Building MPCitH-based Signatures from MQ, MinRank, Rank SD and PKP

Thibauld Feneuil[1,2]

[1] CryptoExperts, Paris, France
[2] Sorbonne Université, CNRS, INRIA, Institut de Mathématiques
de Jussieu-Paris Rive Gauche, Ouragan, Paris, France

`thibauld.feneuil@cryptoexperts.com`

Abstract. The MPC-in-the-Head paradigm is a useful tool to build practical signature schemes. Many such schemes have been already proposed, relying on different assumptions. Some are relying on standard symmetric primitives like AES, some are relying on MPC-friendly primitives like LowMC or Rain, and some are relying on well-known hard problems like the syndrome decoding problem.

This work focuses on the third type of MPCitH-based signatures. Following the same methodology as the work of Feneuil, Joux and Rivain (CRYPTO'22), we apply the MPC-in-the-Head paradigm to several problems: the multivariate quadratic problem, the MinRank problem, the rank syndrome decoding problem and the permuted kernel problem. Our goal is to study how this paradigm behaves for each of those problems.

For the multivariate quadratic problem, our scheme outperforms slightly the existing schemes when considering large fields (as $\mathbb{F}_{256}$), and for the permuted kernel problem, we obtain larger sizes. Even if both schemes do not outperform the existing ones according to the communication cost, they are highly parallelizable and compatible with some MPC-in-the-Head techniques while the former proposals were not.

Moreover, we propose two efficient MPC protocols to check that the rank of a matrix over a field $\mathbb{F}_q$ is upper bounded by a public constant. The first one relies on the rank decomposition while the second one relies on q-polynomials. We then use them to build signature schemes relying on the MinRank problem and the rank syndrome decoding problem. Those schemes outperform the former schemes, achieving sizes below 6 KB (while using only 256 parties for the MPC protocol).

Keywords: zero-knowledge proofs, post-quantum signatures, MPC-in-the-head

1 Introduction

The MPC-in-the-Head paradigm [IKOS07] is a versatile framework to design zero-knowledge proofs of knowledge, by relying on secure multi-party computation (MPC) techniques. After sharing the secret witness, the prover emulates "in her head" an MPC protocol with N parties and commits each party's view independently. The verifier then challenges the prover to reveal the views of a random subset of parties. By the privacy of the MPC protocol, nothing is revealed about the witness, which implies the zero-knowledge property. On the other hand, a malicious prover needs to cheat for at least one party, which shall be discovered by the verifier with high probability, hence ensuring the soundness property.

Combined with the Fiat-Shamir transform [FS87], the MPCitH paradigm provides a useful tool to build practical signatures. The security of the resulting scheme only depends on the security of commitment/hash functions and the security of a one-way function. The choice of this one-way function is left to the signature designers. A first research track [ARS$^+$15,DKR$^+$21] consists to design MPC-friendly primitives and to use them with the MPC-in-the-Head paradigm to get short signatures. This methodology has the disadvantage to require deep cryptanalysis of the introduced primitives. Another strategy would be to use standard symmetric primitives like AES as security assumptions for the MPCitH-based signatures, but it tends to produce larger signatures. As a last option, we can rely on a hard problem which exists for a long time and thus which are well understood. For example, [FJR22] succeeds to design an efficient signature scheme using the syndrome decoding problem (over the Hamming weight), which is one of the oldest problems of code-based cryptography. The case of the syndrome decoding problem has been covered, but a natural question would be

Which performances can we have when using
the MPC-in-the-Head paradigm with other hard problems?

Some articles [Wan22,FJR21,BG22,FMRV22] already apply this paradigm to hard problems (multivariate quadratic problem, MinRank problem, subset sum problem, ...). One of the drawbacks of almost all the schemes is that, when there is no structure to exploit, they need to rely on *protocols with helper* [Beu20]. This technique introduced by [KKW18] and formalized by [Beu20] is quite powerful, but suffers from a high computational cost. As consequence, the number of parties involved in the MPC protocol must stay low to have a practical scheme (in practice, we often take 32 as a limit for the number of parties), preventing achieving smaller sizes. Recently, [BG22] succeeds to leverage the structure when considering structured hard problem (as the ideal rank syndrome decoding problem) and thus succeeds to achieve smaller sizes by removing the helper from [FJR21].

The present work aims to complete the state of the art of the MPC-in-the-Head applied to hard problems. Table 1 overviews schemes producing the shortest signatures for some hard problems.

Hard Problem	**Best scheme**	**Achieved sizes**
Multivariate Quadratic	Over $\mathbb{F}_4$, [Wan22] Over $\mathbb{F}_{256}$, our work	8.4 – 9.4 KB 6.9 – 8.3 KB
Min Rank	Our work	5.4 – 7.0 KB
Permuted Kernel	[BG22]	8.6 – 9.7 KB
Subset Sum	[FMRV22]	21.1 – 33.2 KB
Syndrome Decoding *(Hamming)*	[FJR22]	Over $\mathbb{F}_2$, 10.9 – 15.6 KB Over $\mathbb{F}_{256}$, 8.3 – 11.5 KB
Syndrome Decoding *(Rank)*	Our work	5.8 – 7.2 KB

Table 1: State of the art of the MPCiH-based signatures, including this work.

Our contribution. In this article, we consider several hard problems for which we propose new zero-knowledge proofs using the MPC-in-the-Head paradigm.

First, we propose a new zero-knowledge proof of knowledge for the *multivariate quadratic* problem. The resulting signature scheme outperforms [Wan22] only when the base field is large enough (*e.g.* $\mathbb{F}_{256}$).

Secondly, we propose two efficient MPC protocols which take as input a matrix $M \in \mathbb{F}_q^{n\times m}$ and which check that the rank of M is upper bounded by r, where r is a public positive integer:

- the first one decomposes M as a product TR where $T \in \mathbb{F}_q^{n\times r}$ and $R \in \mathbb{F}_q^{r\times m}$, and uses an MPC protocol that checks the correctness of a matrix multiplication;
- the second one relies on the fact that the rows of M (represented as elements of $\mathbb{F}_{q^m}$) are roots of a q-polynomial of degree q^r and on the fact that computing a q-polynomial is efficient in MPC while exploiting the linearity of the Frobenius endomorphism $v \mapsto v^q$.

We then use those protocols to build efficient signatures relying on the *MinRank* problem or on the *rank syndrome decoding* problem. Our schemes outperform all the previous proposals, by achieving sizes below 7 KB. They also outperform the [BG22]'s proposals which use structured problems (as the ideal rank syndrome decoding problem) to achieve small sizes.

Finally, we propose a new zero-knowledge proof of knowledge for the *permuted kernel* problem. The existing proposals are already quite efficient, achieving sizes below 10 KB [BG22]. They all rely on permutations, which is quite natural since the problem itself uses permutations. However, securely implementing permutations is a tricky exercise. Our proposal achieves larger sizes, but uses no permutation at all. Our proposal is also compatible with the techniques proposed by [FR22] (as fast signature verification) and those proposed by [AMGH$^+$22] (which improves the running times of the MPCitH-based schemes relying on additive sharings), while the previous proposals for PKP are not.

Paper organization. The paper is organized as follows: In Section 2, we introduce the necessary background on the MPC-in-the-Head paradigm. We present our general methodology in Section 3. Then we apply it to the *multivariate quadratic* problem in Section 4, to the *MinRank* problem and the *rank syndrome decoding* problem in Section 5, and to the *permuted kernel* problem in Section 6. Finally, we discuss about the computational performances of the obtained schemes in Section 7.

2 Preliminaries

Throughout the paper, $\mathbb{F}$ shall denote a finite field. For any $m \in \mathbb{N}^*$, the integer set $\{1, \dots, m\}$ is denoted $[m]$. For a probability distribution D, the notation $s \leftarrow D$ means that s is sampled from D. For a finite set S, the notation $s \leftarrow S$ means that s is uniformly sampled at random from S. For an algorithm $\mathcal{A}$, $out \leftarrow \mathcal{A}(in)$ further means that out is obtained by a call to $\mathcal{A}$ on input in (using uniform random coins whenever $\mathcal{A}$ is probabilistic). Along the paper, probabilistic polynomial time is abbreviated PPT.

In this paper, we shall use the standard cryptographic notions of (honest verifier) zero-knowledge proof of knowledge and secure multiparty computation protocols (in the semi-honest model). We refer to [FR22] for the formal definition of those notions.

2.1 The MPC-in-the-Head Paradigm

The MPC-in-the-Head (MPCitH) paradigm introduced in [IKOS07] offers a way to build zero-knowledge proofs from secure multi-party computation (MPC) protocols. Let us assume we have an MPC protocol in which N parties $\mathcal{P}_1, \dots, \mathcal{P}_N$ securely and correctly evaluate a function f on a secret input x with the following properties:

- the secret x is encoded as a sharing $[\![x]\!]$ and each $\mathcal{P}_i$ takes a share $[\![x]\!]_i$ as input;
- the function f outputs Accept or Reject;
- the views of t parties leak no information about the secret x.

We can use this MPC protocol to build a zero-knowledge proof of knowledge of an x for which $f(x)$ evaluates to Accept. The prover proceeds as follows:

- she builds a random sharing $[\![x]\!]$ of x;
- she simulates locally ("in her head") all the parties of the MPC protocol;
- she sends commitments to each party's view, *i.e.* party's input share, secret random tape and sent and received messages, to the verifier;
- she sends the output shares $[\![f(x)]\!]$ of the parties, which should correspond to Accept.

Then the verifier randomly chooses t parties and asks the prover to reveal their views. After receiving them, the verifier checks that they are consistent with an honest execution of the MPC protocol and with the commitments. Since only t parties are opened, revealed views leak no information about the secret x, while the random choice of the opened parties makes[3] the cheating probability upper bounded by $(N-t)/N$, thus ensuring the soundness of the zero-knowledge proof.

