solution vector with length $t \geq k/2$ in time smaller than z^t (since else one could just enumerate all $\mathbf{e}_T \in \mathbb{E}^t$). This can be done with the following procedure, which starts from a set $T \subseteq \{1, \ldots, n\}$ of size t and returns all candidates for $\mathbf{e}_T$:

1) set $\mathbf{M}' \in \mathbb{F}_z^{m \times t}$ consisting of the columns of $\mathbf{M}_G$ indexed by T;
2) enumerate all length-t vectors which can be obtained as linear combinations of the rows of $\mathbf{M}'$;
3) use any such vector as exponents for a candidate $\mathbf{e}_T$. To do this, one first enumerates $\mathscr{B}' = \{\mathbf{u}\mathbf{M}' \mid \mathbf{u} \in \mathbb{F}_z^m\} \subseteq \mathbb{F}_z^t$. Then, to each $\mathbf{x} \in \mathscr{B}'$, associates a candidate $\mathbf{e}_T = \ell^{-1}(\mathbf{x})$.

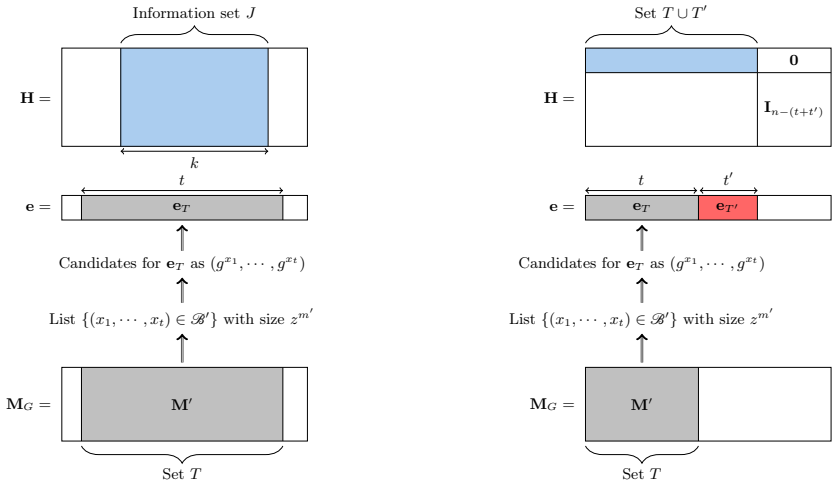

Fig. 3: Strategies to speed-up ISD with the knowledge about G.

With the above approach, one can enumerate all candidates for $\mathbf{e}_T$ in time $O(|\mathscr{B}'|) = O\big(z^{m'}\big)$, where $m' = \mathrm{rank}(\mathbf{M}') \leq \min\{m, t\}$. If $m' = t$, we have that enumeration takes time $O(|\mathscr{B}'|) = O(z^t)$, and as we already chose $z^t > 2^\lambda$, we can ignore this case. Consequently, an attacker can only hope for a speed-up if $m' < t$.

The problem of finding a set T with the desired properties can be stated as follows.

Problem 4 (Submatrix Rank Problem) *Let $\mathbf{M}_G \in \mathbb{F}_z^{m\times n}$, with $m < n$ and $m' \leq m$. Is there a set $T \subset \{1,\ldots,n\}$ of size t, such that $rk((\mathbf{M}_G)_T) = m'$?*

Assuming that one is able to find a set T such that $\mathbf{M}' := (\mathbf{M}_G)_T$ has rank $m' < t$, one can possibly speed-up ISD algorithms:

- if $t > k$, then T contains[5] an information set $J \subseteq T$. So, we can enumerate all candidates for $\mathbf{e}_J$ in time $z^{m'}$. If m' is particularly low (say, lower than $\lambda \log_z(2)$) the attack can use a single list of size $z^{m'}$ in which we put candidates for $\mathbf{e}_J$. See Figure 3a for a representation of this strategy;
- if $t < k$, then we can use the $z^{m'}$ candidates for $\mathbf{e}_T$ to build one of the lists for Stern's algorithm. However, we also require an enumeration of all $\mathbf{e}_{T'}$, with T' disjoint from T, of size t', such that $t + t' \geq k$. Thus, a collision search leads to a cost of $\max\left\{z^{m'}, z^{t'}, z^{m+m'}q^{k-(t+t')}\right\}$. So, this approach can be convenient only if $t \geq k/2$. See Figure 3b for a representation of this strategy;

[5] Unless all $k\times t$ matrices are singular, however, a random $k\times t$ matrix has probability $\prod_{i=0}^{k-1}(1-q^{i-t}) \geq (1-q^{-(t-k+1)})^k$ to be invertible.

- assume that one is able to find several sets T of size t such that $m' < t$. Then, we can enumerate several portions of the solution $\mathbf{e}$, of size t, in time $z^{m'}$. We can use them to build several lists, which we can combine with a collision search approach with more than one level. Again, there is no guarantee that this yields an attack with overall cost $z^{m'}$ since we also need to consider how list sizes grow after merging.

5.4 Criteria to Design R-SDP(G)

We will adopt the following very conservative criterion to completely cut out all the above possibilities.

Requirement 1 *Let* $rk((\boldsymbol{M}_G)_T) = m'$. *For any* $t \in \{1, \ldots, n\}$ *and any set* $T \subseteq \{1, \ldots, n\}$ *of size* t, *we want that* $m' \geq \min\left\{t, \frac{\lambda}{\log_2(z)}\right\}$.

In the case of full rank, one cannot improve over enumerating all errors in the space. Regarding rank deficiency, Requirement 1 ensures that the enumeration of possible error vectors exceeds the security level.

This implies that any strategy that exploits the structure of G to speed up ISD attacks will not be more efficient than generic ISD attacks and we choose our instances such that these have a cost of at least 2^{λ}.

We now provide strong evidence that Requirement 1 is rather conservative. First, we show that Problem 4 is NP-hard. This implies that, even if some set T with the desired properties exists, finding it is hard. The NP-hardness proof will make use of the following result.

Theorem 1. Relation between m' and subcodes of $\mathscr{B}$
Let $\mathbf{M}_G \in \mathbb{F}_z^{m \times n}$ *and* $\mathscr{B} = \langle \mathbf{M}_G \rangle \subseteq \mathbb{F}_z^n$ *be a linear code of dimension* m. *Then, there exists* $T \subseteq \{1, \ldots, n\}$ *of size* t, *such that* $m' = rk((\mathbf{M}_G)_T)$, *if and only if*

i) $m' \leq t \leq m$, *then* $\mathscr{B}^{\perp}$ *contains a subcode with dimension* $t - m'$ *and support size* $\leq t$;
ii) $m' \leq m \leq t$, *then* $\mathscr{B}$ *contains a subcode with dimension* $m - m'$ *and support size* $\leq n - t$.

Proof. We start with the case $m' \leq t \leq m$. Since $\mathbf{M}' = (\mathbf{M}_G)_T$ has m rows and $t \leq m$ columns, if its rank is lower than t this implies that there exist $k' = t - m'$ linearly independent vectors $\mathbf{x}_1, \ldots, \mathbf{x}_{k'} \in \mathbb{F}_z^t$ such that $\mathbf{M}'\mathbf{x}_i^{\top} = \mathbf{0}$. We can use such vectors to define a generator matrix $\mathbf{X} \in \mathbb{F}_z^{k' \times t}$ for the right kernel of $\mathbf{M}'$. Now, let $\mathbf{C} \in \mathbb{F}_q^{k' \times n}$ be a matrix such that $\mathbf{C}_T = \mathbf{X}$ and $\mathbf{C}_{T^C} = \mathbf{0} \in \mathbb{F}_z^{k' \times (n-t)}$. By construction, it holds that $\mathbf{C}$ has rank k' and is such that $\mathbf{M}_G\mathbf{C}^{\top} = \mathbf{0}$, so $\mathbf{C}$ is a generator matrix for a k'-dimensional subcode of $\mathscr{B}^{\perp}$. Since $\mathbf{C}$ has at least $n - t$ null columns (the ones indexed by T^C), we know that $\mathbf{C}$ generates a code with dimension k' and support size not greater than t. For the proof of the other direction, one can proceed in the same way. If there is a subcode of $\mathscr{B}^{\perp}$ of dimension $t - m'$ and support size $\leq t$, we can find a generator matrix $\mathbf{C}$, which has (at least) $n - t$ zero columns and denote these indices by T^C.

The case of $m' \leq m \leq t$ is treated analogously, with the only difference that we need to focus on the left kernel of $\mathbf{M}'$. □

Theorem 2. *The Submatrix Rank Problem is NP-complete.*

Proof. We present a reduction from the low weight codeword finding problem, which is NP-complete [16]: given $d \in \mathbb{N}$ and $\mathbf{G} \in \mathbb{F}_z^{k\times n}$, are there codewords in $\mathscr{C} = \langle \mathbf{G} \rangle$ with weight $\leq d$? Due to the Singleton bound, we focus on $d < n-k+1$. We show that any instance $\{\mathbf{G}, d\}$ can be transformed, in polynomial time, into an instance of the Submatrix Rank Problem. We will denote by Solve an algorithm that, on input a matrix $\mathbf{M}_G \in \mathbb{F}_z^{m\times n}$ and two integers $t, m' \in \mathbb{N}$, returns "yes" if $T \subseteq \{1, \ldots, n\}$ of size t and $m' = \mathrm{rk}((\mathbf{M}_G)_T)$ exists, and "no" otherwise. We can set $\mathbf{M}_G \coloneqq \mathbf{G}$ and $t \coloneqq n - d$. Notice that $m \coloneqq k$ and $t > n - (n - k + 1) = m - 1$. We run Solve for all $m' \leq m - 1$.

- Assume that the call for $m^\star$ on Solve returns a "yes". Since $t \geq m$, we apply ii) of Theorem 1 and learn that $\mathscr{C} = \langle \mathbf{G} \rangle$ has a subcode $\mathscr{C}'$ of dimension $m - m^\star$ with support size $s \leq n - t = d$. Since $d(\mathscr{C}) \leq d(\mathscr{C}') \leq s \leq d$, we return "yes" for the original problem.
- Assume that none of the calls on Solve return a "yes", then all subcodes have a support size greater than $t - n = d$. Notice that we also tried $m' = m - 1$, so Solve has also considered existence of subcodes of dimension $m - m' = 1$, that is, codewords. So, we return "no" for the original problem.

□

Notice that, as a consequence of Theorem 1, finding sets T with the desired properties implies finding subcodes with small supports. This can be done using ISD, with a time complexity that (more or less) grows exponentially with the desired support size. Thus, finding a set T is also inefficient. However, we describe how to choose the value of m so that such useful subcodes are not expected to exist. For a random code with length n and dimension k, over $\mathbb{F}_z$, the average number of subcodes with dimension k' and support size w is well estimated by [38, Theorem 1]

$$N_k(k', w) = \binom{n}{w} (z^{k'} - 1)^{w-k'} \begin{bmatrix} k \\ k' \end{bmatrix}_z \begin{bmatrix} n \\ k' \end{bmatrix}_z^{-1} . \tag{6}$$

Since for $\mathbf{M}_G$ we do not impose any structure, apart from the full rank property, we can safely study its row space $\mathscr{B}$ as a random code with dimension m. Analogously, we can treat its dual $\mathscr{B}^\perp$ as a random code, with dimension $n - m$. So, we can update Requirement 1 as follows.

Requirement 2 *We set $m > \lambda \log_z(2)$ as the minimum integer such that*

- *for any $m' \leq t \leq m$ with $\sum_{i=1}^{t} N_{n-m}(t - m', i) < 1$, we have $m' > \lambda \log_z(2)$;*
- *for any $m' \leq m < t$ with $\sum_{i=1}^{n-t} N_m(m - m', i) < 1$, we have $m' > \lambda \log_z(2)$.*

Thus, even if such subcodes exist, the enumeration cost $z^{m'}$ exceeds the security level.

5.5 R-SDP(G) in Practice: Easy to Implement and Tight Parameters

Let us first briefly comment on some implementation aspects for R-SDP(G).

When R-SDP(G) is used, the generators $\mathbf{x}_1, \ldots, \mathbf{x}_m$ must be publicly known. To do so, one can use a matrix $\mathbf{M}_G$ in systematic form, i.e., $\mathbf{M}_G = (\mathbf{I}_m, \mathbf{U})$ with $\mathbf{U} \in \mathbb{F}_z^{m \times (n-m)}$ sampled at random from the seed. Since $\mathbf{M}_G$ is guaranteed to have full rank, we can then take its rows $\mathbf{m}_i$ and define the generators $\mathbf{x}_1, \ldots, \mathbf{x}_m$ as the vectors with exponents $\mathbf{m}_1, \ldots, \mathbf{m}_m$ respectively. The seed can also be used to sample $\mathbf{H} \in \mathbb{F}_q^{r \times n}$ (again, one can conveniently use the systematic form). This way, the public key is $\{\mathbf{s}, \mathtt{Seed}_{\mathsf{pk}}\}$ and has size $(n-k)\log_2(q) + |\mathtt{Seed}_{\mathsf{pk}}|$; we will use $\mathtt{Seed}_{\mathsf{pk}}$ with λ bits.

We consider that both restricted maps and vectors are always sampled from G. Property (3), which guarantees ZK, holds since G acts transitively and freely on itself. In other words, for any $\mathbf{e} \in G$, $\sigma(\mathbf{e})$ is uniformly random over G when $\sigma \xleftarrow{\$} G$. When $\mathbf{e}$ and $\widetilde{\mathbf{e}}$ are two restricted vectors, the map σ that maps $\widetilde{\mathbf{e}}$ into $\mathbf{e}$ is $\ell_G(\mathbf{e}) - \ell_G(\widetilde{\mathbf{e}})$ and it can be represented using only $m\log_2(z)$ bits. To sample uniformly at random some $\mathbf{a} \in G$, a convenient procedure is:

1. sample $\mathbf{u} \xleftarrow{\$} \mathbb{F}_z^m$,
2. obtain the exponents $\mathbf{x} = \mathbf{u}\mathbf{M}_G \in \mathbb{F}_z^n$,
3. set $\mathbf{a} = \left(g^{x_1}, \ldots, g^{x_n}\right) \in \mathbb{F}_q^n$.