All MPC protocols described in this article fit the model described in [FR22], meaning that the parties take as input an additive sharing $[\![x]\!]$ of the secret x (one share per party) and that they compute one or several rounds in which they perform three types of actions:

---

[3] We implicitly assume here that the communication between parties is broadcast.

Receiving randomness: the parties receive a random value ε from a randomness oracle $\mathcal{O}_R$. When calling this oracle, all the parties get the same random value ε.

Receiving hint: the parties can receive a sharing $[\![\beta]\!]$ (one share per party) from a hint oracle $\mathcal{O}_H$. The hint β can depend on the witness w and the previous random values sampled from $\mathcal{O}_R$.

Computing & broadcasting: the parties can locally compute $[\![\alpha]\!] := [\![\varphi(v)]\!]$ from a sharing $[\![v]\!]$ where φ is an $\mathbb{F}$-linear function, then broadcast all the shares $[\![\alpha]\!]_1, \ldots, [\![\alpha]\!]_N$ to publicly reconstruct $\alpha := \varphi(v)$. The function φ can depend on the previous random values $\{\varepsilon^i\}_i$ from $\mathcal{O}_R$ and on the previous broadcasted values.

We refer to [FR22] for the detailed transformation of such MPC protocol into zero-knowledge proofs of knowledge and for the resulting performances.

3 Methodology

In each of the following sections, we focus on a specific hard problem which is supposed quantum-resilient:

- Section 4: Multivariate Quadratic Problem;
- Section 5.2: Min Rank Problem;
- Section 5.3: Syndrome Decoding in the *rank* metric;
- Section 6: Permuted Kernel Problem.

For each of them, we will use the MPC-in-the-Head paradigm to build a new zero-knowledge protocol. To proceed, we will first describe the MPC protocol we use. This MPC protocol will fit the model described in [FR22] and will satisfy the following properties:

- it takes as input an additive sharing of a candidate solution of the studied problem, and eventually an additive sharing of auxiliary data;
- the MPC parties get (only once) a common random value from an oracle $\mathcal{O}_R$;
- when the tested solution is valid (*i.e.* a solution of the studied hard problem) and when the auxiliary data are genuinely computed, the MPC protocol always outputs ACCEPT; otherwise, it outputs ACCEPT with probability at most p (over the randomness of $\mathcal{O}_R$), where p is called the *false positive rate*;
- the views of all the parties except one leak no information about the candidate solution.

By applying the MPC-in-the-Head paradigm to this MPC protocol, we get a 5-round zero-knowledge proof of knowledge of a solution of the studied problem (see [FR22, Theorem 2] with the privacy threshold $\ell := N-1$), with soundness error

$$\frac{1}{N} + \left(1 - \frac{1}{N}\right) \cdot p$$

where N is the number of parties involved in the multi-party computation. We do not exhibit the obtained proof of knowledge since the transformation is standard. We refer the reader to [FR22] for a detailed explanation about how concretely apply the MPC-in-the-Head paradigm.

To obtain a signature scheme, we apply the Fiat-Shamir transform [FS87] to the previous protocol. Since this protocol has 5 rounds, the security of the resulting scheme should take into account the attack of [KZ20]. More precisely, the forgery cost of the signature scheme is given by

$$\mathrm{cost}_{\mathrm{forge}} := \min_{\tau_1,\tau_2:\tau_1+\tau_2=\tau} \left\{ \frac{1}{\sum_{i=\tau_1}^{\tau} \binom{\tau}{i} p^i (1-p)^{\tau-i}} + N^{\tau_2} \right\}$$

where τ is the number of parallel executions.

Remark 1. For the permuted kernel problem, the MPC protocol we propose slightly differ from the above description. The parties call the oracle $\mathcal{O}_R$ twice (instead of once). The resulting scheme is a 7-round proof (not a 5-round proof) with the same soundness error as before. However, the forgery cost is not the same (see Section 6).

Finally, we compare the resulting scheme with all the former schemes which are non-interative identification schemes based on the same security assumption. To proceed, we first list all these schemes with their formulae of the forgery security and of the communication cost. Since some quantities occurs several times, we define some notations to ease the readability. For the forgery cost, we introduce the two following notations:

- $\varepsilon_{\text{helper}}(\tau, M, \varepsilon)$ is the soundness error of a protocol with helper [Beu20] when the helper entity is emulated by a cut-and-choose phase. M is the total number of repetitions in the cut-and-choose phase, ε is the soundness of the unitary protocol relying on the helper, and τ is the number of repetitions of this unitary protocol. We have
$$\varepsilon_{\text{helper}}(\tau, M, \varepsilon) := \max_{M-\tau \le k \le M} \left\{ \frac{\binom{k}{M-\tau}}{\binom{M}{M-\tau}} \cdot \varepsilon^{k-(M-\tau)} \right\}.$$
- $\mathrm{KZ}(p_1, p_2)$ is the forgery cost of [KZ20] for a 5-round protocol[4]. We have
$$\mathrm{KZ}(p_1, p_2) := \min_{\tau_1, \tau_2 : \tau_1 + \tau_2 = \tau} \left\{ \frac{1}{\sum_{i=\tau_1}^{\tau} \binom{\tau}{i} p_1^i (1-p_1)^{\tau - i}} + \frac{1}{p_2^{\tau_2}} \right\}$$

For the communication cost (*i.e.* the signature size), we introduce the following notations:

- μ_{seed} is the cost of sending a λ-bit seed;
- μ_{dig} is the cost of sending a 2λ-bit commitment/hash digest;
- μ_{helper} is the cost (per repetition) of using the helper technique of [Beu20], this cost satisfies
$$\mu_{\text{helper}} \le (\mu_{\text{seed}} + \mu_{\text{dig}}) \cdot \log_2 \binom{M}{\tau}$$
where M is the number of repetitions involved in the cut-and-choose phase emulating the helper. It corresponds to the cost of revealing $M - \tau$ leaves among M in a seed tree, with the cost of sending the authentication paths of τ leaves among M in a Merkle tree.
- μ_{MPCitH} is the fixed cost (per repetition) of using the MPC-in-the-Head paradigm, we have
$$\mu_{\text{MPCitH}} = \mu_{\text{seed}} \cdot \log_2 N + \mu_{\text{dig}}.$$
It corresponds to the cost of revealing all the leaves but one in a seed tree of N leaves (plus a commitment digest).

Then, to get a numerical comparison, we select one or two instances of the studied hard problem and we compare all these schemes for these precise instances. To proceed, we need to select the parameters of the schemes when relevant. The signature schemes based on the MPC-in-the-Head paradigm have as parameter the number N of parties involved in the multi-party computation. When taking a small N, we get a faster scheme, but when taking a large N, we get shorter signature sizes. To have a fair comparison between the different schemes, we will always take the same N:

- when the protocol relies on a helper, we take $N = 8$ to have a fast scheme and $N = 32$ to have short sizes.
- otherwise, we take $N = 32$ to have a fast scheme and $N = 256$ to have short sizes.

3.1 Matrix Multiplication Checking Protocol

In our constructions, we need an MPC protocol that checks that three matrices X, Y, Z satisfy $Z = X \cdot Y$. We describe in Figure 1 such a protocol Π_{MM}^{η} which has a positive parameter η. This protocol is a matrix variant of the multiplication checking protocol of [BN20].

---

[4] in the case where the verifier can not perform some checks after receiving the first response (see [KZ20] for details).

Inputs: Each party takes a share of the following sharings as inputs: $[\![X]\!]$ where $X \in \mathbb{F}_q^{m\times p}$, $[\![Y]\!]$ where $Y \in \mathbb{F}_q^{p\times n}$, $[\![Z]\!]$ where $\mathbb{F}_q^{m\times n}$, $[\![A]\!]$ where A has been uniformly sampled from $\mathbb{F}_q^{p\times \eta}$, and $[\![C]\!]$ where $C \in \mathbb{F}_q^{m\times \eta}$ satisfies $C = XA$.

MPC Protocol:

1. The parties get a random $\Sigma \in \mathbb{F}_q^{n\times \eta}$.
2. The parties locally set $[\![D]\!] = [\![Y]\!]\Sigma + [\![A]\!]$.
3. The parties broadcast $[\![D]\!]$ to obtain $D \in \mathbb{F}_q^{p\times \eta}$.
4. The parties locally set $[\![V]\!] = [\![X]\!]D - [\![C]\!] - [\![Z]\!]\Sigma$.
5. The parties open $[\![V]\!]$ to obtain $V \in \mathbb{F}_q^{m\times \eta}$.
6. The parties outputs ACCEPT if $V = 0$ and REJECT otherwise.

Fig. 1: The MPC protocol Π^{η}_{MM} which checks that $Z = X \cdot Y$.

Lemma 1. *If $Z = X\cdot Y$ and if C are genuinely computed, then Π^{η}_{MM} always outputs* ACCEPT. *If $Z \neq X\cdot Y$, then Π^{η}_{MM} outputs* ACCEPT *with probability at most $\frac{1}{q^\eta}$.*

Proof. We have

$$\begin{aligned} V &= XD - C - Z\Sigma \\ &= X(Y\Sigma + A) - C - Z\Sigma \\ &= (XY - Z)\Sigma - (C - XA). \end{aligned}$$

If $Z = XY$ and $C = XA$, V is equal to zero and thus the parties will always output ACCEPT. In contrast, if $Z \neq XY$, then there exists $(i^*, j^*) \in [m]\times[n]$ such that $Z_{i^*,j^*} - (X\cdot Y)_{i^*,j^*} \neq 0$. Given $k \in \{1,\dots,\eta\}$, $\Sigma_{j^*,k}$ is uniformly sampled in $\mathbb{F}_q$ and then $((Z - X\cdot Y)\Sigma)_{i^*,k}$ is uniformly random in $\mathbb{F}_q$ (because one of the sum term is uniformly random). Thus, the probability that V is zero is at most the probability that $(Z - X\cdot Y)\Sigma$ is equal to $(C - XA)$ on the row i^* whereas the row i^* of $(Z - X\cdot Y)\Sigma$ is uniformly random in $\mathbb{F}_q^\eta$, *i.e.* the probability that V is zero (at row i^*) is at most $\frac{1}{q^\eta}$.

□

3.2 MPCitH Optimizations

It is often possible to optimize the communication cost of a scheme relying on the MPC-in-the-Head paradigm. The common optimization tricks are the following:

- Except for the last party, the input share of a party can be derived from a seed using a pseudo-random generator. Thus, when we need to reveal the input share, we just need to reveal a seed. In practice, a prover must reveal the input shares of $N-1$ parties, so it would imply revealing $N-1$ seeds. To save more communication, we can generate the seeds using a tree structure, decreasing the number of revealed seeds to $\log_2(N)$ (see [KKW18, Sec. 2.3] for details).
- We do not need to reveal shares for shared random values (as A in Figure 1) since they can be entirely derived from the seeds of the previous point.
- We do not need to reveal shares for shared publicly-known values (see [KZ22, Sec. 2.4] for details). For example, we do not need to reveal the share of V broadcast by the hidden party in Figure 1. Indeed, this share can be deduced from the shares of the other parties and knowing that V must be equal to zero (otherwise the verification fails).

4 Proof of Knowledge for $\mathcal{MQ}$

We want to build a zero-knowledge proof of knowledge for the *multivariate quadratic problem*:

Definition 1 (Multivariate Quadratic Problem - Matrix Form). *Let $\mathbb{F}_q$ be the finite field with q elements. Let (m, n) be positive integers. The multivariate quadratic problem with parameters (q, m, n) is the following problem:*

Let $(A_i)_{i\in[m]}$, $(b_i)_{i\in[m]}$, x and y be such that:

1. *x is uniformly sampled from $\mathbb{F}_q^n$,*
2. *for all $i \in [m]$, A_i is uniformly sampled from $\mathbb{F}_q^{n\times n}$,*
3. *for all $i \in [m]$, b_i is uniformly sampled from $\mathbb{F}_q^n$,*
4. *for all $i \in [m]$, y_i is defined as $y_i := x^T A_i x + b_i^T x$.*

From $((A_i)_{i\in[m]}, (b_i)_{i\in[m]}, y)$, find x.

The prover wants to convince the verifier that she knows $x \in \mathbb{F}_q^n$ such that

$$\begin{cases} y_1 = x^T A_1 x + b_1^T x \\ \quad\vdots \\ y_m = x^T A_m x + b_m^T x \end{cases}$$

To proceed, she will rely on the MPC-in-the-Head paradigm: she will first share the secret vector x and then use an MPC protocol which verifies that this vector satisfies the above relations.

MPC Protocol. Instead of checking the m relations separately, we batch them into a linear combination where coefficients $\gamma_1, \ldots, \gamma_m$ are uniformly sampled in the field extension $\mathbb{F}_{q^\eta}$. The MPC protocol will check that

$$\sum_{i=1}^{m} \gamma_i (y_i - x^T A_i x - b_i^T x) = 0. \tag{1}$$

If one of the relations was not satisfied, then Equation (1) would be satisfied only with a probability $\frac{1}{q^\eta}$. We can write the equality as

$$\begin{aligned} \sum_{i=1}^{m} \gamma_i (y_i - b_i^T x) &= \sum_{i=1}^{m} \gamma_i (x^T A_i x) \\ &= x^T \left(\sum_{i=1}^{m} \gamma_i A_i \right) x \\ &= \langle x, w \rangle \quad \text{where} \quad w := \left(\sum_{i=1}^{m} \gamma_i A_i \right) x \end{aligned}$$

By defining $z := \sum_{i=1}^{m} \gamma_i (y_i - b_i^T x)$ and $w := \left(\sum_{i=1}^{m} \gamma_i A_i\right) x$, proving Equation (1) is equivalent to proving that

$$z = \langle x, w \rangle.$$

And to prove the above equality, we can rely on the subprotocol Π_{MM} described in Section 3.1 (assuming that all the scalars live in $\mathbb{F}_{q^\eta}$). Thus, the MPC protocol proceeds as follows:

1. The parties get random $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{q^\eta}$.
2. The parties locally set $[\![z]\!] = \sum_{i=1}^{m} \gamma_i (y_i - b_i^T [\![x]\!])$.
3. The parties locally set $[\![w]\!] = \left(\sum_{i=1}^{m} \gamma_i A_i\right) [\![x]\!]$.
4. The parties execute the protocol Π_{MM} to check that $z = \langle w, x \rangle$.

Since this sub-protocol Π_{MM} produces false positive events with a rate of $\frac{1}{q^\eta}$, if x does not satisfy the m $\mathcal{MQ}$ relations, the complete MPC protocol outputs Accept only with a probability of at most

$$\frac{1}{q^\eta} + \left(1 - \frac{1}{q^\eta}\right) \frac{1}{q^\eta} = \frac{2}{q^\eta} - \frac{1}{q^{2\eta}}.$$

The complete MPC protocol is described in Figure 2.

Public values: The matrices $A_1, \ldots, A_m \in \mathbb{F}_q^{n \times n}$, the vectors $b_1, \ldots, b_m \in \mathbb{F}_q^n$, and the outputs $y_1, \ldots, y_m \in \mathbb{F}_q$.

Inputs: Each party takes a share of the following sharings as inputs: $[\![x]\!]$ where $x \in \mathbb{F}_q^n$, $[\![a]\!]$ where a has been uniformly sampled from $\mathbb{F}_{q^\eta}^n$, and $[\![c]\!]$ where $c \in \mathbb{F}_{q^\eta}$ satisfies $c = -\langle a, x \rangle$.

MPC Protocol:

1. The parties get random $\gamma_1, \ldots, \gamma_m \in \mathbb{F}_{q^\eta}$ and a random $\varepsilon \in \mathbb{F}_{q^\eta}$.
2. The parties locally set $[\![z]\!] = \sum_{i=1}^m \gamma_i (y_i - b_i^T [\![x]\!])$.
3. The parties locally set $[\![w]\!] = \left(\sum_{i=1}^m \gamma_i A_i\right) [\![x]\!]$.
4. The parties locally set $[\![\alpha]\!] = \varepsilon \cdot [\![w]\!] + [\![a]\!]$.
5. The parties open $\alpha \in \mathbb{F}_{q^\eta}^n$.
6. The parties locally set $[\![v]\!] = \varepsilon \cdot [\![z]\!] - \langle \alpha, [\![x]\!] \rangle - [\![c]\!]$.
7. The parties open $v \in \mathbb{F}_{q^\eta}$.
8. The parties outputs ACCEPT if $v = 0$ and REJECT otherwise.

Fig. 2: An MPC protocol that verifies that the given input corresponds to a solution of an $\mathcal{MQ}$ problem.

Proof of Knowledge. Using the MPC-in-the-Head paradigm (see Section 2.1), we transform the above MPC protocol into an interactive zero-knowledge proof of knowledge which enables to convince a verifier that a prover knows the solution of a $\mathcal{MQ}$ problem. The soundness error of the resulting protocol is

$$\varepsilon := \frac{1}{N} + \left(1 - \frac{1}{N}\right)\left(\frac{2}{q^\eta} - \frac{1}{q^{2\eta}}\right).$$

By repeating the protocol τ times, we get a soundness error of ε^τ. To obtain a soundness error of λ bits, we can take $\tau = \left\lceil \frac{-\lambda}{\log_2 \varepsilon} \right\rceil$. We can transform the interactive protocol into a non-interactive argument / signature thanks to the Fiat-Shamir transform [FS87]. According to [KZ20], the security of the resulting scheme is

$$\mathrm{cost}_{\mathrm{forge}} := \min_{\tau_1, \tau_2 : \tau_1 + \tau_2 = \tau} \left\{ \frac{1}{\sum_{i=\tau_1}^{\tau} \binom{\tau}{i} p^i (1-p)^{\tau - i}} + N^{\tau_2} \right\}$$

where $p := \frac{2}{q^\eta} - \frac{1}{q^{2\eta}}$.

The communication cost of the scheme (in bits) is

$$4\lambda + \tau \cdot \left((\underbrace{n \cdot \log_2(q)}_{x} + \underbrace{n \cdot \eta \cdot \log_2(q)}_{\alpha} + \underbrace{\eta \cdot \log_2(q)}_{c} + \underbrace{\lambda \cdot \log_2 N + 2\lambda}_{\text{MPCitH}} \right)$$

where λ is the security level, η is a scheme parameter and τ is computed such that the soundness error is of λ bits in the interactive case and such that $\mathrm{cost}_{\mathrm{forge}}$ is of λ bits in the non-interactive case.

Performances and comparison. In what follows, we compare our scheme with the state of the art on two $\mathcal{MQ}$ instances:

Instance 1. Multivariate Quadratic equations over a small field:

$$(q, m, n) = (4, 88, 88),$$

Instance 2. Multivariate Quadratic equations over a larger field:

$$(q, m, n) = (256, 40, 40).$$

Both of these instances are believed to correspond to a security of 128 bits [BMSV22].

We provide in Tables 2 and 3 a complete comparison of our scheme with the state of the art. In the comparison we put MQ-DSS [CHR+16] which corresponds to the non-interactive version of the 5-round identification scheme of [SSH11]. In the sake of completeness, we also put how the 3-round identification scheme of [SSH11] would perform when applying the Fiat-Shamir transform on it.

Over a small field, the Mesquite [Wan22] scheme has the smallest communication cost, even if our scheme produces competitive signature size. Over a larger field, we can produce signature size close to 7 KB, and thus we outperform all the former schemes.

Remark 2. In constrast with the former state of the art, the communication cost of our scheme is independent to the number m of $\mathcal{MQ}$ relations.