Using $\mathbf{M}_G$ in systematic form, we have some computational advantages since $\mathbf{x} = (\mathbf{u}, \mathbf{u}\mathbf{U})$ and computing $\mathbf{u}\mathbf{U}$ requires only $O\big(m(n-m)\big)$ operations over $\mathbb{F}_z$. To verify that some $\mathbf{a} \in \mathbb{F}_q^n$ is indeed in G, it is enough to check that $\ell(\mathbf{a})$ is a linear combination of the rows of $\mathbf{M}_G$. This can be done using a parity-check matrix $\mathbf{C} \in \mathbb{F}_z^{(n-m)\times n}$ for $\langle \mathbf{M}_G \rangle$: $\mathbf{a} \in G$, if and only if $\ell(\mathbf{a})\mathbf{C}^\top = \mathbf{0}$. We can set $\mathbf{C} = (-\mathbf{U}^\top, \mathbf{I}_{n-m})$, which speeds up the computation of $\ell(\mathbf{a})\mathbf{C}^\top$.

We show that even with the conservative Requirement 2, R-SDP(G) allows us to use much more aggressive parameters than those for R-SDP. From now, on we will write $|G| = z^m = 2^{(1+\alpha)\lambda}$: the value of α gives an idea of how tight we can be, when representing elements of G. We clearly require $\alpha > 0$ to thwart brute-force attacks, yet, by choosing $0 < \alpha < 1$, we can have restricted objects with sizes that are not greater than 2λ, that is, the binary size of a digest. In other words, we are making restricted objects smaller than some of the objects that the parties cannot avoid exchanging (e.g., the initial commitments). As we show in the following, we can use α in the range 0.2 to 0.6: which means we are very close to achieving security with the minimum amount of required bits, i.e., λ bits.

Some example instances for R-SDP(G) used in CVE are shown in Table 2. The parameters are chosen according to the cost of the ISD algorithm presented in Section 5.3 and Requirement 2. Recall from Figure 2 that $\mathbf{y}$ and σ have size $n\log_2(q)$, respectively, $m\log_2(z)$ and that the parameters m, z are chosen such that $z^m > 2^\lambda$.

Table 2: Instances of R-SDP(G) for $\lambda = 128$ and corresponding sizes for objects expressed in bytes.

Range for q	$\frac{z}{q-1}$	q	z	n	k	m	α	Size(σ)	Size($\mathbf{y}$)
$2^8 < q < 2^{10}$	$1/2$	1019	509	40	16	18	0.2644	20.2	49.9
		347	173	41	20	23	0.3359	21.4	43.2
		719	359	49	17	20	0.3262	21.2	58.1
	$< 1/2$	971	97	44	26	26	0.3406	21.4	54.8
		643	107	60	25	26	0.3604	21.9	70.0
		269	67	52	27	29	0.3743	21.9	52.5
$2^6 < q < 2^8$	$1/2$	227	113	43	22	24	0.2789	20.5	42.1
		107	53	53	26	31	0.3872	22.2	44.7
		83	41	73	28	35	0.4650	23.4	58.2
	$< 1/2$	223	37	56	33	34	0.3838	22.1	54.6
		103	17	76	44	48	0.5328	24.5	63.5
		79	13	82	49	54	0.5611	25.0	64.6
$2^4 < q < 2^6$	$1/2$	59	29	63	31	38	0.4422	23.1	46.3
		47	23	69	34	42	0.4843	23.7	47.9
		23	11	93	46	61	0.6486	26.4	52.6
	$< 1/2$	53	13	82	47	54	0.5611	25.0	58.7

We see that there are several trade-offs in how parameters can be chosen. For instance, large values of q lead to slightly smaller sizes for $\mathbf{y}$, while the arithmetic over $\mathbb{F}_q$ becomes slower. Another degree of freedom is in the choice of z: setting $z = \frac{q-1}{2}$ leads to smaller sizes, but choosing large z might make the arithmetic over $\mathbb{F}_z$ slower. Comparing these numbers with those in Table 1, we see that using R-SDP(G) allows for a significant reduction of the communication cost. In the next sections, we apply the problem to existing ZK schemes and derive their performances in terms of signature size.

6 ZK Protocols from the R-SDP: Modern Protocols

This section presents concrete ZK protocols based on R-SDP and R-SDP(G). Note that one can replace the SDP with R-SDP or R-SDP(G) in any ZK protocol, however, in the following, we only present the two schemes which result in the smallest signature sizes, namely the GPS [31] and BG protocol [18].

6.1 R-GPS: the GPS Scheme with R-SDP

The GPS scheme [31] applies the protocol-with-helper paradigm to the CVE scheme. In a nutshell, the idea is that of simulating a trusted third entity (the helper), which generates some of the messages which would be exchanged between the prover and the verifier. The helper is asked to generate the commitments and the first public response (that is, c_0, c_1 and $\mathbf{y}$ for the scheme in Figure

2). The *cut&choose* technique is used to remove the helper. The helper is first simulated by the prover for N rounds, generating random objects from seeds and committing to the obtained quantities. The verifier will ask to *open* only $M < N$ rounds: she will receive the verifying maps for the chosen rounds and the seeds for the other $N - M$ rounds.

Since GPS is based on SDP, converting it to R-SDP is rather straightforward; for the sake of completeness, the resulting protocol is presented in Figure 4. As applying the Fiat-Shamir transform on a ZK protocol is straightforward and a well-known procedure, we omit to write out the resulting signature scheme.

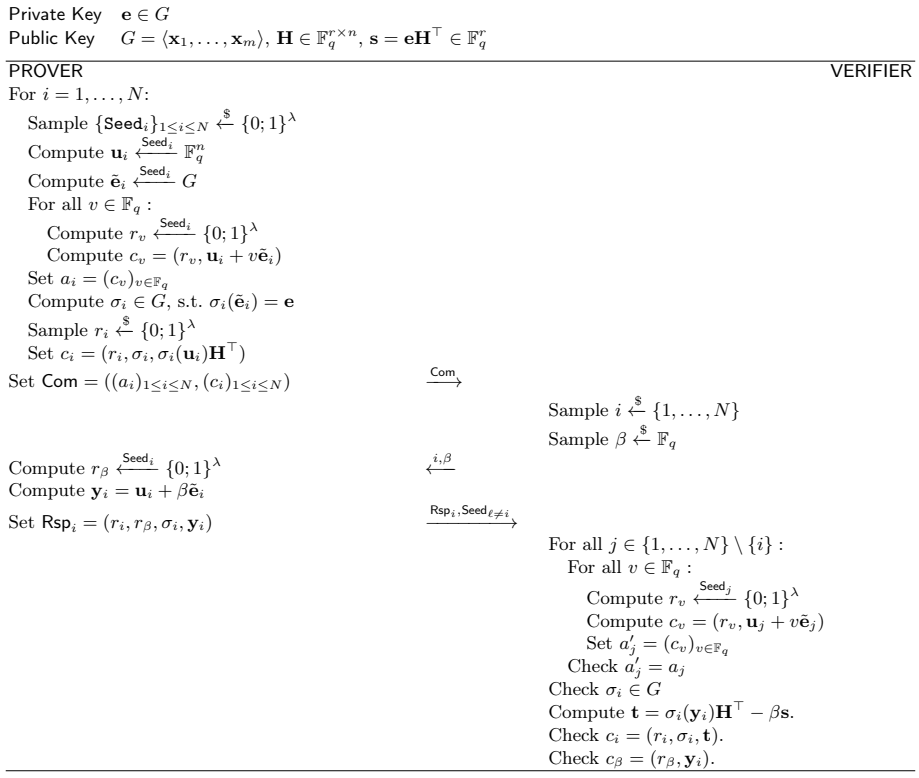

Fig. 4: One round of the R-GPS protocol

The ZK property, as well as soundness and EUF-CMA security, are obtained in the exact same way as for the original GPS scheme. Also, the security analysis and signature size easily follow from [31].

To prevent the attack in [35], N and M must be such that

$$\max_{x \in \{0,\ldots,M\}} \binom{N-x}{M-x} \binom{N}{M}^{-1} (q-1)^{x-M} \ge 2^\lambda. \tag{7}$$

When R-SDP(G) is used, the communication cost of an opened round is

$$L = \underbrace{n \log_2(q)}_{\mathbf{y} \in \mathbb{F}_q^n} + \underbrace{2\lambda}_{\text{Randomness}} + \underbrace{m\lceil \log_2(z) \rceil}_{\sigma \in G}. \tag{8}$$

When relying on R-SDP, the resulting communication cost is obtained by replacing $m\lceil \log_2(z) \rceil$ with $n\lceil \log_2(z) \rceil$. The size of a signature in the resulting R-GPS signature scheme is

$$|\texttt{Signature}| = \underbrace{2\lambda \left(2 + M \log_2 \left(\frac{N(q-1)}{M} \right) \right)}_{\text{Merkle proofs and commitments}} + \underbrace{\lambda M \log_2 \left(\frac{N}{M} \right)}_{\text{Seeds}} + M \cdot L.$$

In Table 3, we report examples for the signature sizes we can achieve and compare them with the ones in [31, Table 1]. We have chosen the parameters according to the cost of the generic decoders (Section 5.3 and Appendix B) and the soundness error. Employing R-SDP, we can reduce the signature sizes by a factor of approximately 0.6. Considering R-SDP(G), the gain becomes more significant, and, with respect to the instances based on R-SDP, we save approximately 1 to 2 kB.These might not be the optimal parameter choices to obtain the security level of 128 bits, but already show the great potential of R-SDP.

Table 3: Performances of the GPS scheme [31] based on different problems.

	q	z	n	k	w	m	N	M	Sign. Size (kB)
SDP	128		220	101	90		512	23	24.6
	256		207	93	90		1024	19	22.4
	512		196	92	84		2024	16	20.6
	1024		187	90	80		4096	14	19.5
R-SDP	67	11	147	63	147		512	24	14.8
	197	14	105	53	105		1024	19	13.4
	991	33	77	48	77		2048	16	12.9
	991	33	77	38	77		4096	14	12.5
R-SDP(G)	53	13	82	47	82	54	512	25	12.7
	103	17	76	44	76	48	1024	21	12.7
	223	37	56	33	56	34	2048	19	11.8
	1019	509	40	16	40	18	4096	14	11.5

6.2 R-BG: the BG-PKP Scheme with R-SDP

As another protocol-with-helper, one may consider the FJR scheme [28]. To reduce the soundness error, FJR uses the idea of shared permutations: the random masking is obtained by combining the actions of N random permutations, so that

a cheating prover cannot cheat for more than one permutation. This reduces the soundness error of a single round to $1/N$. The idea of shared permutations has been applied also to PKP, for a protocol that we will refer to as BG-PKP [18]. Notice that BG-PKP is the PKP-based scheme with the smallest signatures. We show that, with minor modifications, the scheme can be adapted to the R-SDP setting and derive the resulting signature sizes.

Private Key $\sigma \in G$
Public Key $G = \langle \mathbf{x}_1, \ldots, \mathbf{x}_m \rangle$, $\mathbf{e} \in G$, $\mathbf{H} \in \mathbb{F}_q^{r \times n}$, $\mathbf{s} = \sigma(\mathbf{e})\mathbf{H}^\top \in \mathbb{F}_q^r$