Scheme Name	Security	Signature Size
[SSH11] (3 rounds)	$(3/2)^\tau$	$\mu_{\text{dig}} + \tau\,[2\mu_{\text{var}} + \mu_{\text{out}} + 2\mu_{\text{dig}}]$
MQ-DSS [CHR$^+$16]	$\mathrm{KZ}(\frac{1}{q}, \frac{1}{2})$	$2\mu_{\text{dig}} + \tau\,[2\mu_{\text{var}} + \mu_{\text{out}} + 2\mu_{\text{dig}}]$
MudFish [Beu20]	$\varepsilon_{\text{helper}}(\tau, M, \frac{1}{q'})^{-1}$	$\mu_{\text{dig}} + \tau\,[2\mu_{\text{var}} + \mu_{\text{out}} + 2\mu_{\text{seed}} + \mu_{\text{dig}} \cdot \log_2(q') + \mu_{\text{helper}}]$
Mesquite [Wan22]	$\varepsilon_{\text{helper}}(\tau, M, \frac{1}{N})^{-1}$	$\mu_{\text{dig}} + \tau\,[\mu_{\text{var}} + \mu_{\text{out}} + \mu_{\text{MPCitH}} + \mu_{\text{helper}}]$
Our scheme	$\mathrm{KZ}(\frac{2}{q^\eta} - \frac{1}{q^{2\eta}}, \frac{1}{N})$	$2\mu_{\text{dig}} + \tau\,[(1+\eta) \cdot \mu_{\text{var}} + \eta \cdot \log_2 q + \mu_{\text{MPCitH}}]$

Table 2: Sizes of the signatures relying on the $\mathcal{MQ}$ problem (restricting to the schemes using the FS heuristics). The used notations are: $\mu_{\text{var}} := n \log_2 q$, $\mu_{\text{out}} := m \log_2 q$, plus all the notations defined in Section 3.

Instance	Protocol Name	Variant	Parameters				Signature Size
			N	M	τ	η	
$q=4$ $m=88$ $n=88$	[SSH11] (3 rounds)	-	-	219	-	-	28 502 B
	MQ-DSS [CHR$^+$16]	-	-	316	-	-	41 444 B
	MudFish [Beu20]	-	4	191	68	-	14 640 B
	Mesquite [Wan22]	Fast	8	187	49	-	9 578 B
		Short	32	389	28	-	**8 609 B**
	Our scheme	Fast	32	-	40	6	10 764 B
		Short	256	-	25	8	9 064 B
$q=256$ $m=40$ $n=40$	[SSH11] (3 rounds)	-	-	219	-	-	40 328 B
	MQ-DSS [CHR$^+$16]	-	-	156	-	-	28 768 B
	MudFish [Beu20]	Fast	8	176	51	-	15 958 B
		Short	16	250	36	-	13 910 B
	Mesquite [Wan22]	Fast	8	187	49	-	11 339 B
		Short	32	389	28	-	9 615 B
	Our scheme	Fast	32	-	36	2	8 488 B
		Short	256	-	25	2	**7 114 B**

Table 3: Sizes of the signatures relying on the $\mathcal{MQ}$ problem (restricting to the schemes using the FS heuristics). Numerical comparison.

5 Proofs of Knowledge for MinRank and Rank SD

In this section, we propose arguments of knowledge for the MinRank problem (Section 5.2) and the Rank SD problem (Section 5.3). But before that, in Section 5.1, we propose two efficient MPC protocols which check that a matrix M has a rank of at most r.

In what follows, we denote $\mathrm{wt}_R(M)$ the rank of a matrix M.

5.1 Matrix Rank Checking Protocols

We want to build MPC protocols which check that a matrix has a rank of a most r. Such MPC protocols will be used for arguments of knowledge with the MPC-in-the-Head paradigm. We propose two protocols:

- the first one relies on the rank decomposition of matrices. It has the advantage to be quite *simple*, but its false positive rate is *large*.
- the second one relies on linearized polynomials. It has the advantage to have a *very small* false positive rate, but it sometimes requires to manipulate field extensions of *large degrees*.

Using Rank Decomposition. Let us design an MPC protocol which checks that a matrix $M \in \mathbb{F}^{m\times n}$ has a rank of at most r, *i.e.* $\mathrm{wt}_R(M) \leq r$. To proceed, we will rely on the *decomposition rank*:

$$\text{a matrix } M \in \mathbb{F}_q^{n\times m} \text{ has a rank of at most } r$$
$$\text{if and only if there exists } T \in \mathbb{F}_q^{n\times r} \text{ and } R \in \mathbb{F}_q^{r\times m} \text{ such that } M = TR.$$

In practice, our MPC protocol that we will denote $\Pi^{\eta}_{\text{RC-RD}}$ takes as input such matrices T and R (in addition to M) and simply executes the matrix multiplication checking protocol Π^{η}_{MM} (see Section 3.1), for some positive integer η.

Theorem 1. *If* $\mathrm{wt}_R(M) \leq r$ *and if* T, R *are genuinely computed, then* $\Pi^{\eta}_{\text{RC-RD}}$ *always outputs* ACCEPT. *If* $\mathrm{wt}_R(M) > r$, *then* $\Pi_{\text{RC-RD}}$ *outputs* ACCEPT *with probability at most* $\frac{1}{q^\eta}$. *More precisely, if* $\mathrm{wt}_R(M) = w + \delta$ *with* $\delta \geq 1$, *then* $\Pi^{\eta}_{\text{RC-RD}}$ *outputs* ACCEPT *with probability at most* $\frac{1}{q^{\delta\cdot\eta}}$.

Proof. The final broadcast matrix V in Π^{η}_{MM} satisfies

$$V = (TR - M)\Sigma - (C - TA)$$

where matrices A and C have been built before receiving the random Σ. We have

$$\begin{aligned} \mathrm{wt}_R(M - TR) &\geq \mathrm{wt}_R(M) - \mathrm{wt}_R(TR) \\ &\geq (r + \delta) - r = \delta \end{aligned}$$

It means that $TR - M$ has at least δ non-zero coefficients $(i_1, j_1), \ldots, (i_\delta, j_\delta)$ which are over δ different rows and over δ different columns, *i.e.*

$$\forall k_1, k_2 \in [\delta],\ (i_{k_1} \neq i_{k_2}) \wedge (j_{k_1} \neq j_{k_2}).$$

Let us consider $k \in [\delta]$. The j_kth row of Σ is uniformly sampled in $\mathbb{F}_q^\eta$ and thus the i_kth row of $(M - TR)\Sigma$ is uniformly random in $\mathbb{F}_q^\eta$ (because one of the sum term is uniformly random). Thus, the probability that the i_kth row of V is zero is the probability that $(M - TR)\Sigma$ is equal to $(C - TA)$ on the row i_k whereas the row i_k of $(M - TR)\Sigma$ is uniformly random in $\mathbb{F}_q^\eta$, *i.e.* the probability that the i_kth row of V is zero is $\frac{1}{q^\eta}$. By taking a union bound over all k, we get that the probability that V is zero is at most $\frac{1}{q^{\delta\cdot\eta}}$. $\square$

Using Linearized Polynomials. In what follows, we represent a matrix of $\mathbb{F}_q^{m\times n}$ as an element of $(\mathbb{F}_q^m)^n$. We want to design an MPC protocol which checks that a matrix $M = (x_1, \ldots, x_n) \in (\mathbb{F}_{q^m})^n$ has a rank of at most r. Equivalently, it means that all x_i belongs to an $\mathbb{F}_q$-linear subspace U of $\mathbb{F}_{q^m}$ of dimension r. Let us define the polynomial $L_U(X)$ as

$$L_U(X) := \prod_{u\in U} (X - u) \in \mathbb{F}_{q^m}[X].$$

The degree of L_U is q^r since U has q^r elements. Showing that $\mathrm{wt}(M) \leq r$ can be done by showing that all x_i's are roots of L_U.

According to [LN96, Theorem 3.52], L_U is a q-polynomial over $\mathbb{F}_{q^m}$, meaning that it is of the form

$$L_U(X) = X^{q^r} + \sum_{i=0}^{r-1} \beta_i X^{q^i}.$$

Such polynomials are convenient for multi-party computation since the Frobenius endomorphism $X \mapsto X^q$ is a linear application in field extensions of $\mathbb{F}_q$ and thus it is communication-free to compute $[\![x^q]\!], [\![x^{q^2}]\!], \ldots$ from $[\![x]\!]$.

The core idea of the rank checking protocol is to check that $L_U(x_1) = L_U(x_2) = \ldots = L_U(x_n) = 0$. To proceed, the MPC protocol will batch these checkings by uniformly sampling $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_{q^m}$ and checking that

$$\sum_{j=1}^{n} \gamma_j \cdot L_U(x_j) = 0. \tag{2}$$

If one x_i is not a root of the polynomial L_U, then Equation (2) is satisfies only with probability $\frac{1}{q^m}$. Let us rewrite the left term of (2):

$$\begin{aligned}\sum_{j=1}^{n} \gamma_j \cdot L_U(x_j) &= \sum_{j=1}^{n} \gamma_j \cdot \left(x_j^{q^r} + \sum_{i=0}^{r-1} \beta_i x_j^{q^i} \right) \\ &= \underbrace{\sum_{j=1}^{n} \gamma_j \cdot x_j^{q^r}}_{:=-z} + \sum_{i=0}^{r-1} \beta_i \cdot \underbrace{\sum_{j=1}^{n} \gamma_j x_j^{q^i}}_{:=w_i}\end{aligned}$$

By defining $z := -\sum_{j=1}^{n} \gamma_j \cdot x_j^{q^r}$ and $w_i := \sum_{j=1}^{n} \gamma_j x_j^{q^i}$ for $i \in \{0, \ldots, r\}$, proving Equation (2) is equivalent to proving

$$z = \langle \beta, w \rangle.$$

Our MPC protocol that we will denote $\Pi^{\eta}_{\text{RC-LP}}$ takes as input $[\![x_1]\!], \ldots, [\![x_n]\!]$ and $[\![L_U]\!] := X^{q^r} + \sum_{i=0}^{r-1} [\![\beta_i]\!] X^{q^i}$ proceeds as follows:

1. The parties get random $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_{q^{m \cdot \eta}}$.
2. The parties locally set $[\![z]\!] = -\sum_{j=1}^{n} \gamma_j [\![x_j]\!]^{q^r}$.
3. The parties locally set $[\![w_i]\!] = \sum_{j=1}^{n} \gamma_j [\![x_j]\!]^{q^i}$ for all $i \in \{0, \ldots, r-1\}$.
4. The parties execute the protocol Π_{MM} to check that $z = \langle \beta, w \rangle$ over $\mathbb{F}_{q^{m \cdot \eta}}$.