PROVER		VERIFIER
Sample $\mathtt{MSeed} \xleftarrow{\$} \{0;1\}^\lambda$		
Compute $\{\mathtt{Seed}_i\}_{1 \le i \le N} = \mathsf{SeedTree}(\mathtt{MSeed})$		
For $i = 2, \ldots, N$:		
Sample $\mathtt{Seed}_i^*, \mathtt{Salt}_i \xleftarrow{\mathtt{Seed}_i} \{0;1\}^\lambda$		
Sample $\sigma_i \xleftarrow{\mathtt{Seed}_i^*} G$, $\mathbf{v}_i \xleftarrow{\mathtt{Seed}_i^*} \mathbb{F}_q^n$		
Set $c_i = \mathsf{Hash}\big(\mathtt{Salt}_i, \mathtt{Seed}_i^*\big)$		
Set $\sigma_1 = \sigma_2^{-1} \circ \cdots \circ \sigma_N^{-1} \circ \sigma$		
Sample $\mathtt{Seed}_1^*, \mathtt{Salt}_1 \xleftarrow{\mathtt{Seed}_1} \{0;1\}^\lambda$		
Sample $\mathbf{v}_1 \xleftarrow{\mathtt{Seed}_1^*} \mathbb{F}_q^n$		
Set $c_1 = \mathsf{Hash}\big(\mathtt{Salt}_1, \mathtt{Seed}_1^*, \sigma_1\big)$		
Compute $\mathbf{v} = \mathbf{v}_N + \sum_{i=1}^{N-1} \sigma_N \circ \cdots \circ \sigma_{i+1}(\mathbf{v}_i)$		
Set $c = \mathsf{Hash}\big(\mathbf{v}\mathbf{H}^\top, \{c_i\}_{1 \le i \le N}\big)$	$\xrightarrow{c}$	
		Sample $\beta \xleftarrow{\$} \mathbb{F}_q^*$
Set $\widetilde{\mathbf{e}}_0 = \beta\mathbf{e}$	$\xleftarrow{\beta}$	
For $i = 1, \ldots, N$:		
Set $\widetilde{\mathbf{e}}_i = \sigma_i(\widetilde{\mathbf{e}}_{i-1}) + \mathbf{v}_i$		
Set $h = \mathsf{Hash}\big(\{\widetilde{\mathbf{e}}_i\}_{1 \le i \le N}\big)$	$\xrightarrow{h}$	
		Sample $i \xleftarrow{\$} \{1, \ldots, N\}$
Compute $\mathtt{Seeds} = \mathsf{SeedPath}(\mathtt{MSeed}, i)$	$\xleftarrow{i}$	
If $i \neq 1$:		
Set $\mathtt{Resp} = \{c_i, \widetilde{\mathbf{e}}_i, \sigma_1, \mathtt{Seeds}\}$		
Else:		
Set $\mathtt{Resp} = \{c_1, \widetilde{\mathbf{e}}_1, \mathtt{Seeds}\}$	$\xrightarrow{\mathtt{Resp}}$	
		Generate $\{\mathtt{Seed}_j\}_{j \neq i} = \mathsf{GenSeeds}(\mathtt{Seeds})$
		For $j \neq i$:
		Sample $\mathtt{Seed}_j^*, \mathtt{Salt}_j \xleftarrow{\mathtt{Seed}_j} \{0;1\}^\lambda$
		Sample $\sigma_j \xleftarrow{\mathtt{Seed}_j^*} G$, $\mathbf{v}_j \xleftarrow{\mathtt{Seed}_j^*} \mathbb{F}_q^n$
		Set $c_j = \mathsf{Hash}\big(\mathtt{Salt}_j, \mathtt{Seed}_j^*\big)$
		If $i \neq 1$:
		Compute $c_1 = \mathsf{Hash}\big(\mathtt{Salt}_1, \mathtt{Seed}_1^*, \sigma_1\big)$
		Set $\widetilde{\mathbf{e}}_0 = \beta\mathbf{e}$
		For $j \neq i$:
		$\widetilde{\mathbf{e}}_j = \sigma_j(\widetilde{\mathbf{e}}_{j-1}) + \mathbf{v}_j$
		Compute $\widetilde{\mathbf{s}} = \widetilde{\mathbf{e}}_N\mathbf{H}^\top - \beta\mathbf{s}$
		Verify $c = \mathsf{Hash}\big(\widetilde{\mathbf{s}}, \{c_j\}_{1 \le j \le N}\big)$
		Verify $h = \mathsf{Hash}\big(\{\widetilde{\mathbf{e}}_j\}_{1 \le j \le N}\big)$
		If $i \neq 1$: verify also $\sigma_1 \in G$

Fig. 5: One round of the R-BG protocol

For PKP, the prover first samples a vector $\mathbf{e} \in \mathbb{F}_q^n$, a full rank $\mathbf{H} \in \mathbb{F}_q^{r \times n}$, a permutation $\pi \in S_n$ and computes $\mathbf{s} = \pi(\mathbf{e})\mathbf{H}^\top$. The secret key is the permutation π and the public key is $\{\mathbf{H}, \mathbf{e}, \mathbf{s}\}$. For R-SDP, we can do the same, with the only difference that $\mathbf{e}$ and the map σ are sampled from the restricted subgroup G; namely, once $\mathbf{H}$ and G have been defined, we sample $\mathbf{e}, \sigma \xleftarrow{\$} G$, set the secret key as σ and the public key as $\{G,\ \mathbf{H},\ \mathbf{e},\ \mathbf{s} = \sigma(\mathbf{e})\mathbf{H}^\top\}$. To compress the public key size, everything but $\mathbf{s}$ can be generated from a seed $\mathtt{Seed}^{(pk)}$.

The resulting protocol is shown in Figure 5. We have implicitly introduced some additional notation: $\mathsf{SeedTree}$, $\mathsf{SeedPath}$ are the functions to operate with the seed tree (respectively, generate the tree from a master seed, compute a path and regenerate all seeds but one), while $\xleftarrow{\mathtt{Seed}}$ means sampling with randomness source $\mathtt{Seed}$. It can be seen that the protocol structure is the same as BG, so it inherits all of its features, in particular, the soundness error is as in [18, Thm. 2]

$$\varepsilon(N, q) = \frac{1}{N} + \frac{N-1}{N(q-1)}.$$

Also, the completeness and ZK property follow in a straightforward manner. Nevertheless, we give the proof for soundness, completeness, and the ZK property in Appendix E.

Signature scheme To obtain a signature scheme, we consider t parallel executions and then apply the Fiat-Shamir transform. The corresponding algorithms for signature generation and verification are given in Figure 6 and 7. In the algorithms, we have indicated by $\mathtt{Msg}$ the message to be signed and by t the number of executed rounds. The round index has been indicated with u, and the quantities referred to each round are specified by the superscript (u). For instance, $\sigma_1^{(u)}, \ldots, \sigma_N^{(u)}$ are the transformations used in the u-th round. The resulting scheme is essentially the repetition of t rounds of the R-BG protocol, plus some minor modifications which we list below.

- A length-2λ salt is employed for the commitments (namely, in the computation of each $c_i^{(u)}$ and $c^{(u)}$), as well as to generate the challenges. This is necessary to prevent certain types of attacks (see e.g., [21]).
- As recommended in [21], to compute seeds and commitments, the salt is concatenated also with the round index u.
- To reduce the signature the signature size, the t commitments $c^{(1)}, \ldots, c^{(t)}$ are hashed into a single commitment c. The validity of c is checked at the end of the verification algorithm. The same optimization is employed for the t first responses $h^{(1)}, \ldots, h^{(t)}$.

Private Key $\sigma \in G$
Public Key $G = \langle \mathbf{x}_1, \ldots, \mathbf{x}_m \rangle$, $\mathbf{e} \in G$, $\mathbf{H} \in \mathbb{F}_q^{r \times n}$, $\mathbf{s} = \sigma(\mathbf{e})\mathbf{H}^\top \in \mathbb{F}_q^r$

PROVER	VERIFIER
Sample $\mathtt{Salt} \xleftarrow{\$} \{0;1\}^{2\lambda}$	
For $u = 1, \ldots, t$:	
Sample $\mathtt{MSeed}^{(u)} \xleftarrow{\$} \{0;1\}^{\lambda}$	
Set $\overline{\mathtt{MSeed}}^{(u)} = \big(\mathtt{MSeed}^{(u)}, \mathtt{Msg}, \mathtt{Salt}, u\big)$	\\Concatenate salt and round index to master seed
Compute $\{\mathtt{Seed}_i^{(u)}\}_{1 \le i \le N} = \mathsf{SeedTree}(\overline{\mathtt{MSeed}}^{(u)})$	\\Generate seeds for u-th round
For $i = 2, \ldots, N$:	
Sample $\mathtt{Seed}_i^{*(u)}, \mathtt{Salt}_i^{(u)} \xleftarrow{\mathtt{Seed}_i^{(u)}} \{0;1\}^{\lambda}$	
Sample $\sigma_i^{(u)} \xleftarrow{\mathtt{Seed}_i^{*(u)}} G$, $\mathbf{v}_i^{(u)} \xleftarrow{\mathtt{Seed}_i^{*(u)}} \mathbb{F}_q^n$	
Set $c_i^{(u)} = \mathsf{Hash}\big(\mathtt{Salt}_i^{(u)}, \mathtt{Seed}_i^{*(u)}, \mathtt{Msg}, \mathtt{Salt}, u\big)$	\\Hash also message, salt and round index
Set $\sigma_1^{(u)} = \sigma_2^{(u)-1} \circ \cdots \sigma_N^{(u)-1} \circ \sigma^{(u)}$	
Sample $\mathtt{Seed}_1^{*(u)}, \mathtt{Salt}_1^{(u)} \xleftarrow{\mathtt{Seed}_1^{(u)}} \{0;1\}^{\lambda}$	
Sample $\mathbf{v}_1^{(u)} \xleftarrow{\mathtt{Seed}_1^{*(u)}} \mathbb{F}_q^n$	
Set $c_1^{(u)} = \mathsf{Hash}\big(\mathtt{Salt}_1^{(u)}, \mathtt{Seed}_1^{*(u)}, \sigma_1^{(u)}, \mathtt{Msg}, \mathtt{Salt}, u\big)$	
Compute $\mathbf{v}^{(u)} = \mathbf{v}_N^{(u)} + \sum_{i=1}^{N-1} \sigma_N^{(u)} \circ \cdots \circ \sigma_{i+1}^{(u)}(\mathbf{v}_i^{(u)})$	
Set $c^{(u)} = \mathsf{Hash}\big(\mathbf{v}^{(u)}\mathbf{H}^\top, \{c_i^{(u)}\}_{1 \le i \le N}, \mathtt{Msg}, \mathtt{Salt}, u\big)$	
Set $c = \mathsf{Hash}\big(\{c^{(u)}\}_{1 \le u \le t}\big)$	\\Single commitment of size 2λ
Set $\big(\beta^{(1)}, \ldots, \beta^{(t)}\big) = \mathsf{Hash}\big(\mathtt{Msg}, \mathtt{Salt}, c\big)$	\\Generate first challenge
For $u = 1, \ldots, t$:	
Set $\widetilde{\mathbf{e}}_0^{(u)} = \beta^{(u)}\mathbf{e}$	
For $i = 1, \ldots, N$:	
Set $\widetilde{\mathbf{e}}_i^{(u)} = \sigma_i^{(u)}(\widetilde{\mathbf{e}}_{i-1}^{(u)}) + \mathbf{v}_i^{(u)}$	
Set $h^{(u)} = \mathsf{Hash}\big(\{\widetilde{\mathbf{e}}_i^{(u)}\}_{1 \le i \le N}\big)$	
Set $h = \mathsf{Hash}\big(\{h^{(u)}\}_{1 \le u \le t}\big)$	
	\\Single first response of size 2λ
Set $\big(i^{(1)}, \ldots, i^{(t)}\big) = \mathsf{Hash}\big(\mathtt{Msg}, \mathtt{Salt}, c, h\big)$	\\Generate second challenge
For $u = 1, \ldots, t$:	
Compute $\mathtt{Seeds}^{(u)} = \mathsf{SeedPath}(\overline{\mathtt{MSeed}}^{(u)}, i^{(u)})$	\\Compute seeds path for u-th round
If $i^{(u)} \neq 1$:	
Set $\mathtt{Resp}^{(u)} = \{c_i^{(u)}, \widetilde{\mathbf{e}}_i^{(u)}, \sigma_1^{(u)}, \mathtt{Seeds}^{(u)}\}$	
Else:	
Set $\mathtt{Resp}^{(u)} = \{c_1^{(u)}, \widetilde{\mathbf{e}}_1^{(u)}, \mathtt{Seeds}^{(u)}\}$	
Set $\mathtt{Signature} = \Big\{\mathtt{Salt}, c, h, \{\mathtt{Rsp}^{(u)}\}_{1 \le u \le t}\Big\}$	
	$\xrightarrow{\mathtt{Signature}}$

Fig. 6: The R-BG signature scheme: signature generation

To set the value of t such that the attack in [35] is mitigated, we rely on the analysis in [18, Section 4.2]. To this end, let

$$P(t', t, N) = \sum_{j=t'}^{t} \binom{t}{j} \left(\frac{1}{q-1}\right)^j \left(\frac{N-1}{N}\right)^{t-j},$$

$$t^* = \arg\min_{0 \le x \le t} \left\{ \frac{1}{P(x, t, N)} + N^{t-x} \right\}. \tag{9}$$

Then, we choose t so that $P(t^*, t, N)^{-1} + N^{t-t^*} > 2^\lambda$.

Private Key $\sigma \in G$
Public Key $G = \langle \mathbf{x}_1, \ldots, \mathbf{x}_m \rangle$, $\mathbf{e} \in G$, $\mathbf{H} \in \mathbb{F}_q^{r \times n}$, $\mathbf{s} = \sigma(\mathbf{e})\mathbf{H}^\top \in \mathbb{F}_q^r$

PROVER		VERIFIER
	$\xrightarrow{\texttt{Signature}}$	
Generate first challenge\\		Set $\left(\beta^{(1)}, \ldots, \beta^{(t)}\right) = \mathsf{Hash}\big(\mathtt{Msg}, \mathtt{Salt}, c\big)$
Generate second challenge\\		Set $\left(i^{(1)}, \ldots, i^{(t)}\right) = \mathsf{Hash}\big(\mathtt{Msg}, \mathtt{Salt}, c, h\big)$
		For $u = 1, \ldots, t$:
Generate seeds for u-th round \\		Generate $\{\mathtt{Seed}_j^{(u)}\}_{j \neq i} = \mathsf{GenSeeds}(\mathtt{Seeds}^{(u)})$
		For $j \neq i^{(u)}$:
		Sample $\mathtt{Seed}_j^{*(u)}, \mathtt{Salt}_j^{(u)} \xleftarrow{\mathtt{Seed}_j^{(u)}} \{0;1\}^\lambda$
		Sample $\sigma_j^{(u)} \xleftarrow{\mathtt{Seed}_j^{*(u)}} G$, $\mathbf{v}_j^{(u)} \xleftarrow{\mathtt{Seed}_j^{*(u)}} \mathbb{F}_q^n$
		Set $c_j^{(u)} = \mathsf{Hash}\big(\mathtt{Salt}_j, \mathtt{Seed}_j^*, \mathtt{Msg}, \mathtt{Salt}, u\big)$
		If $i^{(u)} \neq 1$:
		Compute $c_1^{(u)} = \mathsf{Hash}\big(\mathtt{Salt}_1^{(u)}, \mathtt{Seed}_1^{*(u)}, \sigma_1^{(u)}, \mathtt{Msg}, \mathtt{Salt}, u\big)$
		Set $\widetilde{\mathbf{e}}_0^{(u)} = \beta^{(u)}\mathbf{e}$
		For $j \neq i^{(u)}$:
		$\widetilde{\mathbf{e}}_j^{(u)} = \sigma_j^{(u)}(\widetilde{\mathbf{e}}_{j-1}^{(u)}) + \mathbf{v}_j^{(u)}$
		Compute $\widetilde{\mathbf{s}}^{(u)} = \widetilde{\mathbf{e}}_N^{(u)}\mathbf{H}^\top - \beta^{(u)}\mathbf{s}$
		Compute $c^{(u)} = \mathsf{Hash}\big(\widetilde{\mathbf{s}}, \{c_j\}_{1 \leq j \leq N}, \mathtt{Msg}, \mathtt{Salt}, u\big)$
		Compute $h^{(u)} = \mathsf{Hash}\big(\{\widetilde{\mathbf{e}}_j^{(u)}\}_{1 \leq j \leq N}\big)$
Reject the signature if $\sigma_1^{(u)}$ is badly formed\\		If $\left(i^{(u)} = 1\right) \wedge \left(\sigma_1^{(u)} \notin G\right)$:
		Output `Reject`
Verify single commitment\\		If $c \neq \mathsf{Hash}\big(\{c_{1 \leq u \leq t}^{(u)}\}\big)$:
		Output `Reject`
Verify single first response\\		If $h \neq \mathsf{Hash}\big(\{h_{1 \leq u \leq t}^{(u)}\}\big)$:
		Output `Reject`
Signature is valid\\		Output `Accept`

Fig. 7: The R-BG signature scheme: signature verification

The signature size is given by

$$|\mathtt{Signature}| = 6\lambda + t\Big(\underbrace{n\lceil \log_2(q) \rceil}_{\widetilde{\mathbf{e}}_i} + \underbrace{m\lceil \log_2(z) \rceil}_{\sigma_1} + \underbrace{\lambda\lceil \log_2(N) \rceil}_{\text{Seeds}} + \underbrace{2\lambda}_{c_i}\Big).$$

When R-SDP(G) is used, $n\lceil \log_2(z) \rceil$ gets replaced by $m\lceil \log_2(z) \rceil$.