Theorem 2. *If $\mathrm{wt}_R(M) \leq r$ and if L_U are genuinely computed, then $\Pi^{\eta}_{\text{RC-LP}}$ always outputs* Accept. *If $\mathrm{wt}_R(M) > r$, then $\Pi^{\eta}_{\text{RC-LP}}$ outputs* Accept *with probability at most $\frac{1}{q^{m\cdot\eta}} + \left(1 - \frac{1}{q^{m\cdot\eta}}\right) \frac{1}{q^{m\cdot\eta}}$.*

Proof. $[\![L_U]\!]$ is a q-polynomial over $\mathbb{F}_{q^m}$ of degree exactly q^r. It means that its number of roots is at most q^r. According to [LN96, Theorem 3.50], the roots form a $\mathbb{F}_q$-linear subspace V of the field extension $\mathbb{F}_{q^s}$ of $\mathbb{F}_{q^m}$. Since $\mathbb{F}_{q^m}$ is also a linear subspace of $\mathbb{F}_{q^s}$, $V \cap \mathbb{F}_{q^m}$ is a linear subspace of $\mathbb{F}_{q^s}$ (and of $\mathbb{F}_{q^m}$). Its dimension is at most r (since it has at most q^r elements). If $\mathrm{wt}_R(M) > r$, there exist i^* such that

$$L_U(x_{i^*}) \neq 0.$$

We then have two options resulting in $\Pi^{\eta}_{\text{RC-LP}}$ outputing Accept:

- Either $\sum_{j=1}^{n} \gamma_j \cdot L_U(x_j) = 0$, which occurs with probability $\frac{1}{q^{m\cdot\eta}}$;
- Or $\sum_{j=1}^{n} \gamma_j \cdot L_U(x_j) \neq 0$, *i.e.* $z \neq \langle \beta, w \rangle$ and Π_{MM} outputs Accept, which occurs with probability $\frac{1}{q^{m\cdot\eta}}$ since Π_{MM} has a false positive rate of $\frac{1}{q^{m\cdot\eta}}$.

□

5.2 Proof of Knowledge for MinRank

We want to build a zero-knowledge proof of knowledge for the *MinRank problem*:

Definition 2 (MinRank Problem). *Let $\mathbb{F}_q$ be the finite field with q elements. Let m, n, and k be positive integers. The MinRank problem with parameters (q, m, n, k) is the following problem:*

Let $M_0, M_1, \ldots, M_k$, E and x such that:

- *x is uniformly sampled from $\mathbb{F}_q^k$,*
- *for all $i \in [k]$, M_i is uniformly sampled from $\mathbb{F}_q^{n \times m}$,*
- *E is uniformly sampled from $\{E \in \mathbb{F}_q^{n \times m} : \mathrm{wt}_R(E) \leq w\}$,*
- *M_0 is defined as $M_0 = E - \sum_{i=1}^{k} x_i M_i$.*

From $(M_0, M_1, \ldots, M_k)$, find x.

The prover wants to convince the verifier that she knows such an x. To proceed, the prover will first share the secret vector x and then use an MPC protocol which verifies that this vector satisfies the above property.

MPC Protocol. We want to build an MPC protocol which takes as input (a sharing of) x and which outputs

$$\begin{cases} \textsc{Accept} & \text{if } \mathrm{wt}_R(E) \leq r \\ \textsc{Reject} & \text{otherwise.} \end{cases}$$

where $E := M_0 + \sum_{i=1}^{k} x_i M_i$.

Given $\llbracket x \rrbracket$, the parties can locally build $\llbracket E \rrbracket$ as $M_0 + \sum_{i=1}^{k} \llbracket x_i \rrbracket M_i$. It remains to check that $\llbracket E \rrbracket$ corresponds to the sharing of a matrix of rank at most r. It can be done using one of the two rank checking protocols described in Section 5.1: $\Pi^{\eta}_{\text{RC-RD}}$ relying on the rank decomposition or $\Pi^{\eta}_{\text{RC-LP}}$ relying on linearized polynomials, for some parameter η.

The complete MPC protocol is described in Figure 3 when relying on the rank decomposition and in Figure 4 when relying on linearized polynomials. In the second case, the rows of the matrix E are rewritten as elements of $\mathbb{F}_{q^m}$, but when $m \neq n$, it can be more convenient to work on the columns (depending of the values of m and n).

Public values: $M_0, M_1, \ldots, M_k \in \mathbb{F}_q^{n \times m}$.

Inputs: Each party takes a share of the following sharings as inputs: $\llbracket x \rrbracket$ where $x \in \mathbb{F}_q^k$, $\llbracket T \rrbracket$ where $T \in \mathbb{F}_q^{n \times r}$, $\llbracket R \rrbracket$ where $R \in \mathbb{F}_q^{r \times m}$, $\llbracket a \rrbracket$ where a has been uniformly sampled from $\mathbb{F}_q^{r \times \eta}$, and $\llbracket c \rrbracket \in \mathbb{F}_q^{n \times \eta}$, such that $M_0 + \sum_{i=1}^{k} x_i M_i = TR$ and $c = Ta$.

MPC Protocol:

1. The parties get a random $\Sigma \in \mathbb{F}_q^{m \times \eta}$.
2. The parties locally set $\llbracket E \rrbracket = M_0 + \sum_{i=1}^{k} \llbracket x_i \rrbracket M_i$.
3. The parties locally set $\llbracket \alpha \rrbracket = \llbracket R \rrbracket \Sigma + \llbracket a \rrbracket$.
4. The parties open $\alpha \in \mathbb{F}_q^{r \times \eta}$.
5. The parties locally set $\llbracket v \rrbracket = \llbracket T \rrbracket \alpha - \llbracket c \rrbracket - \llbracket E \rrbracket \Sigma$.
6. The parties open $v \in \mathbb{F}_q^{n \times \eta}$.
7. The parties outputs $\textsc{Accept}$ if $v = 0$ and $\textsc{Reject}$ otherwise.

Fig. 3: An MPC protocol based on the *rank decomposition* technique ($\Pi_{\text{RC-RD}}$) which verifies that the given input corresponds to a solution of a MinRank problem.

Public values: $M_0, M_1, \ldots, M_k \in \mathbb{F}_q^{n\times m}$.

Inputs: Each party takes a share of the following sharings as inputs: $[\![x]\!]$ where $x \in \mathbb{F}_q^k$, $[\![L_U]\!] := X^{q^r} + \sum_{i=0}^{r-1} [\![\beta_i]\!] X^{q^i}$ where $L_U(X) := \prod_{u\in U}(X-u) \in \mathbb{F}_{q^m}[X]$, $[\![a]\!]$ where a has been uniformly sampled from $\mathbb{F}_{q^{m\cdot\eta}}^r$, and $[\![c]\!] \in \mathbb{F}_{q^{m\cdot\eta}}$, such that $c = -\langle \beta, a\rangle$.

MPC Protocol:

1. The parties get random $\gamma_1, \ldots, \gamma_n \in \mathbb{F}_{q^{m\cdot\eta}}$.
2. The parties get a random $\varepsilon \in \mathbb{F}_{q^{m\cdot\eta}}$.
3. The parties locally set $[\![E]\!] = M_0 + \sum_{i=1}^k [\![x_i]\!] M_i$.
4. The parties locally write the rows of $[\![E]\!]$ as elements $(e_1, \ldots, e_m)$ of $\mathbb{F}_{q^m}$
5. The parties locally set $[\![z]\!] = -\sum_{j=1}^n \gamma_j [\![e_j]\!]^{q^r}$.
6. The parties locally set $[\![w_i]\!] = \sum_{j=1}^n \gamma_j [\![e_j]\!]^{q^i}$ for all $i \in \{0, \ldots, r-1\}$.
7. The parties locally set $[\![\alpha]\!] = \varepsilon \cdot [\![w]\!] + [\![a]\!]$.
8. The parties open $\alpha \in \mathbb{F}_{q^{m\cdot\eta}}^r$.
9. The parties locally set $[\![v]\!] = \varepsilon \cdot [\![z]\!] - \langle \alpha, [\![\beta]\!]\rangle - [\![c]\!]$.
10. The parties open $v \in \mathbb{F}_{q^{m\cdot\eta}}$.
11. The parties outputs ACCEPT if $v = 0$ and REJECT otherwise.

Fig. 4: An MPC protocol based on the technique using *linearized polynomials* ($\mathit{\Pi}_{\text{RC-LP}}$) which verifies that the given input corresponds to a solution of a MinRank problem. U is a $\mathbb{F}_q$-linear subspace of $\mathbb{F}_{q^m}$ of dimension r which contains the rows $(e_1, \ldots, e_n)$ of $E := M_0 + \sum_{i=1}^k x_i M_i \in \mathbb{F}_q^{n\times m}$ represented as elements of $\mathbb{F}_{q^m}$.