Some instances of the resulting signature scheme are reported in Table 4, where the parameters are chosen in accordance with the above formula for the number of executions and the respective generic decoders. Signatures obtained from R-SDP are slightly larger than those based on PKP; instead, when using R-SDP(G), we achieve significant reductions with respect to R-SDP and, ultimately, beat PKP. We have considered the case of $z = (q-1)/2$ and $z \ll q$; the latter has slightly larger signatures but, when implemented, should lead to a faster scheme, due to the arithmetics over $\mathbb{F}_z$.

Timings We have developed a Proof of Concept implementation for the R-BG protocol based on R-SDP(G)[6,7]. The measured timings are reported in Table 5.

[6] `https://github.com/secomms/RBG`

[7] The provided code considers only one round of the protocol. Multiplying the timings by t (the number of parallel executions), we obtain a very reliable estimate of the overall required time. Indeed, applying the Fiat-Shamir transform requires only a negligible overhead (namely, two additional hashes).

Table 4: Performances of the BG scheme [18] based on different problems

	q	z	n	k	m	N	t	Sign. Size (kB)
PKP	997		61	33		32	42	10.0
						256	31	8.9
R-SDP	991	33	77	38		32	42	10.0
						256	31	8.9
R-SDP(G)	971	97	44	26	26	32	42	8.1
						256	31	7.5
	1019	509	40	16	18	32	42	7.8
						256	31	7.2

Table 5: Benchmarks for the R-BG protocol based on R-SDP(G), taken on a 3.4 GHz Intel i7-6700 CPU. The reported timings are the average values, measured with 10 000 tests.

Parameters	variant	Sign. Size (kB)	**KeyGen**		**Sign**		**Verify**	
(q, z, n, k, m)			MCycles	ms	MCycles	ms	MCycles	ms
971, 97, 44, 26, 26	fast	8.1	0.06	< 0.1	18.7	5.46	12.2	3.57
	small	7.5	0.06	< 0.1	108.4	31.08	72.5	21.3
1019, 509, 40, 16, 18	fast	7.8	0.05	< 0.1	20.4	6.0	12.8	3.8
	small	7.2	0.05	< 0.1	117.7	34.5	75.8	22.2

Even though the implementation is very basic and does not use any advanced optimization (e.g., no AVX2 instructions nor parallelism), the obtained timings are already very promising. This was expected, as all the required operations are essentially symmetric primitives and linear algebra (multiplications and sums) with small vectors and matrices. As expected, choosing small values for z leads to some speed-ups, since operations over $\mathbb{F}_z$ (e.g., sampling from G and combining restricted objects) get easier.

7 Comparison with NIST Candidates

In this section, compare the schemes discussed in this paper with signature schemes submitted to the NIST additional call. We first consider code-based schemes and then widen the comparison to other relevant schemes.

Comparison with code-based signatures In Table 6, we compare the two schemes R-GPS and R-BG with the code-based signature schemes submitted to NIST. Data about these schemes are also visualized in Figure 8; Figure 8a reports signature sizes and public key size, while Figure 8b shows signatures size and verification times[8]. All schemes in the category use the ZK/MPCitH

[8] Which we have collected from `https://pqshield.github.io/nist-sigs-zoo/`. Data are referred to October 15, 2023.

Table 6: Comparison between NIST submissions and R-BG, R-GPS for Level I; all sizes are expressed in kB.

Problem	**Scheme**	**Pk size**	**Sign. size**	**Pk+Sign. size**	**Variant**
SDP	SDitH [3]	0.12	8.24	8.36	small
		0.12	10.12	10.24	fast
Rank SDP	RYDE [5]	0.09	5.96	6.04	small
		0.09	7.45	7.53	fast
Matrix rank SDP	MIRA [6]	0.08	5.64	5.72	small
		0.08	7.38	7.46	fast
	MiRitH [2]	0.13	4.54	4.67	small
		0.14	9.11	9.25	fast
PKP	PERK [1]	0.24	6.10	6.30	small
		0.15	8.35	8.50	fast
large weight $(U, U+V)-$code	WAVE [12]	3677	0.82	3678	-
Code Equivalence	LESS [9]	14.03	8.60	22.63	small
		98.20	5.33	103.53	fast
Matrix code equivalence	MEDS [23]	9.92	9.90	19.82	small
		13.22	12.98	26.20	fast
R-SDP	CROSS [10]	0.06	10.30	10.36	small
		0.06	12.94	13.01	fast
	R-BG	0.1	8.9	9.0	small
		0.1	10.0	10.1	fast
	R-GPS	0.1	12.5	12.6	small
		0.1	14.8	14.9	fast
R-SDP(G)	CROSS [10]	0.04	7.63	7.66	small
		0.04	8.67	8.7	fast
	R-BG	0.1	7.2	7.3	small
		0.1	7.8	7.9	fast
	R-GPS	0.1	11.5	11.6	small
		0.1	12.7	12.8	fast

approach and have a signing time that is more or less equal to the verification time. The only remarkable exception is WAVE, which is a hash&sign scheme, and for which signing is approximately 5 times slower than verifying. Note that some of the proposed schemes have more than just the usual two "fast" and "small" parameter sets, however, in order for all the schemes to fit into one table, we chose to show for each scheme the smallest and largest total sizes, i.e., signature size plus public key size. This is a common measure, as for example certificates would require to download both. We did not include the two broken schemes FuLeeca [37] and Enhanced pqsigRM [22]. We only compare the schemes for Level 1, which corresponds to 128 AES gates.

For R-GPS, the fast instances are those with $N = 512$ and the short ones have $N = 4096$; for R-BG, fast instances have $N = 32$ and $t = 42$ and short variants have $N = 256$ and $t = 31$. As we see from Table 6, the presented

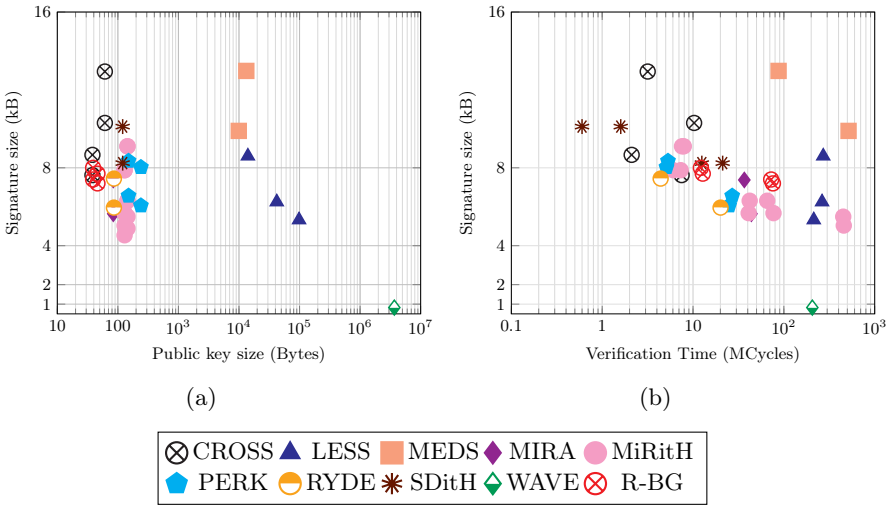

Fig. 8: Comparison between R-BG and code-based signature schemes from the NIST competition.

protocols are very competitive towards the submitted schemes. In particular, comparing R-BG using R-SDP(G), we are able to achieve signatures that are smaller than those of SDitH [3], which is based on SDP.

Among the considered schemes, there is also CROSS [10], a NIST candidate that originated from the work presented in this paper. CROSS uses a very simple ZK protocol, with soundness error $\approx \frac{1}{2}$, which can be thought of as an optimized version of the CVE protocol [20]. It has been designed aiming for algorithmic efficiency and simplicity, while R-BG aims to reduce signature sizes at the cost of some computational overhead. The trade-offs achieved by the two schemes are different, as it is visible in Figure 8. Observe that the only scheme with faster verification is SDitH. Notice that timings for SDitH are already referred to those of an optimized implementation; instead, the timings for CROSS are referred to the reference implementation. Likely, an optimized implementation can significantly boost timings and make CROSS much faster.

More generally, we see that solutions based on R-SDP and R-SDP(G) compare favorably with the other code-based schemes. Again, a remarkable exception is WAVE which has much shorter signatures but has much larger public keys and is slower. LESS and MEDS, instead, have a somewhat similar profile analysis. They have large public keys and are generically slower than the other schemes. Signature sizes are in the same ballpark as those of the other schemes, even though LESS instances with larger public keys have shorter signatures (using more equivalent codes in the public key reduces linearly the soundness error).

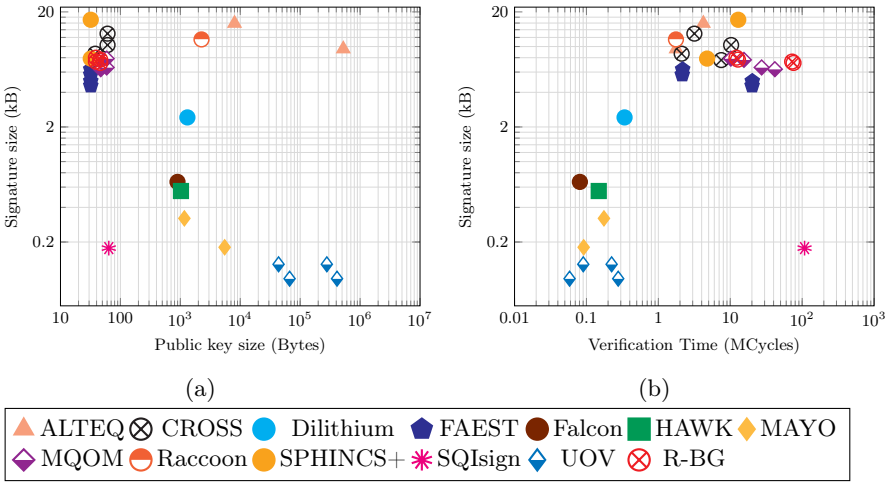

Fig. 9: Comparison between CROSS, R-BG, and other signature schemes from the NIST competition

Finally, we remark that we have not explored optimizations for the implementation of R-BG. We expect that a more careful implementation can receive a significant boost.

Comparison with signatures based on other problems To see more comparisons, in Figure 9 we consider other relevant signature schemes from the NIST competition. The comparison shows that both CROSS and R-BG have performances which are, essentially, analogous to those of FAEST and MQOM (even though the latter scheme appears to be slower); this was somehow expected, since also these two schemes are based on ZK/MPCitH paradigms. SQIsign has signatures which are among the shortest in the competition and very compact public keys, but pays a significant price in terms of efficiency (it is the slowest scheme among those in Figure 9b).

The figures show that there is still a significant gap between schemes based on restricted errors and lattice-based schemes such as HAWK, Falcon and Dilithium (for both signature sizes and timings). An exception is Raccoon, whose signatures and timings are in the same range as those of CROSS and R-BG, hence, are larger than those of typical lattice-based schemes: this is not surprising since, differently from the other lattice-based schemes, Raccoon has been designed aiming for inherent protection against side-channel attacks.

Multivariate hash&sign schemes (MAYO and UOV) have shorter signatures and are faster, but require larger public keys, in particular, UOV. The public keys of MAYO are significantly shorter but the scheme is based on somewhat very new security assumptions. MQOM is, instead, a multivariate-based MPCitH scheme: its performances are essentially analogous to those of CROSS and R-BG.

Finally, we consider SPHINCS+, which comes with two versions. The short version has signatures which are essentially as large as those of R-BG ($\approx 8\,\text{kB}$), while the fast version has much larger signatures ($\approx 17\,\text{kB}$). Verification for SPHINCS+ takes time in the same ballpark as both CROSS and R-BG, for what concerns verification. Signing is, for both the fast and short versions, much slower.

After these comparisons, we can conclude that using R-SDP and R-SDP(G) leads to competitive solutions. The performances for what concerns all the relevant figures (public keys, signatures and computational complexity) are analogous to those of other ZK/MPCitH schemes, even though schemes based on restricted errors are really promising for what concerns timings.