Proof of Knowledge. Using the MPC-in-the-Head paradigm (see Section 2.1), we transform the above MPC protocol into an interactive zero-knowledge proof of knowledge which enables to convince a verifier that a prover knows the solution of a rank syndrome decoding problem. The soundness error of the resulting protocol is

$$\varepsilon := \frac{1}{N} + \left(1 - \frac{1}{N}\right) p_\eta$$

where $p_\eta := \frac{1}{q^\eta}$ when using $\mathit{\Pi}_{\text{RC-RD}}^\eta$ and $p_\eta := \frac{2}{q^{m\cdot\eta}} - \frac{1}{q^{2\cdot m\cdot\eta}}$ when using $\mathit{\Pi}_{\text{RC-LP}}^\eta$. By repeating the protocol τ times, we get a soundness error of ε^τ. To obtain a soundness error of λ bits, we can take $\tau = \left\lceil \frac{-\lambda}{\log_2 \varepsilon} \right\rceil$. We can transform the interactive protocol into a non-interactive proof / signature thanks to the Fiat-Shamir transform [FS87]. According to [KZ20], the security of the resulting scheme is

$$\text{cost}_{\text{forge}} := \min_{\tau_1,\tau_2:\tau_1+\tau_2=\tau} \left\{ \frac{1}{\sum_{i=\tau_1}^{\tau} \binom{\tau}{i} p_\eta^i (1-p_\eta)^{\tau-i}} + N^{\tau_2} \right\}.$$

When using $\mathit{\Pi}_{\text{RC-RD}}$, the communication cost of the scheme (in bits) is

$$4\lambda + \tau \cdot \left((\underbrace{k}_{x} + \underbrace{r\times m}_{R} + \underbrace{r\times n}_{T} + \underbrace{r\times\eta}_{\alpha} + \underbrace{n\times\eta}_{c}) \cdot \log_2 q + \underbrace{\lambda \cdot \log_2 N + 2\lambda}_{\text{MPCitH}} \right)$$

where λ is the security level, r is a scheme parameter and τ is computed such that the soundness error is of λ bits in the interactive case and such that $\text{cost}_{\text{forge}}$ is of λ bits in the non-interactive case.

And when using $\mathit{\Pi}_{\text{RC-LP}}$, the communication cost of the scheme (in bits) is

$$4\lambda + \tau \cdot \left((\underbrace{k}_{x} + \underbrace{r\times m}_{L_U} + \underbrace{r\times m\times\eta}_{\alpha} + \underbrace{m\times\eta}_{c}) \cdot \log_2 q + \underbrace{\lambda \cdot \log_2 N + 2\lambda}_{\text{MPCitH}} \right).$$

Performances and comparison. In what follows, we compare our scheme with the state of the art on the MinRank instance [BESV22]:

$$(q, m, n, k, r) = (16, 16, 16, 142, 4).$$

We provide in Tables 4 and 5 a complete comparison of our scheme with the state of the art. To provide a fair comparison, we propose two variants for [Cou01] and [SINY22]: the first one corresponds to the scheme as described in the original article and the second one is an optimized version. This optimized version includes the following tricks:

- Instead of revealing all the commitments during the first round, the prover just sends a hash digest of them. Then, to enable the verifier to recompute this digest, the prover just needs to send the commitment digests that the verifier can not compute herself.
- The random combination used in the schemes (usually denoted β) is derived from a seed. Then, instead of sending the coefficients of β, the prover can just send this seed. Moreover, this seed and the masks involved in the schemes (usually denote T, S and X) are also derived from a *common* seed.
- Instead of revealing two matrices such that the difference are of rank (at most) r, the prover send one of the matrices and directly the difference (which is cheaper to send), and thus the verifier can deduce the non-sent matrix.

In the comparison we put how [BG22, Section 2] would perform if we apply the same technique for MinRank problem ([BG22] does not consider the MinRank problem in their article).

First, let us remark that [SINY22] presents no advantage compared to [Cou01]. The soundness error of each iteration is 1/2 instead of 2/3, but each iteration is more expensive. The achieved communication cost is thus equivalent to [Cou01]. [BESV22] is a protocol with helper [Beu20]. The components in the proof transcript are the same as for [Cou01] (and [SINY22]), but it succeeds to achieve a bit smaller signature size just by sending a smaller number of seeds and digests. The MPC-in-the-Head paradigm enables to obtain much smaller sizes. Using techniques from [BG22], the resulting size is around 10 KB. In an independent work, [ARZV22] proposes recently a new scheme using techniques which are similar to our protocol with $\Pi_{\text{RC-RD}}$: they are working on another matrix relation[5] but use a less efficient matrix multiplication checking protocol. They succeed to produces signature with sizes below 8 KB. Our scheme with $\Pi_{\text{RC-RD}}$ achieves similar sizes than [ARZV22], but our scheme with $\Pi_{\text{RC-LP}}$ outperforms all the previous ones achieving sizes below 6 KB. For the sake of completeness, we put in the comparison tables how [ARZV22] would perform if we use Π_{MM} as subroutine.

Scheme Name	Security	Signature Size
[Cou01]	$(3/2)^\tau$	$3\tau \cdot \mu_{\text{dig}} + \tau\left[\frac{2}{3}\mu_{\text{mat}} + \frac{2}{3}\mu_{\text{combi}} + \frac{2}{3}\mu_{\text{seed}}\right]$
[Cou01], opt.	$(3/2)^\tau$	$\mu_{\text{dig}} + \tau\left[\frac{1}{3}(\mu_{\text{mat}} + \mu_{\text{rank}} + \mu_{\text{combi}} + 2\mu_{\text{seed}}) + \mu_{\text{dig}}\right]$
[SINY22]	2^τ	$6\tau \cdot \mu_{\text{dig}} + \tau\left[\mu_{\text{mat}} + \frac{1}{2}\mu_{\text{combi}} + \frac{10}{4}\mu_{\text{seed}}\right]$
[SINY22], opt.	2^τ	$\mu_{\text{dig}} + \tau\left[\frac{1}{2}(\mu_{\text{mat}} + \mu_{\text{rank}} + \mu_{\text{combi}} + 3\mu_{\text{seed}}) + 2\mu_{\text{dig}}\right]$
[BESV22]	$\varepsilon_{\text{helper}}(\tau, M, \frac{1}{2})^{-1}$	$\mu_{\text{dig}} + \tau\left[\frac{1}{2}(\mu_{\text{mat}} + \mu_{\text{rank}} + \mu_{\text{combi}} + \mu_{\text{seed}}) + \mu_{\text{dig}} + \mu_{\text{helper}}\right]$
[BG22]	$\varepsilon_{\text{helper}}(\tau, M, \frac{1}{N})^{-1}$	$\mu_{\text{dig}} + \tau\left[\mu_{\text{combi}} + \mu_{\text{rank}} + \mu_{\text{MPCitH}} + \mu_{\text{helper}}\right]$
[ARZV22]	$\text{KZ}(\frac{1}{q^n}, \frac{1}{N})$	$2\mu_{\text{dig}} + \tau\left[\mu_{\text{combi}} + (n^2 + 2rn - r^2)\log_2 q + \mu_{\text{MPCitH}}\right]$
[ARZV22]+Π_{MM}	$\text{KZ}(\frac{1}{q^\eta}, \frac{1}{N})$	$2\mu_{\text{dig}} + \tau\left[\mu_{\text{combi}} + (r(n-r) + \eta(n-2r))\log_2 q + \mu_{\text{MPCitH}}\right]$
Our scheme (RD)	$\text{KZ}(\frac{1}{q^\eta}, \frac{1}{N})$	$2\mu_{\text{dig}} + \tau\left[\mu_{\text{combi}} + \mu_{\text{rank}} + \eta(n+r)\log_2 q + \mu_{\text{MPCitH}}\right]$
Our scheme (LP)	$\text{KZ}(\frac{2}{q^{m\eta}} - \frac{1}{q^{2m\eta}}, \frac{1}{N})$	$2\mu_{\text{dig}} + \tau\left[\mu_{\text{combi}} + rm\log_2 q + \eta(r+1)m\log_2 q + \mu_{\text{MPCitH}}\right]$

Table 4: Sizes of the signatures relying on the MinRank problem (restricting to the schemes using the FS heuristics). The used notations are: $\mu_{\text{mat}} := mn\log 2q$, $\mu_{\text{rank}} := r(m+n)\log_2 q$, $\mu_{\text{combi}} := k\log_2 q$, plus all the notations defined in Section 3.

---

[5] They express the $m - r$ last columns *w.r.t.* the r first ones.

Instance	Protocol Name	Variant	Parameters				Signature Size
			N	M	τ	η	
$q=16$ $m=16$ $n=16$ $k=142$ $r=4$	[Cou01]	-	-	-	219	-	52 430 B
		Optimized	-	-	219	-	28 575 B
	[SINY22]	-	-	-	128	-	50 640 B
		Optimized	-	-	128	-	28 128 B
	[BESV22]	-	-	256	128	-	26 405 B
	[BG22]	Fast	8	187	49	-	13 644 B
		Short	32	389	28	-	10 937 B
	[ARZV22]	Fast	32	-	28	-	10 116 B
		Short	256	-	18	-	7 422 B
	[ARZV22]+Π_{MM}	Fast	32	-	33	9	8 155 B
		Short	256	-	19	9	6 277 B
	Our scheme (RD)	Fast	32	-	33	5	9 288 B
		Short	256	-	19	9	7 122 B
	Our scheme (LP)	Fast	32	-	28	1	7 204 B
		Short	256	-	18	1	**5 518 B**

Table 5: Comparison of the signatures relying on the MinRank problem (restricting to the schemes using the FS heuristics). Numerical comparison.

5.3 Proof of Knowledge for Rank SD

We want to build a zero-knowledge proof of knowledge for the *rank syndrome decoding problem*:

Definition 3 (Rank Syndrome Decoding Problem - Standard Form). *Let $\mathbb{F}_{q^m}$ be the finite field with q^m elements. Let (n, k, r) be positive integers such that $k \leq n$. The rank syndrome decoding problem with parameters (q, m, n, k, r) is the following problem:*

Let H, x and y be such that:

1. *H is uniformly sampled from $\{(H'|I_{n-k}), H' \in \mathbb{F}_{q^m}^{(n-k)\times n}\}$,*
2. *x is uniformly sampled from $\{x \in \mathbb{F}_{q^m}^n : \mathrm{wt}_R(x) \leq r\}$,*
3. *y is built as $y := Hx$.*

From (H, y), find x.