8 Conclusion

We studied the Restricted Syndrome Decoding Problem (R-SDP) and introduced a new version of the problem, called R-SDP(G). Both problems are NP-complete, however, as for most code-based problems it is unknown whether an average case reduction exists - we leave this as an interesting open problem. These two problems allow us to represent data to be exchanged in ZK protocols very compactly. We analyzed the security of these problems and gave conservative criteria for parameter choices. We adapted some existing ZK protocols to these new problems and considered the resulting signature schemes, called R-GPS and R-BG. The resulting schemes are able to achieve signatures in the order of $7\,\text{kB}$, which are highly competitive and compare well with other signature schemes submitted to NIST. The theory developed in this paper has been used as a basis for CROSS, a signature scheme submitted to the NIST call for additional signatures.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments.

Violetta Weger is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 899987.

Marco Baldi and Paolo Santini are supported by the Italian Ministry of University's PRIN 2022 program under the "Mathematical Primitives for Post Quantum Digital Signatures" (P2022J4HRR) and "POst quantum Identification and eNcryption primiTives: dEsign and Realization (POINTER)" (2022M2JLF2) projects funded by the European Union - Next Generation EU.

Sebastian Bitzer and Antonia Wachter-Zeh acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the program of "Souverän. Digital. Vernetzt.". Joint project 6G-life, project identification number: 16KISK002.

References

1. Aaraj, N., Bettaieb, S., Bidoux, L., Budroni, A., Dyseryn, V., Esser, A., Gaborit, P., Kulkarni, M., Mateu, V., Palumbi, M., Perin, L., Tillich, J.P.: PERK: PERmuted Kernels. Submission to the NIST Post-Quantum Standardization project (2023)

2. Adj, G., Rivera-Zamarripa, L., Verbel, J., Bellini, E., Barbero, S., Esser, A., Sanna, C., Zweydinger, F.: MiRitH: MinRank in-the-head. Submission to the NIST Post-Quantum Standardization project (2023)
3. Aguilar Melchor, C., Feneuil, T., Gama, N., Gueron, S., Howe, J., Joseph, D., Joux, A., Persichetti, E., H. Randrianarisoa, T., Rivain, M., Yue, D.: SDitH: Syndrome decoding in-the-head. Submission to the NIST Post-Quantum Standardization project (2023)
4. Aguilar-Melchor, C., Gama, N., Howe, J., Hülsing, A., Joseph, D., Yue, D.: The return of the SDitH. Cryptology ePrint Archive (2022)
5. Aragon, N., Bardet, M., Bidoux, L., Chi-Domínguez, J.J., Dyseryn, V., Feneuil, T., Gaborit, P., Joux, A., Rivain, M., Tillich, J.P., Vinçotte, A.: RYDE: Rank decoding in-the-head. Submission to the NIST Post-Quantum Standardization project (2023)
6. Aragon, N., Bardet, M., Bidoux, L., Chi-Domínguez, J.J., Dyseryn, V., Feneuil, T., Gaborit, P., Neveu, R., Rivain, M., Tillich, J.P.: MIRA: MinRank in-the-head. Submission to the NIST Post-Quantum Standardization project (2023)
7. Attema, T., Cramer, R., Kohl, L.: A compressed σ-protocol theory for lattices. In: Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II. pp. 549–579. Springer (2021)
8. Attema, T., Fehr, S., Klooß, M.: Fiat-Shamir transformation of multi-round interactive proofs. In: Theory of Cryptography: 20th International Conference, TCC 2022, Chicago, IL, USA, November 7–10, 2022, Proceedings, Part I. pp. 113–142. Springer (2022)
9. Baldi, M., Barenghi, A., Beckwith, L., Biasse, J.F., Esser, A., Gaj, K., Mohajerani, K., Pelosi, G., Persichetti, E., Saarinen, M.J.O., Santini, P., Wallace, R.: LESS: Linear equivalence signature scheme. Submission to the NIST Post-Quantum Standardization project (2023)
10. Baldi, M., Barenghi, A., Bitzer, S., Karl, P., Manganiello, F., Pavoni, A., Pelosi, G., Santini, P., Schupp, J., Slaughter, F., Wachter-Zeh, Antonia Weger, V.: CROSS: Codes and restricted objects signature scheme. Submission to the NIST Post-Quantum Standardization project (2023)
11. Baldi, M., Battaglioni, M., Chiaraluce, F., Horlemann-Trautmann, A.L., Persichetti, E., Santini, P., Weger, V.: A new path to code-based signatures via identification schemes with restricted errors. arXiv preprint arXiv:2008.06403 (2020)
12. Banegas, G., Carrier, K., Chailloux, A., Couvreur, A., Debris-Alazard, T., Gaborit, P., Karpman, P., Loyer, J., Niederhagen, R., Sendrier, N., Smith, B., Tillich, J.P.: WAVE. Submission to the NIST Post-Quantum Standardization project (2023)
13. Barg, S.: Some new NP-complete coding problems. Problemy Peredachi Informatsii **30**(3), 23–28 (1994)
14. Becker, A., Coron, J.S., Joux, A.: Improved generic algorithms for hard knapsacks. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 364–385. Springer (2011)
15. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31. pp. 520–536. Springer (2012)
16. Berlekamp, E., McEliece, R., Van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory **24**(3), 384–386 (1978)

17. Beullens, W.: Sigma protocols for MQ, PKP and SIS, and fishy signature schemes. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 183–211. Springer (2020)
18. Bidoux, L., Gaborit, P.: Shorter Signatures from Proofs of Knowledge for the SD, MQ, PKP and RSD Problems. arXiv preprint arXiv:2204.02915 (2022)
19. Carrier, K., Debris-Alazard, T., Meyer-Hilfiger, C., Tillich, J.P.: Statistical decoding 2.0: reducing decoding to lpn. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 477–507. Springer (2022)
20. Cayrel, P.L., Véron, P., Yousfi Alaoui, S.M.E.: A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In: International Workshop on Selected Areas in Cryptography. pp. 171–186. Springer (2010)
21. Chailloux, A., Etinski, S.: On the (in) security of optimized stern-like signature schemes. Designs, Codes and Cryptography (2023)
22. Cho, J., No, J.S., Lee, Y., Kim, Y.S., Koo, Z.: Enhanced pqsigRM. Submission to the NIST Post-Quantum Standardization project (2023)
23. Chou, T., Niederhagen, R., Persichetti, E., Ran, L., Hajatiana Randrianarisoa, T., Reijnders, K., Samardjiska, S., Trimoska, M.: MEDS: Matrix equivalence digital signature. Submission to the NIST Post-Quantum Standardization project (2023)
24. Debris-Alazard, T., Sendrier, N., Tillich, J.P.: Wave: A new code-based signature scheme. In: Asiacrypt 2019 (2019)
25. Debris-Alazard, T., Tillich, J.P.: Statistical decoding. In: 2017 IEEE International Symposium on Information Theory (ISIT). pp. 1798–1802. IEEE (2017)
26. Dumer, I.I.: Two decoding algorithms for linear codes. Problemy Peredachi Informatsii **25**(1), 24–32 (1989)
27. Feneuil, T.: Building MPCitH-based signatures from MQ, MinRank, Rank SD and PKP. Cryptology ePrint Archive (2022)
28. Feneuil, T., Joux, A., Rivain, M.: Shared permutation for syndrome decoding: New zero-knowledge protocol and code-based signature. Designs, Codes and Cryptography pp. 1–46 (2022)
29. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: shorter signatures from zero-knowledge proofs. In: Annual International Cryptology Conference. pp. 541–572. Springer (2022)
30. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Crypto. vol. 86, pp. 186–194. Springer (1986)
31. Gueron, S., Persichetti, E., Santini, P.: Designing a practical code-based signature scheme from zero-knowledge proofs with trusted setup. Cryptography **6**(1), 5 (2022)
32. Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass MQ-based identification to MQ-based signatures. IACR Cryptol. ePrint Arch. **2016**, 708 (2016)
33. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. In: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing. pp. 21–30 (2007)
34. Jabri, A.A.: A statistical decoding algorithm for general linear block codes. In: Cryptography and Coding: 8th IMA International Conference Cirencester, UK, December 17–19, 2001 Proceedings 8. pp. 1–8. Springer (2001)
35. Kales, D., Zaverucha, G.: An attack on some signature schemes constructed from five-pass identification schemes. In: Cryptology and Network Security: 19th International Conference, CANS 2020, Vienna, Austria, December 14–16, 2020, Proceedings. pp. 3–22. Springer (2020)

36. Meurer, A.: A coding-theoretic approach to cryptanalysis. Ph.D. thesis, Ruhr-Universität Bochum (2013)
37. Ritterhoff, S., Maringer, G., Bitzer, S., Weger, V., Karl, P., Schamberger, T., Schupp, J., Wachter-Zeh, A.: FuLeeca: A Lee-based Signature Scheme. Submission to the NIST Post-Quantum Standardization project (2023)
38. Santini, P., Baldi, M., Chiaraluce, F.: Computational hardness of the permuted kernel and subcode equivalence problems. Cryptology ePrint Archive (2022)
39. Shamir, A.: An efficient identification scheme based on permuted kernels. In: Brassard, G. (ed.) Advances in Cryptology — CRYPTO' 89 Proceedings. CRYPTO 1989. Lecture Notes in Computer Science, vol. 435, pp. 606–609. Springer (1989)
40. Stern, J.: A method for finding codewords of small weight. In: International Colloquium on Coding Theory and Applications. pp. 106–113. Springer (1988)
41. Stern, J.: A new identification scheme based on syndrome decoding. In: Annual International Cryptology Conference. pp. 13–21. Springer (1993)
42. Stern, J.: Designing identification schemes with keys of short size. In: Advances in Cryptology—CRYPTO'94: 14th Annual International Cryptology Conference Santa Barbara, California, USA August 21–25, 1994 Proceedings. pp. 164–173. Springer (2001)
43. Weger, V., Khathuria, K., Horlemann, A.L., Battaglioni, M., Santini, P., Persichetti, E.: On the hardness of the Lee syndrome decoding problem. Advances in Mathematics of Communications (2022)

A Appendix: NP-hardness Proof for R-SDP

Problem 1. Let $\mathbb{E} = \{g^i \mid i \in \{0, \dots, z-1\}\}$, for some $z \in \mathbb{F}_q$ of order $1 < z < q-1$. Given $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_q^{n-k}$, is there a $\mathbf{e} \in \mathbb{E}^n$ such that $\mathbf{e}\mathbf{H}^\top = \mathbf{s}$?

Theorem 3. *The R-SDP (Problem 1) for $n < q$ is NP-complete.*

Proof. Recall the NP-hard 3-Dimensional Matching (3DM) problem, where one is given the instance $T = \{b_1, \dots, b_t\}$, with $|T| = w, U \subset T \times T \times T$ and $|U| = u$ and asks whether there exists a $W \subset U$ with $|W| = w$ and no two words in W coincide in any position.

Recall that the original SDP has a reduction from 3DM, through the following construction: let $\mathbf{H} \in \mathbb{F}_q^{3w\times u}$ be the incidence matrix, i.e., each column of $\mathbf{H}$ corresponds to a word in U and the rows correspond to $T \times T \times T$, thus the rows $\{1, \dots, w\}$ correspond to the first position of the word $\mathbf{u}$, the rows $\{w+1, \dots, 2w\}$ correspond to the second position of $\mathbf{u}$ and the rows $\{2w+1, \dots, 3w\}$ correspond to the third position of $\mathbf{u}$. More formally, let $T = \{b_1, \dots, b_w\}$, $U = \{\mathbf{a}_1, \dots, \mathbf{a}_u\}$ and

- for $i \in \{1, \dots, w\}$, we set $h_{i,j} = 1$ if $\mathbf{a}_j[1] = b_i$ and $h_{i,j} = 0$ else,
- for $i \in \{w+1, \dots, 2w\}$, we set $h_{i,j} = 1$ if $\mathbf{a}_j[2] = b_i$ and $h_{i,j} = 0$ else,
- for $i \in \{2w+1, \dots, 3w\}$, we set $h_{i,j} = 1$ if $\mathbf{a}_j[3] = b_i$ and $h_{i,j} = 0$ else.

We also set $\mathbf{s} \in \mathbb{F}_q^{3w}$ be the all one vector. From the original reduction, we know that any solution $\mathbf{e} \in \mathbb{F}_q^u$ with $\mathbf{H}\mathbf{e}^\top = \mathbf{s}^\top$ has weight w (it has weight at least w as we need to reach the all one vector in $\mathbb{F}_q^{3w}$ and each column gives weight 3, and it has weight at most w as q is larger than u and else we would get syndrome entries larger than 1) and its support corresponds to the solution W. That is the columns of $\mathbf{H}$ indexed by the support of $\mathbf{e}$ are the w words in W.

The polynomial reduction from 3DM to R-SDP uses this construction as well. Let T of size w and $U \subset T \times T \times T$ of size u be an instance of 3DM. Let $\mathbf{H} \in \mathbb{F}_q^{(3w)\times u}$ be the incidence matrix and let

$$\widetilde{\mathbf{H}} = \begin{pmatrix} \mathbf{H} & -g\mathbf{H} \\ \mathrm{Id}_u & \mathrm{Id}_u \end{pmatrix} \in \mathbb{F}_q^{(3w+u)\times 2u}$$

be a parity-check matrix. Let us consider the syndrome $(\mathbf{s}, \mathbf{s}') \in \mathbb{F}_q^{3w+u}$ with $\mathbf{s} = (1-g^2, \ldots, 1-g^2) \in \mathbb{F}_q^{3w}$ and $\mathbf{s}' = (1+g, \ldots, 1+g) \in \mathbb{F}_q^u$. Thus, the instance of R-SDP given by $\widetilde{\mathbf{H}}$ and $(\mathbf{s}, \mathbf{s}')$ is asking for $(\mathbf{e}, \mathbf{e}') \in \mathbb{E}^{2u}$ such that

$$(\mathbf{e}, \mathbf{e}')\widetilde{\mathbf{H}}^\top = (\mathbf{s}, \mathbf{s}'),$$

where $\mathbb{E} = \{g^i \mid i \in \{0, \ldots, z-1\}\}$. By assumption of R-SDP, we use a g of order $2 < z < q-1$.

We consider two cases.

1. Assume that the R-SDP solver returns "yes", i.e., there exists $\mathbf{e}, \mathbf{e}' \in \mathbb{E}^u$ such that $(\mathbf{e}, \mathbf{e}')\widetilde{\mathbf{H}}^\top = (\mathbf{s}, \mathbf{s}')$. Hence,

$$\begin{aligned} \mathbf{H}\mathbf{e}^\top - g \star \mathbf{H}\mathbf{e}'^\top &= (1-g^2, \ldots, 1-g^2)^\top, \\ \mathbf{e} + \mathbf{e}' &= (1+g, \ldots, 1+g). \end{aligned}$$

   Hence, for each $i \in \{1, \ldots, u\}$ we have $e_i + e'_i = 1 + g$. Let us assume (we later show that this hypothesis is not needed, but it facilitates the proof) that the only elements in $\mathbb{E}$ that add to $1 + g$ is 1 and g.
   Hence, whenever $e_i = 1$, we must have $e'_i = g$ and whenever $e_i = g$, we must have $e'_i = 1$. Thus, we split $\mathbf{e} = \mathbf{e}_1 + \mathbf{e}_g$ and $\mathbf{e}' = \mathbf{e}'_1 + \mathbf{e}'_g$ where $\mathbf{e}_1, \mathbf{e}'_1 \in \{0,1\}^u, \mathbf{e}_g, \mathbf{e}'_g \in \{0, g\}^u$ and $\mathrm{supp}(\mathbf{e}_1) = S = \mathrm{supp}(\mathbf{e}'_g)$ and $\mathrm{supp}(\mathbf{e}'_1) = S^C = \mathrm{supp}(\mathbf{e}_g)$. From this also follows that $\mathbf{e}_g = g \star \mathbf{e}'_1$ and $\mathbf{e}'_g = g \star \mathbf{e}_1$.
   The first parity-check equation can now be reformulated as

$$\begin{aligned} &\mathbf{H}\mathbf{e}^\top - g \star \mathbf{H}\mathbf{e}'^\top \\ =&\mathbf{H}\mathbf{e}_1^\top - g \star \mathbf{H}\mathbf{e}_g'^\top + \mathbf{H}\mathbf{e}_g^\top - g \star \mathbf{H}\mathbf{e}_1'^\top \\ =&\mathbf{H}\mathbf{e}_1^\top - g^2 \star \mathbf{H}\mathbf{e}_1^\top + g \star \mathbf{H}\mathbf{e}_1'^\top - g \star \mathbf{H}\mathbf{e}_1'^\top \\ =&(1-g^2) \star \mathbf{H}\mathbf{e}_1^\top \\ =&(1-g^2, \ldots, 1-g^2) = \mathbf{s}', \end{aligned}$$

   thus, $\mathbf{H}\mathbf{e}_1^\top = (1, \ldots, 1)$ is such that $\mathrm{supp}(\mathbf{e}_1)$ corresponds to a solution W of 3DM, as in the classical reduction.
2. Assume that the R-SDP solver returns "no", i.e., there exists no $\mathbf{e}, \mathbf{e}' \in \mathbb{E}^u$ such that $(\mathbf{e}, \mathbf{e}')\widetilde{\mathbf{H}}^\top = (\mathbf{s}, \mathbf{s}')$. Let us assume by contradiction, that the 3DM has a solution W. We can then define S to be the indices of words in U belonging to the solution W. Let us define $\mathbf{e}_1, \mathbf{e}'_1 \in \{0,1\}^u, \mathbf{e}_g, \mathbf{e}'_g \in \{0, g\}^u$ with $\mathrm{supp}(\mathbf{e}_1) = S = \mathrm{supp}(\mathbf{e}'_g)$ and $\mathrm{supp}(\mathbf{e}'_1) = S^C = \mathrm{supp}(\mathbf{e}_g)$. From this also follows that $\mathbf{e}_g = g \star \mathbf{e}'_1$ and $\mathbf{e}'_g = g \star \mathbf{e}_1$. Then the vector $(\mathbf{e}_1 + \mathbf{e}_g, \mathbf{e}'_1 + \mathbf{e}'_g) \in \mathbb{E}^{2u}$ is a solution to the R-SDP, as in case 1, which gives the desired contradiction, to the R-SDP solver returning "no".

Note that the hypothesis, that only 1 and g in $\mathbb{E}$ add up to $1+g$ is not necessary. For this assume that there exists $g^i, g^j \in \mathbb{E}$, with $0 \neq i < j < z$ such that $g^i + g^j = 1 + g$. Thus, the splitting of $\mathbf{e}$ and $\mathbf{e}'$ is a bit more complicated:

$$\begin{aligned}\mathbf{e} &= \mathbf{e}_1 + \mathbf{e}_g + \mathbf{e}_i + \mathbf{e}_j,\\ \mathbf{e}' &= \mathbf{e}'_1 + \mathbf{e}'_g + \mathbf{e}'_i + \mathbf{e}'_j,\end{aligned}$$

where $\mathbf{e}_1, \mathbf{e}'_1 \in \{0,1\}^u$,$\mathbf{e}_g, \mathbf{e}'_g \in \{0,g\}^u$,$\mathbf{e}_i, \mathbf{e}'_i \in \{0,g^i\}^u$,$\mathbf{e}_j, \mathbf{e}'_j \in \{0,g^j\}^u$ with

$$\begin{aligned}\mathrm{supp}(\mathbf{e}_1) &= S_1 = \mathrm{supp}(\mathbf{e}'_g),\\ \mathrm{supp}(\mathbf{e}_g) &= S'_1 = \mathrm{supp}(\mathbf{e}'_1),\\ \mathrm{supp}(\mathbf{e}_i) &= S_i = \mathrm{supp}(\mathbf{e}'_j),\\ \mathrm{supp}(\mathbf{e}_j) &= S'_i = \mathrm{supp}(\mathbf{e}'_i),\end{aligned}$$

and the supports S_1, S'_1, S_i, S'_i are distinct and partition $\{1, \ldots, u\}$. Again it follows that

$$\begin{aligned}\mathbf{e}_g &= g \star \mathbf{e}'_1,\\ \mathbf{e}'_g &= g \star \mathbf{e}_1,\\ \mathbf{e}_j &= g^{j-i} \star \mathbf{e}'_i,\\ \mathbf{e}'_j &= g^{j-i} \star \mathbf{e}_i.\end{aligned}$$

Thus, rewriting the first parity-check equation, we get

$$\begin{aligned}&\mathbf{H}\mathbf{e}^\top - g \star \mathbf{H}\mathbf{e}'^\top\\ =&\mathbf{H}\mathbf{e}_1^\top + \mathbf{H}\mathbf{e}_g^\top + \mathbf{H}\mathbf{e}_i^\top + \mathbf{H}\mathbf{e}_j^\top\\ &- g \star \mathbf{H}\mathbf{e}_1'^\top - g \star \mathbf{H}\mathbf{e}_g'^\top - g \star \mathbf{H}\mathbf{e}_i'^\top - g \star \mathbf{H}\mathbf{e}_j'^\top\\ =&\mathbf{H}\mathbf{e}_1^\top + g \star \mathbf{H}\mathbf{e}_1'^\top + \mathbf{H}\mathbf{e}_i^\top + g^{j-i} \star \mathbf{H}\mathbf{e}_i'^\top\\ &- g \star \mathbf{H}\mathbf{e}_1'^\top - g^2 \star \mathbf{H}\mathbf{e}_1^\top - g \star \mathbf{H}\mathbf{e}_i'^\top - g^{j-i+1} \star \mathbf{H}\mathbf{e}_i^\top\\ =&(1-g^2) \star \mathbf{H}\mathbf{e}_1^\top + (1-g^{j-i+1}) \star \mathbf{H}\mathbf{e}_i^\top + (g^{j-i}-g) \star \mathbf{H}\mathbf{e}_i'^\top\\ =&(1-g^2, \ldots, 1-g^2) = \mathbf{s}'.\end{aligned}$$

Since $\mathbf{e}_1, \mathbf{e}_i, \mathbf{e}'_i$ all have different supports, the only way to get $1-g^2$ in each entry, is to have $\mathbf{e}_i = \mathbf{e}'_i = 0$. In fact, any other sum leads to a contradiction:

- If $(1-g^2) + (1-g^{j-i+1}) = 1-g^2$ then $1 = g^{j-i+1}$ and hence $j = i-1$ which contradicts $j > i$.
- If $(1-g^2) + (g^{j-i}-g) = 1-g^2$ then $g^{j-i} = g$ and hence $j - i = 1$. However, as then $g^j + g^i = g^i(1+g) = 1+g$, it follows that $g^i = 1$, which contradicts $i \neq 0$.
- If $(1-g^2)+(1-g^{j-i+})+(g^{j-i}-g) = 1-g^2$, then $1+g^{j-i} = g^{j-i+1}+g = g(1+g^{j-i})$ and thus $g = 1$, which contradicts $\mathbb{E} \neq \mathbb{F}_q^\star$.
- If $(1-g^{j-i+1}) + (g^{j-i}-g) = 1-g^2$, then $g^{j-i} - g^{j-i+1} = g - g^2$ and hence $g^{j-i}(1-g) = g(1-g)$ and thus $j - i = 1$, which is a contradiction again as in the second case.

□

B Appendix: Representation Technique for R-SDP

Note that algebraic attacks that try to exploit the small order of the entries of $\mathbf{e}$ cannot be mounted in a straightforward manner. This is mainly due to the fact that $\mathbb{E}$ is a multiplicative group and thus not compatible with the additive linearity of the syndrome computation. The situation would be different if the restricted error vectors would be additive, i.e., any $\mathbf{e} = \sum_{i=1}^{m} \lambda_i \mathbf{x}_i$, for some publicly known $\mathbf{x}_i \in \mathbb{F}_q^n$ and unknown $\lambda_i \in \mathbb{F}_q$. In this case, one can write the syndrome equation as a linear system and solve it in polynomial time. As our choice of $\mathbb{E}$ is however a multiplicative one, such an approach is not possible. In this Appendix, we present a generic solver for the R-SDP, which is an adaption of the BJMM algorithm [15] in combination with the technique of [14] for subset sum solvers. The security levels provided in this paper are computed taking also this algorithm into account.

For this section, we require some additional notation. For $n \geq \sum_{i=1}^{m} k_i$ we denote by

$$\binom{n}{k_1, \ldots, k_m} = \prod_{i=1}^{m} \binom{\sum_{j=1}^{i} k_j}{k_i} \binom{n}{n - \sum_{i=1}^{m} k_i}$$

the multinomial coefficient. Recall that

$$\lim_{n\to\infty} \frac{1}{n} \log_2 \left(\binom{f(n)}{f_1(n), \ldots, f_m(n)} \right) = F \cdot g_m \left(\tfrac{F_1}{F}, \ldots, \tfrac{F_m}{F} \right),$$

$$\text{with } g_m(x_1, \ldots, x_m) = -\sum_{i=1}^{m} x_i \log_2(x_i) - \left(1 - \sum_{i=1}^{m} x_i\right) \log_2 \left(1 - \sum_{i=1}^{m} x_i\right)$$

and $F = \lim_{n\to\infty} \frac{f(n)}{n}, F_i = \lim_{n\to\infty} \frac{f_i(n)}{n}$ for all $i \in \{1, \ldots, m\}$. Notice that $g_1 = h_2$ corresponds to the binary entropy function.

After the PGE step, explained in Section 3, we are left with solving the smaller instance, i.e., $\mathbf{e}_1 \mathbf{H}_2^\top = \mathbf{s}_2$ and $\mathbf{e}_1 \in \mathbb{E}_0^{k+\ell}$ has weight v. The main idea of the BJMM algorithm is to use a sum partition $\mathbf{e}_1 = \mathbf{e}_1^{(1)} + \mathbf{e}_2^{(1)}$. The number of ways in which we can write a vector $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$, where both the $\mathbf{x}_i$ have to satisfy certain conditions, is called the *number of representations*.

We start with the *representation merge*: given two lists $\mathscr{L}_1, \mathscr{L}_2$ containing $\mathbf{x}_i$ of a certain weight, we add $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$ to the resulting list $\mathscr{L}$, whenever $\mathbf{x}$ attains some target weight and some syndrome equations are satisfied. These are $\mathbf{x}\mathbf{H}_2^\top = \mathbf{t}$, for either $\mathbf{t} = \mathbf{s}_2$, the target syndrome or $\mathbf{t} = \mathbf{0}$, the zero vector. Assume that for any $\mathbf{x} \in \mathscr{L}$ there are r representations $\{\mathbf{x}_1, \mathbf{x}_2\}$, which lead to the same $\mathbf{x}$. By checking the syndrome equations only on $u = \log_q(r)$ positions, we have with high probability that one representation for each possible $\mathbf{x}$ survives the merge. A representation merge of two lists $\mathscr{L}_1, \mathscr{L}_2$ on u positions costs

$$|\mathscr{L}_1| + |\mathscr{L}_2| + |\mathscr{L}_1| \cdot |\mathscr{L}_2|\, q^{-u}.$$

After the representation merge, one performs a filtering step, which removes vectors that are not well-formed, e.g., do not achieve a given weight constraint or do not live in a desired space. Further steps can then utilize this smaller list.