Remark 3. The rank $\mathrm{wt}_R(x)$ of an element x of $\mathbb{F}_{q^m}^n$ is the dimension of the $\mathbb{F}_q$-linear subspace spanned by $x_1, \ldots, x_n$. Equivalently, it is the rank of the matrix M for which the rows are $x_1, \ldots, x_n$ represented as vectors of $\mathbb{F}_q^m$.

The prover wants to convince the verifier that she knows such an x, *i.e.* a vector $x \in \mathbb{F}_{q^m}^n$ such that $y = Hx$ and $\mathrm{wt}_R(x) \leq r$. Previous works propose proofs of knowledge where the constraint on the weight is an equality, but it is sometimes easier to just prove an inequality (see [FJR22] for the case of the Hamming weight). To proceed, the prover will first share the secret vector x and then use an MPC protocol which verifies that this vector satisfies the above property.

Remark 4. In the above definition, the parity-check matrix is in *standard form*. It does not decrease the hardness of the problem (since the transformation into a standard form is a polynomial transformation), but it enables to simplify the contruction we propose.

MPC Protocol. We want to build an MPC protocol which takes as input (a sharing of) x and which outputs

$$\begin{cases} \textsc{Accept} & \text{if } y = Hx \text{ and } \mathrm{wt}_R(x) \leq r \\ \textsc{Reject} & \text{otherwise.} \end{cases}$$

Since H is in standard form, having the equality $y = Hx$ is equivalent to define x as

$$\begin{pmatrix} x_A \\ y - H'x_A \end{pmatrix}$$

for some $x_A \in \mathbb{F}_q^k$. Therefore, we will build an MPC protocol which takes as input (a sharing of) x_A and which outputs

$$\begin{cases} \textsc{Accept} \text{ if } \mathrm{wt}_R(x) \leq r \text{ where } x := \begin{pmatrix} x_A \\ y - H'x_A \end{pmatrix} \\ \textsc{Reject} \text{ otherwise.} \end{cases}$$

Given $\llbracket x_A \rrbracket$, the parties can locally build $\llbracket x_B \rrbracket$ as $\llbracket x_B \rrbracket := y - H'\llbracket x_A \rrbracket$, and so they can deduce a sharing $\llbracket x \rrbracket$ of x (simply by concatenating the shares of $\llbracket x_A \rrbracket$ with the shares of $\llbracket x_B \rrbracket$). It remains to check that $\llbracket x \rrbracket$ corresponds to the sharing of a vector of $\mathbb{F}_{q^m}^n$ of rank at most r. The latter can be done using one of the two rank checking protocols described in Section 5.1: $\Pi^{\eta}_{\text{RC-RD}}$ relying on the rank decomposition or $\Pi^{\eta}_{\text{RC-LP}}$ relying on linearized polynomials, for some parameter η.

The complete MPC protocol is described in Figure 5 when relying on the rank decomposition and in Figure 6 when relying on linearized polynomials.

Public values: $H = (H'|I_{n-k}) \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and $y \in \mathbb{F}_{q^m}^{n-k}$.

Inputs: Each party takes a share of the following sharings as inputs: $\llbracket x_A \rrbracket$ where $x_A \in \mathbb{F}_{q^m}^k$, $\llbracket T \rrbracket$ where $T \in \mathbb{F}_q^{n\times r}$, $\llbracket R \rrbracket$ where $R \in \mathbb{F}_q^{r\times m}$, $\llbracket a \rrbracket$ where a has been uniformly sampled from $\mathbb{F}_q^{r\times\eta}$, and $\llbracket c \rrbracket$ where $c \in \mathbb{F}_q^{n\times\eta}$, such that $c = Ta$ and $X = TR$ where X is the matrix form of x.

MPC Protocol:

1. The parties get a random $\Sigma \in \mathbb{F}_q^{m\times\eta}$.
2. The parties locally set $\llbracket x_B \rrbracket = y - H'\llbracket x_A \rrbracket$.
3. The parties locally write $\llbracket x \rrbracket := (\llbracket x_A \rrbracket, \llbracket x_B \rrbracket)$ as a matrix $\llbracket X \rrbracket$.
4. The parties locally set $\llbracket \alpha \rrbracket = \llbracket R \rrbracket \Sigma + \llbracket a \rrbracket$.
5. The parties open $\alpha \in \mathbb{F}_q^{r\times\eta}$.
6. The parties locally set $\llbracket v \rrbracket = \llbracket T \rrbracket \alpha - \llbracket c \rrbracket - \llbracket X \rrbracket \Sigma$.
7. The parties open $v \in \mathbb{F}_q^{m\times\eta}$.
8. The parties outputs $\textsc{Accept}$ if $v = 0$ and $\textsc{Reject}$ otherwise.

Fig. 5: An MPC protocol based on the *rank decomposition* technique ($\Pi_{\text{RC-RD}}$) which verifies that the given input corresponds to a solution of a rank syndrome decoding problem.

Proof of Knowledge. Using the MPC-in-the-Head paradigm (see Section 2.1), we transform the above MPC protocol into an interactive zero-knowledge proof of knowledge which enables to convince a verifier that a prover knows the solution of a rank syndrome decoding problem. The soundness error of the resulting protocol is

$$\varepsilon := \frac{1}{N} + \left(1 - \frac{1}{N}\right) p_\eta$$

where $p_\eta := \frac{1}{q^\eta}$ when using $\Pi^{\eta}_{\text{RC-RD}}$ and $p_\eta := \frac{2}{q^{m\cdot\eta}} - \frac{1}{q^{2\cdot m\cdot\eta}}$ when using $\Pi^{\eta}_{\text{RC-LP}}$. By repeating the protocol τ times, we get a soundness error of ε^τ. To obtain a soundness error of λ bits, we can take $\tau = \left\lceil \frac{-\lambda}{\log_2 \varepsilon} \right\rceil$. We can transform the interactive protocol into a non-interactive proof / signature thanks to the Fiat-Shamir transform [FS87]. According to [KZ20], the security of the resulting scheme is

$$\text{cost}_{\text{forge}} := \min_{\tau_1,\tau_2:\tau_1+\tau_2=\tau} \left\{ \frac{1}{\sum_{i=\tau_1}^{\tau} \binom{\tau}{i} p_\eta^i (1-p_\eta)^{\tau-i}} + N^{\tau_2} \right\}.$$

When using $\Pi_{\text{RC-RD}}$, the communication cost of the scheme (in bits) is

$$4\lambda + \tau \cdot \left((\underbrace{k \cdot m}_{x_A} + \underbrace{r \times m}_{R} + \underbrace{r \times n}_{T} + \underbrace{r \times \eta}_{\alpha} + \underbrace{n \times \eta}_{c}) \cdot \log_2 q + \underbrace{\lambda \cdot \log_2 N + 2\lambda}_{\text{MPCitH}} \right)$$

Public values: $H = (H'|I_{n-k}) \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and $y \in \mathbb{F}_{q^m}^{n-k}$.

Inputs: Each party takes a share of the following sharings as inputs: $[\![x_A]\!]$ where $x \in \mathbb{F}_{q^m}^k$, $[\![L_U]\!] := X^{q^r} + \sum_{i=0}^{r-1} [\![\beta_i]\!] X^{q^i}$ where $L_U(X) := \prod_{u \in U}(X-u) \in \mathbb{F}_{q^m}[X]$, $[\![a]\!]$ where a has been uniformly sampled from $\mathbb{F}_{q^{m\cdot\eta}}^r$, and $[\![c]\!] \in \mathbb{F}_{q^{m\cdot\eta}}$, such that $c = -\langle \beta, a\rangle$.

MPC Protocol:

1. The parties get random $\gamma_1, \dots, \gamma_n \in \mathbb{F}_{q^{m\cdot\eta}}$.
2. The parties get a random $\varepsilon \in \mathbb{F}_{q^{m\cdot\eta}}$.
3. The parties locally set $[\![x_B]\!] = y - H'[\![x_A]\!]$.
4. The parties locally set $[\![z]\!] = -\sum_{j=1}^n \gamma_j [\![x_j]\!]^{q^r}$.
5. The parties locally set $[\![w_i]\!] = \sum_{j=1}^n \gamma_j [\![x_j]\!]^{q^i}$ for all $i \in \{0, \dots, r-1\}$.
6. The parties locally set $[\![\alpha]\!] = \varepsilon \cdot [\![w]\!] + [\![a]\!]$.
7. The parties open $\alpha \in \mathbb{F}_{q^{m\cdot\eta}}^r$.
8. The parties locally set $[\![v]\!] = \varepsilon \cdot [\![z]\!] - \langle \alpha, [\![\beta]\!]\rangle - [\![c]\!]$.
9. The parties open $v \in \mathbb{F}_{q^{m\cdot\eta}}$.
10. The parties outputs ACCEPT if $v = 0$ and REJECT otherwise.

Fig. 6: An MPC protocol based on the technique using *linearized polynomials* ($\Pi_{\text{RC-LP}}$) which verifies that the given input corresponds to a solution of a rank syndrome decoding problem. U is a $\mathbb{F}_q$-linear subspace U of $\mathbb{F}_{q^m}$ of dimension r which contains $x_1, \dots, x_n$.

where λ is the security level, η is a scheme parameter and τ is computed such that the soundness error is of λ bits in the interactive case and such that $\text{cost}_{\text{forge}}$ is of λ bits in the non-interactive case.

And when using $\Pi_{\text{RC-LP}}$, the communication cost of the scheme (in bits) is

$$4\lambda + \tau \cdot \left((\underbrace{k \cdot m}_{x_A} + \underbrace{r \times m}_{L_U} + \underbrace{r \times m \times \eta}_{\alpha} + \underbrace{m \times \eta}_{c}) \cdot \log_2 q + \underbrace{\lambda \cdot \log_2 N + 2\lambda}_{\text{MPCitH}} \right).$$

Performances and comparison. In what follows, we compare our scheme with the state of the art on the Rank Syndrome Decoding instance [BG22]:

$$(q, m, n, k, r) = (2, 32, 30, 14, 9).$$

We provide in Tables 6 and 7 a complete comparison of our scheme with the state of the art. To get a more complete comparison, we include the schemes [Ste94], [Vér96] and [FJR21] which can be easily adapted for the rank metric (by replacing the permutations by rank isometries). Moreover, we put in Table 8 the achieved performances of [BG22] when relying on structured rank syndrome decoding problem (the parameters of the structured problem come from the original article).