The representation merge can clearly be used several times, thus we denote by BJMM(a) an algorithm that has a *levels*, where in the first level we do a concatenation merge à la Stern/Dumer. For more details on how exactly the algorithm proceeds, we refer the reader to [36].

Since we have many non-zero entries in our solution, we want many representations of elements in $\mathbb{E}$. For this we have to choose the *search space*, i.e., where $\mathbf{e}_1^{(1)}, \mathbf{e}_2^{(1)}$ live, in a smart way: we want to choose it large enough to gain representations, but small enough to have reasonable list sizes.

To get some fixed entry $x \in \mathbb{E}$ as $x = y + y'$, we could choose $y \in \mathbb{E}, y' \in \mathbb{D} \coloneqq \{a - b \mid a, b \in \mathbb{E}\} \setminus \{\pm\mathbb{E}_0\}$. If $\mathbb{E}$ already has a lot of additive structure, e.g. when z is even, that is there are many elements $y, y' \in \mathbb{E}$ such that $y + y' \in \mathbb{E}$, then $\mathbb{D}$ becomes small. Thus, we only need a few additional elements in the search space to gain many representations for elements in $\mathbb{E}$. We propose the following search space $X = \mathbb{E} \cup \mathbb{D} \cup -\mathbb{E}$. On each level i, we are considering vectors $\mathbf{x}$ living in X_0, with $v_e^{(i)}$ entries in $\mathbb{E}$, $v_d^{(i)}$ entries in $\mathbb{D}$ and $v_m^{(i)}$ entries in $-\mathbb{E}$.

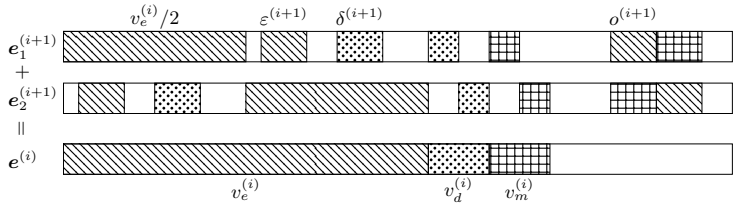

Fig. 10: Counting the number of representations on level i.

To count the number of representations we use Figure 10. We denote by $\varepsilon^{(i+1)}$ the number of entries which are obtained through a $\mathbb{E} + \mathbb{E}$ representation. That is, for a fixed entry x of $\mathbf{e}^{(i)}$, we need to compute the number of possible $y \in \mathbb{E}$ that can reach x, through addition with $\mathbb{E}$:

$$n_e(q, z, x) \coloneqq \left|\{y \in \mathbb{E} \mid \exists y' \in \mathbb{E} : y + y' = x \in \mathbb{E}\}\right|.$$

We denote by $\delta^{(i+1)}$ the number of entries of $\mathbf{e}^{(i)}$ obtained through representations $\mathbb{E} + \mathbb{D}$. Hence, for a fixed entry x of $\mathbf{e}^{(i)}$, we need to compute the number of possible $y \in \mathbb{E}$ that can reach x through addition with $\mathbb{D}$:

$$n_d(q, z, x) \coloneqq \left|\{y \in \mathbb{E} \mid \exists y' \in \mathbb{D} : y + y' = x \in \mathbb{E}\}\right|.$$

Since $n_e(q, z, x), n_d(q, z, x)$ are independent of x, we just write $n_e(q, z), n_d(q, z)$. Finally, outside of the support of $\mathbf{e}^{(i)}$, we allow for $o^{(i+1)}$ representations of 0 as $0 = y + (-y) = (-y) + y$, for $y \in \mathbb{E}$. We could also allow for cancellations via $\mathbb{D}$, but as these entries are already only few, they will be optimized to zero.

The vectors $\mathbf{e}_i^{(i+1)}$ have $v_e^{(i+1)} = v_e^{(i)}/2 + \varepsilon^{(i+1)} + o^{(i+1)}$ entries in $\mathbb{E}$, $v_d^{(i+1)} = v_d^{(i)}/2 + \delta^{(i+1)}$ in $\mathbb{D}$ and $v_m^{(i+1)} = v_m^{(i)}/2 + o^{(i+1)}$ in $-\mathbb{E}$. Hence, we get the number of representations

$$\begin{aligned} r^{(i)} = &\binom{v_e^{(i-1)}}{v_e^{(i-1)}/2} \left(\binom{v_e^{(i-1)}/2}{\delta^{(i)}, \varepsilon^{(i)}} n_d(q, z)^{\delta^{(i)}} n_e(q, z)^{\varepsilon^{(i)}} \right)^2 \\ &\cdot \binom{v_d^{(i-1)}}{v_d^{(i-1)}/2} \binom{v_m^{(i-1)}}{v_m^{(i-1)}/2} \binom{k + \ell - v_e^{(i-1)} - v_d^{(i-1)} - v_m^{(i-1)}}{o^{(i)}, o^{(i)}} z^{2o^{(i)}}. \end{aligned} \tag{10}$$

After each merge, the obtained lists are filtered, to get rid of vectors that are not well-formed. After the filtering we are considering vectors in $S^{(i)}$ that have $v_e^{(i)}$ entries in $\mathbb{E}$, $v_d^{(i)}$ entries in $\mathbb{D}$ and $v_m^{(i)}$ entries in $-\mathbb{E}$. Hence,

$$\left|S^{(i)}\right| = \binom{k+\ell}{v_e^{(i)}, v_m^{(i)}, v_d^{(i)}} z^{v_e^{(i)}+v_m^{(i)}} |\mathbb{D}|^{v_d^{(i)}}.$$

To give the asymptotic cost, we need the following notation:

$$\begin{array}{lll}
Q = \log_2(q) & V_e^{(i)} = \lim_{n\to\infty} \frac{v_e^{(i)}(n)}{n} & N_e = \lim_{n\to\infty} \frac{1}{n}\log_2(n_e(q,z)) \\
Z = \log_2(z) & V_m^{(i)} = \lim_{n\to\infty} \frac{v_m^{(i)}(n)}{n} & N_d = \lim_{n\to\infty} \frac{1}{n}\log_2(n_d(q,z)) \\
L = \lim_{n\to\infty} \frac{\ell(n)}{n} & V_d^{(i)} = \lim_{n\to\infty} \frac{v_d^{(i)}(n)}{n} & \Sigma^{(i)} = \lim_{n\to\infty} \frac{1}{n}\log_2\left(\left|S^{(i)}\right|\right) \\
U^{(i)} = Q \lim_{n\to\infty} \frac{u^{(i)}(n)}{n} & \Delta = \lim_{n\to\infty} \frac{1}{n}\log_2(|\mathbb{D}|) & \\
D^{(i)} = \lim_{n\to\infty} \frac{\delta^{(i)}(n)}{n} & E^{(i)} = \lim_{n\to\infty} \frac{\varepsilon^{(i)}(n)}{n} & O^{(i)} = \lim_{n\to\infty} \frac{o^{(i)}(n)}{n}
\end{array}$$

Theorem 4. *The presented BJMM(3) algorithm has a cost of* $2^{nF(R,q,z,\omega)}$*, where*

$$F(R,q,z,W) = N(R,q,z,W) + C(R,q,z,W),$$

where $N(R,q,z,W)$ *denotes the number of iterations and is given by*

$$h_2(W) - (R+L)h_2\left(\tfrac{V}{R+L}\right) - (1-R-L)h_2\left(\tfrac{W-V}{1-R-L}\right)$$

and $C(R,q,z,W)$ *denotes the cost of one iteration, which is given by*

$$\max\left\{\Sigma^{(2)}/2, \Sigma^{(2)} - U^{(2)}, 2\Sigma^{(2)} - U^{(2)} - U^{(1)}, 2\Sigma^{(1)} - U^{(1)} - LQ\right\},$$

where for $i \in \{1,2\}$ *and* $V_e^{(0)} = V$*,* $V_d^{(0)} = V_m^{(0)} = 0$ *we set*

$$\begin{aligned}
U^{(i)} &= R + L - R^{(i-1)} + R^{(i-1)}h_2\left(\tfrac{2O^{(i)}}{R^{(i-1)}}\right) + O^{(i)} \\
&\quad + V_e^{(i-1)} g_2\left(\tfrac{2E^{(i)}}{V_e^{(i-1)}}, \tfrac{2D^{(i)}}{V_e^{(i-1)}}\right) + 2\left(D^{(i)}N_d + E^{(i)}N_e + O^{(i)}Z\right), \\
\Sigma^{(i)} &= (R+L)g_3\left(\tfrac{V_e^{(i)}}{R+L}, \tfrac{V_m^{(i)}}{R+L}, \tfrac{V_d^{(i)}}{R+L}\right) + \left(V_e^{(i)} + V_m^{(i)}\right)Z + V_d^{(i)}\Delta, \\
R^{(i)} &= R + L - V_e^{(i)} - V_d^{(i)} - V_m^{(i)}, \\
V_e^{(i)} &= V_e^{(i-1)}/2 + E^{(i)} + O^{(i)},\ V_d^{(i)} = V_d^{(i-1)}/2 + D^{(i)},\ V_m^{(i)} = V_m^{(i-1)}/2 + O^{(i)}.
\end{aligned}$$

B.1 Refinements

For large-weight vectors, it makes sense to first shift the considered instance. That is for a fixed $c \in \mathbb{F}_q$, we shift the whole error set $\mathbb{E}$ to $\widetilde{\mathbb{E}} = \{a+c \mid a \in \mathbb{E}\}$. Let us denote by $\mathbf{c}$ the all-c vector. Then, such shifting can easily be done by computing the syndrome

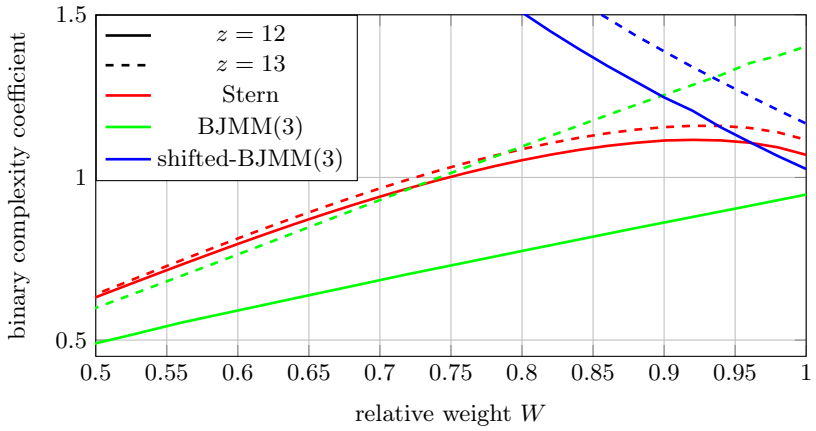

Fig. 11: Comparison of the asymptotic complexity for the restricted Stern/Dumer algorithm and the restricted BJMM algorithm.

$\mathbf{s}_c$ of $\mathbf{c}$ and adding it to the original syndrome $\mathbf{s}$: $(\mathbf{e}+\mathbf{c})\mathbf{H}^\top = \mathbf{s} + \mathbf{s}_c$. By choosing $c \in -\mathbb{E}$, one can set the error at all positions with value c to zero. Hence, one obtains $\widetilde{\mathbb{E}} = \{a + c \mid a \in \mathbb{E}\} \setminus \{0\}$ of size $\widetilde{z} = z - 1$. With this error set of reduced size, one can proceed as before. That is we again use the sets

$$\widetilde{\mathbb{D}} = \left\{a - b \mid a, b \in \widetilde{\mathbb{E}}\right\} \setminus \left\{\pm\widetilde{\mathbb{E}}_0\right\} \text{ and } -\widetilde{\mathbb{E}} = \left\{-e \mid e \in \widetilde{\mathbb{E}}\right\} \setminus \widetilde{\mathbb{E}}.$$

Note that for these sets, $n_e(q, z, x)$ and $n_d(q, z, x)$ are indeed dependent on the element x. In order to avoid a more complicated analysis, we resolve this issue by defining the *average* number of representations for an element in $\widetilde{\mathbb{E}}$ as

$$\widetilde{n_e}(q, z, c) = \frac{1}{\widetilde{z}} \sum_{x \in \widetilde{\mathbb{E}}} n_e(q, z, x) \quad \text{and} \quad \widetilde{n_d}(q, z, c) = \frac{1}{\widetilde{z}} \sum_{x \in \widetilde{\mathbb{E}}} n_d(q, z, x),$$

which depends not on the particular element but only on the chosen shift. Hence, $\widetilde{n_e}$ and $\widetilde{n_d}$ can be directly used in Theorem 4. In Figure 11, we compare the complexity coefficients of different information set decoders as a function of the relative error weight W[9]. The considered code rate is $R = 0.45$. The field size $q = 157$ allows for $z = 12$ and $z = 13$, which correspond to the solid and dashed lines, respectively. While the performance of Stern depends only on the size of $\mathbb{E}$, the performance of the BJMM algorithms depends on its structure. For $z = 12$, $\mathbb{E}$ possesses a lot of additive structure, which is why BJMM(3) can improve over Stern. In particular, $\mathbb{E} = -\mathbb{E}$ and $n_e(157, 12) = 2$ allow for an increased number of representations. This is not the case for $z = 13$, where we only improve over Stern in the low-error-weight regime. Finally, we observe that shifting has to be taken into account for high error weights, but becomes quickly impractical as the weight decreases. Taking these observations into account, we avoid choosing instances for which the BJMM algorithm can achieve a significantly lower complexity than restricted Stern.

[9] Code for recreating the figure is available at `https://github.com/secomms/RBG`.

C Appendix: a Sketch of MPCitH based on R-SDP

We quickly recall the general idea of [29] and, for the sake of simplicity, omit several technical details. The secret $\mathbf{e}$ is split into N shares $\mathbf{e}^{(i)}$, which are such that $\sum_{i=1}^{N} \mathbf{e}^{(i)} = \mathbf{e}$ and for each share $\mathbf{e}^{(i)}$ one computes the broadcast $\alpha_i = \mathbf{e}^{(i)}\mathbf{H}^\top$. Additionally, the simulated N parties also check that the shares $\mathbf{e}^{(i)}$ sum to a weight-w vector. The broadcast as well as a commitment c_i to each share are then sent to the verifier. A verifier can challenge any share ℓ and upon this challenge the prover will open all shares $\mathbf{e}^{(i)}$ for $i \neq \ell$. The verifier can then check the commitments c_i and broadcasts α_i for $i \neq \ell$. This changes the soundness error to $\frac{1}{N}$. However, the computation of the broadcasts has still to be done N times during signing as well as $N-1$ times during verification, which makes this approach very slow. As the parties should also check for the correct weight of the secret, these approaches are also highly complex and difficult to implement.

For the latter, the authors of [29] propose an MPC protocol which is based on polynomial relations. The idea is to construct a polynomial whose degree is the same as the weight of $\mathbf{e}$: the degree verification demands an ad-hoc MPC protocol.

For the case of R-SDP, using just the degree verification will not be enough. Indeed, the MPC parties should also verify that $\mathbf{e}$ has only restricted entries. So, converting the SDItH protocol to the use of R-SDP seems inappropriate. However, the MPCitH protocol employed in [27, Section 6] for PKP should better fit our scopes. Indeed, the author proposes an MPC protocol to verify that the polynomials

$$P(x) = \prod_{j=1}^{n}(x - \mathbf{e}^{(j)}), \quad P'(x) = \prod_{j=1}^{n}(x - \mathbf{e}^{(j)'}), \tag{11}$$

for which the parties receive additive shares, have the same roots. A similar idea may be used also for R-SDP with maximum Hamming weight. In our case, all the roots of $P(x)$ (which would be the sharing of the secret restricted vector $\mathbf{e}$), as well as those of $P'(x)$ (which would be the sharing of $\mathbf{e}' = \sigma(\mathbf{e})$) live in $\mathbb{E}$: this is what we can demand the MPC protocol to verify. Since the protocol also checks that $P(0) \neq 0$, this will convince the parties that $\mathbf{e}$ does not have zero entries and, consequently, has maximum Hamming weight.

D Appendix: Examples of subgroups of $\mathbb{E}^n$

Example 1 Let $q = 13$ and $g = 5$, with multiplicative order $z = 4$; consequently

$$\mathbb{E} = \left\{1 = g^0,\ 5 = g^1,\ 12 = g^2,\ 8 = g^3\right\}.$$

Let us consider $n = 5$ and $m = 3$. As generating set for G, we take

$$\mathbf{x}_1 = (12, 5, 5, 5, 12), \quad \mathbf{x}_2 = (12, 1, 5, 5, 1), \quad \mathbf{x}_3 = (8, 12, 1, 1, 1)$$
$$\text{with } \ell(\mathbf{x}_1) = (2, 1, 1, 1, 2), \quad \ell(\mathbf{x}_2) = (2, 0, 1, 1, 0), \quad \ell(\mathbf{x}_3) = (3, 2, 0, 0, 0).$$

Each of these vectors has maximum order z and one can check that $|G| = |\mathscr{B}|$ is maximal, i.e., $z^m = 4^3 = 64$. Each vector in G is associated with a length-3 vector over $\mathbb{Z}_4$. For instance, to $(1, 3, 0)$ we associate the vector

$$\begin{aligned}\mathbf{a} &= \ell_G^{-1}\big((1,3,0)\big) = \mathbf{x}_1^1 \star \mathbf{x}_2^3 \star \mathbf{x}_2^0 \\ &= \big(g^2, g^1, g^1, g^1, g^2\big) \star \big(g^2, g^0, g^3, g^3, g^0\big) \star \big(g^0, g^0, g^0, g^0, g^0\big) \\ &= \big(g^0, g^1, g^0, g^0, g^2\big) = \big(1, 5, 1, 1, 12\big).\end{aligned}$$

Example 2 Let $q = 13$, $n = 5$, $m = 4$ and $g = 3$, with multiplicative order 3. We consider the group G whose generating set contains the vectors with $\ell(\mathbf{x}_i)$ being

$$(2,0,2,0,2), \quad (2,2,0,2,2), \quad (0,2,2,1,1), \quad (1,2,2,2,2).$$

The resulting $\mathbf{M}_G$ has full rank, i.e., 4, so $\mathscr{B}$ and G contain $3^4 = 81$ elements.

Example 3 Let us consider the case of $q = 11$ and $g = 3$, having order $z = 5$. Let $n = 10$ and $m = 3$, and assume that the considered group G is such that

$$\mathbf{M}_G = \begin{pmatrix} 1\,4\,3\,4\,2\,3\,2\,0\,1\,0 \\ 0\,2\,1\,0\,1\,2\,4\,2\,3\,3 \\ 1\,3\,3\,4\,3\,0\,1\,4\,4\,3 \end{pmatrix},$$

with rank 3. Hence, the group G has maximum order 5^3. Let $t = 4$ and consider $T = \{1,2,3,4\}$. The columns of $\mathbf{M}_G$ which are indexed by T form a matrix $\mathbf{M}'$ with the three linearly independent rows. Consequently, $m' = \mathrm{rk}(\mathbf{M}') = 3$ and to enumerate all candidates for $\mathbf{e}_T$ it is enough to enumerate all the exponents vectors which can be generated by linear combinations of the rows of $\mathbf{M}'$. Instead of $z^t = 5^4$, we can enumerate all candidates for $\mathbf{e}_T$ in time $z^{m'} = 5^3$. However, if $T = \{4,6,7,9\}$, then the corresponding columns form a matrix $\mathbf{M}'$ with rank 2. Thus, we can enumerate all candidates for $\mathbf{e}_T$ in time $z^{m'} = 5^2$.

E Appendix: The R-BG signature scheme

In this section, we prove the soundness, special-soundness, completeness, and zero-knowledge properties of the proposed R-BG scheme. For this, we closely follow the proofs given in [18], as all the properties follow directly from the original BG protocol.

Proposition 3. *The R-BG scheme presented in Figure 5 is complete.*

Proof. The completeness follows from the protocol description, once it is observed that

$$\tilde{\mathbf{e}}_N = \sigma(\beta\mathbf{e}) + \mathbf{v}$$

and thus

$$\tilde{\mathbf{s}} = \tilde{\mathbf{e}}_N\mathbf{H}^\top - \beta\mathbf{s} = \sigma(\beta\mathbf{e})\mathbf{H}^\top - \beta\sigma(\mathbf{e})\mathbf{H}^\top + \mathbf{v}\mathbf{H}^\top = \mathbf{v}\mathbf{H}^\top.$$

□

Proposition 4. *The R-BG scheme presented in Figure 5 is* $(2,2)$ *out of* $(q-1,N)$*-special sound.*

Proof. For this, we need to build an efficient extractor of knowledge Ext, which returns a solution of the R-SDP(G) on the instance $\mathbf{H}, \mathbf{s}, \mathbf{e}, G$ with high probability, given a $(q-1,N)$-tree of accepting transcripts. For this we require only four leafs of the tree, corresponding to $(\beta,i),(\beta,i'),(\beta',i)$ and (β',i') with $\beta \neq \beta'$ and $i \neq i'$. The extractor Ext computes the solution as follows. First, by taking the responses to the challenges (β,i) and (β,i'), one can generate σ_j for all $\ell \in \{0,\ldots,N\}$ and thus also $\sigma = \sigma_N \circ \cdots \circ \sigma_2 \circ \sigma_1$.

Let us denote by $\tilde{\mathbf{e}}_j$ the vectors for transcripts with challenge β and by $\tilde{\mathbf{e}}'_j$ the vectors for transcripts with challenge β'.

Using the responses corresponding to the challenges (β, i) and (β', i) we can also generate all $(\sigma_j, \mathbf{v}_j)$ and thus also the commitments c_j. Due to the binding property, the commitments do not depend on β or β', i.e., generating $(\sigma'_j, \mathbf{v}'_j)$ from (β', i) and (β', i') the commitments must be the same.

Using the transcripts corresponding to the challenges (β, i) and (β', i) we can compute all $\tilde{\mathbf{e}}_j, \tilde{\mathbf{e}}'_j$ for all $\ell \in \{0, \ldots, N\}$.

By construction we have $\tilde{\mathbf{e}}_0 = \beta\mathbf{e}$ is same for the transcripts from (β, i) and (β, i') as well as $\tilde{\mathbf{e}}'_0 = \beta'\mathbf{e}$ is same for the transcripts from (β', i) and (β', i'), the same holds for the remaining $\tilde{\mathbf{e}}_j$ and $\tilde{\mathbf{e}}'_j$.

Hence, we get $\tilde{\mathbf{e}}_N\mathbf{H}^\top - \beta\mathbf{s} = \tilde{\mathbf{e}}'_N\mathbf{H}^\top - \beta'\mathbf{s}$. From the binding property of the commitments, $(\sigma(\beta\mathbf{e}) + \mathbf{v})\mathbf{H}^\top - \beta\mathbf{s} = (\sigma(\beta'\mathbf{e}) + \mathbf{v})\mathbf{H}^\top - \beta'\mathbf{s}$. It follows that $(\beta - \beta')\sigma(\mathbf{e})\mathbf{H}^\top = (\beta - \beta')\mathbf{s}$. With this we get that $\sigma(\mathbf{e}) \in G$ and $\sigma(\mathbf{e})\mathbf{H}^\top = \mathbf{s}$. Thus, $\sigma(\mathbf{e})$ is a solution to R-SDP on the instance $\mathbf{H}, \mathbf{s}, \mathbf{e}, G$. □

Proposition 5. *The R-BG scheme presented in Figure 5 is zero-knowledge.*

Proof. A valid transcript upon the challenge (β, i) consists of $(c, \beta, h, i, \mathsf{Rsp}_i)$, where the response $\mathsf{Rsp}_i = (c_i, \tilde{\mathbf{e}}_i, \mathsf{Seed}_j)$ for $j \neq i$ and for $i \neq 1$ also includes σ_1. In both cases the verifier has enough information to recover all commitments c_j, but does not know σ_i for the challenge i.

Thus, having all σ_j for $\ell \neq i$, one cannot recover σ from $\sigma_1 = \sigma_2^{-1} \circ \cdots \circ \sigma_N$.

We can prove ZK by building a PPT simulator Sim that given the public key $\mathbf{H}, \mathbf{s}, \mathbf{e}, G$ and challenges (β, i) outputs a transcript $(c, \beta, h, i, \mathsf{Rsp}_i)$ that is indistinguishable from the transcript of an honest execution of the protocol. The simulator Sim proceeds as follows:

1. Compute $\sigma_j, \mathbf{v}_j$ and the corresponding commitments c_j as in the protocol, except for σ'_1 which is chosen randomly in G.
2. Compute $\sigma = \sigma_N \circ \cdots \circ \sigma'_1$.
3. Compute $\mathbf{v}$ and the commitment c as in the protocol.
4. Compute $\mathbf{y}$ such that $\mathbf{y}\mathbf{H}^\top = \beta\mathbf{s}$. This $\mathbf{y}$ is not necessarily in G, as else the simulator has to solve the R-SDP(G).
5. Compute $\tilde{\mathbf{e}}_0 = \beta\mathbf{e}$ and for all $\ell \in \{1, \ldots, i-1\}$ the $\tilde{\mathbf{e}}_j = \sigma_j(\tilde{\mathbf{e}}_{\ell-1}) + \mathbf{v}_j$.
6. Compute $\tilde{\mathbf{e}}'_i = \sigma_i(\tilde{\mathbf{e}}_{i-1}) + \mathbf{v}_i + \sigma_{i+1}^{-1} \circ \cdots \circ \sigma_N^{-1}(\mathbf{y} - \sigma(\beta\mathbf{e}))$.
7. Compute the remaining $\tilde{\mathbf{e}}'_j = \sigma_j(\tilde{\mathbf{e}}'_{\ell-1}) + \mathbf{v}_j$ for all $\ell \in \{i+1, \ldots, N\}$.
8. Compute the commitment $h = \mathsf{Hash}(\tilde{\mathbf{e}_1}, \ldots, \tilde{\mathbf{e}}_N)$
9. If $i = 1$, respond with $\mathsf{Rsp}_1 = (c_1, \tilde{\mathbf{e}}_1, \mathsf{Seeds})$ and if $i \neq 1$ respond with $\mathsf{Rsp}_i = (c_i, \tilde{\mathbf{e}}_i, \sigma_1, \mathsf{Seeds})$.
10. Output transcript $(c, \beta, h, i, \mathsf{Rsp}_i)$.

Since $\tilde{\mathbf{e}}'_i$ of the simulator and $\tilde{\mathbf{e}}_i$ of an honest prover are masked by a random $\mathbf{v}_i$, which is not known to the verifier, the responses $\tilde{\mathbf{e}}_i$ and $\tilde{\mathbf{e}}'_i$ are indistinguishable. Since also σ'_1 was chosen at random in G, σ'_1 of the simulator and σ_1 of an honest prover are also indistinguishable. Since we assume the commitments are hiding, c_i do not leak any information on σ_i and $\mathbf{v}_i$. Thus, the transcript of the simulator and the transcript of an honest prover are indistinguishable. □

Proposition 6. *The R-BG scheme presented in Figure 5 has a soundness error of*

$$\varepsilon(N, q) = \frac{1}{N} + \frac{N-1}{N(q-1)}.$$

Proof. Since we have a $(2,2)$ out of $(q-1,N)$-special sound protocol, we get (e.g., from [7,8]) that the soundness error is given by

$$1-\left(1-\frac{1}{q-1}\right)\left(1-\frac{1}{N}\right)=\frac{1}{N}+\frac{N-1}{N(q-1)}.$$

□