The first schemes [Ste94] and [Vér96] can achieve signature sizes of around 30 KB (let us remark that some optimization tricks have been used to achieve these sizes). Then, using the MPC-in-the-Head technique of the "shared permutation", [FJR21] and [BG22] divide this size by half, achieving communication cost around 15 KB (13 − 19 KB). Finally, our new schemes outperform all these schemes by achieving sizes around 6 − 11 KB. The scheme using a q-polynomial even outperforms the [BG22]'s proposals[6] which rely on structured rank syndrome decoding problems.

Remark 5. Let us focus on the zero-knowledge proof relying on linearized polynomials. Thanks to the structure of the MPC protocol, it is possible to use Shamir's secret sharings over $\mathbb{F}_{q^m}$ instead of additive sharings (even if the base field is $\mathbb{F}_q$ due to the $\mathbb{F}_q$-linearity of the Frobenius endomorphism). We describe in Appendix A how the MPC protocol behaves when using Shamir's secret sharing. As explained in [FR22], using such sharings reduces the computational cost of emulating the MPC protocol and enables to have fast signature verifications. The fact that we can share values over $\mathbb{F}_{q^m}$ implies that we can use techniques from [FR22],

---

[6] Theses sizes are larger than the ones in [BG22] because they take $N = 1024$, but here to have a fair comparison with the other schemes, we take $N = 256$.

Scheme Name	Security	Signature Size
[Ste94]	$(3/2)^\tau$	$\mu_{\text{dig}} + \tau\left[\frac{1}{3}(2\mu_{\text{mat}} + \mu_{\text{rank}} + 2\mu_{\text{seed}}) + \mu_{\text{dig}}\right]$
[Vér96]	$(3/2)^\tau$	$\mu_{\text{dig}} + \tau\left[\frac{1}{3}(\mu_{\text{mat}} + \mu_{\text{ptx}} + \mu_{\text{rank}} + 2\mu_{\text{seed}}) + \mu_{\text{dig}}\right]$
[FJR21]	$\varepsilon_{\text{helper}}(\tau, M, \frac{1}{N})^{-1}$	$\mu_{\text{dig}} + \tau\left[\mu_{\text{mat}} + \mu_{\text{ptx}} + \mu_{\text{rank}} + \mu_{\text{MPCitH}} + \mu_{\text{helper}}\right]$
[BG22]	$\varepsilon_{\text{helper}}(\tau, M, \frac{1}{N})^{-1}$	$\mu_{\text{dig}} + \tau\left[\mu_{\text{mat}} + \mu_{\text{rank}} + \mu_{\text{MPCitH}} + \mu_{\text{helper}}\right]$
Our scheme (RD)	$\text{KZ}(\frac{1}{q^\eta}, \frac{1}{N})$	$2\mu_{\text{dig}} + \tau\left[\mu_{\text{ptx}} + \mu_{\text{rank}} + \eta(n+r)\log_2 q + \mu_{\text{MPCitH}}\right]$
Our scheme (LP)	$\text{KZ}(\frac{2}{q^{m\cdot\eta}} - \frac{1}{q^{2\cdot m\cdot\eta}}, \frac{1}{N})$	$2\mu_{\text{dig}} + \tau\left[\mu_{\text{ptx}} + rm\log_2 q + \eta(r+1)m\log_2 q + \mu_{\text{MPCitH}}\right]$

Table 6: Sizes of the signatures relying on the rank syndrome decoding problem (restricting to the schemes using the FS heuristics). The used notations are: $\mu_{\text{mat}} := mn\log 2q$, $\mu_{\text{rank}} := r(m+n)\log_2 q$, $\mu_{\text{ptx}} := mk\log_2 q$, plus all the notations defined in Section 3.

Instance	Protocol Name	Variant	Parameters				Signature Size
			N	M	τ	η	
$q=2$ $m=31$ $n=30$ $k=15$ $r=9$	Stern [Ste94]	-	-	-	219	-	31 358 B
	Véron [Vér96]	-	-	-	219	-	27 115 B
	[FJR21]	Fast	8	187	49	-	19 328 B
		Short	32	389	28	-	14 181 B
	[BG22]	Fast	8	187	49	-	15 982 B
		Short	32	389	28	-	12 274 B
	Our scheme (RD)	Fast	32	-	33	19	11 000 B
		Short	256	-	21	24	8 543 B
	Our scheme (LP)	Fast	32	-	30	1	7 376 B
		Short	256	-	20	1	**5 899 B**

Table 7: Sizes of the signatures relying on the rank syndrome decoding problem (restricting to the schemes using the FS heuristics). Numerical comparison.

Protocol Name	Structure	Variant	Parameters		Signature Size
			N	τ	
[BG22]	Ideal RSD	Fast	32	37	12 607 B
		Short	256	26	10 126 B
[BG22]	Ideal RSL	Fast	32	27	9 392 B
		Short	256	17	6 754 B

Table 8: Sizes of the signatures relying on the structured rank syndrome decoding problem (restricting to the schemes using the FS heuristics).

with a very large number N of parties (N is upper bounded by q^m). This is not true for the zero-knowledge proof relying on the rank decomposition, or for both zero-knowledge proofs about the MinRank problem. For those proofs, the number N of parties would be upper bounded by q, which is small when considering concrete instances.

Remark 6. It is possible to transform a proof of knowledge for rank syndrome decoding (RSD) problem in a proof of knowledge for sum-rank syndrome decoding (SRSD) problem. The latter consists, given (H, y), in finding $x \in \mathbb{F}^n$ such that $y = Hx$ and

$$\text{wt}_{SR}(x) \leq r$$

where $\text{wt}_{SR}((x_1, \ldots, x_{n/\ell})) := \sum_{i=0}^{n/\ell} \text{wt}_R(x_i)$ with ℓ a SRSD parameter and with $x_i \in \mathbb{F}^\ell$. Let us denote $X, X_1, \ldots, X_{\frac{n}{\ell}}$ the matrix form of $x, x_1, \ldots, x_{\frac{n}{\ell}}$. Proving that x (or equivalently X) satisfies $\text{wt}_{SR}(x) \leq r$

can be done by proving that the matrix

$$\begin{pmatrix} X_1 & & & \\ & X_2 & & \\ & & \ddots & \\ & & & X_{\frac{n}{\ell}} \end{pmatrix}$$

has a rank of at most r. Thus all proofs for RSD can be used for SRSD, but they must handle a large matrix. We propose in Appendix B another MPC protocol to check that $\mathrm{wt}_{SR}(x) \leq r$, which does not rely on the above transformation. The core idea of this protocol is to transform x into a vector d (using Π_{MM}) such that

$$\mathrm{wt}_{SR}(x) = \mathrm{wt}_H(d).$$

Then using the MPC protocol of [FJR22], we can check that $\mathrm{wt}_H(d) \leq r$ and thus we get the desired inequality.

6 Proof of Knowledge for Permuted Kernel Problem

We want to build a zero-knowledge proof of knowledge for the *permuted kernel problem*:

Definition 4 (Inhomogenous Permuted Kernel Problem). *Let $\mathbb{F}_q$ be the finite field with q elements. Let m and n be positive integers. The permuted kernel problem with parameters (q, m, n) is the following problem:*

> *Let H, y, v and σ be such that:*
> 1. *H is uniformly sampled from $\mathbb{F}_q^{m \times n}$,*
> 2. *v is uniformly sampled from $\mathbb{F}_q^n$,*
> 3. *σ is a random permutation of $[n]$,*
> 4. *y is built as $y := H\sigma(v)$.*
>
> *From (H, y, v), find σ.*

The prover wants to convince the verifier that she knows a permutation σ such that $y = H\sigma(v)$. Sharing the permutation seems the natural strategy, all the previous works adopt it. However, implementing permutations in a secure way (secure against timing and cache attacks) is a tricky exercise. We propose here a new proof of knowledge which has a larger communication cost, but which has the advantage of not relying on permutations. To proceed, the prover will first share the secret vector $x := \sigma(v)$ and then use an MPC protocol which verifies that this vector satisfies the desired property.

MPC Protocol. We want to build an MPC protocol which takes as input (a sharing of) $x := \sigma(v)$ and which outputs

$$\begin{cases} \textsc{Accept} & \text{if } y = Hx \text{ and } \exists \sigma : x = \sigma(v) \\ \textsc{Reject} & \text{otherwise.} \end{cases}$$

Proving that $y = Hx$ is easy since it is linear. The hard part is to prove that there exists a permutation between x and v, without using any permutation. To proceed, we will check that the two following polynomials are equal:

$$P(X) = (X - x_1) \dots (X - x_n) \quad \text{and} \quad Q(X) = (X - v_1) \dots (X - v_n).$$

If they are equal, it means that they have the same roots, and thus we can deduce that $x := (x_1, \dots, x_n)$ and $v := (v_1, \dots, v_n)$ are equal up to the order of their coordinates. In practice, to check that that $P(X)$ and $Q(X)$ are equal, we will rely on the Schwartz-Zippel Lemma: we sample a random evaluation point ξ in the field extension $\mathbb{F}_{q^{\eta_1}}$ (for some positive integer η_1) and we check that $P(\xi)$ is equal to $Q(\xi)$. If the two polynomials are not equal, the probability to get $P(\xi) = Q(\xi)$ is upper bounded by

$$\frac{n}{|\mathbb{F}_{q^{\eta_1}}|}$$

since n is the degree of $P(X) - Q(X)$. Thus, the MPC protocol will compute $P(\xi) = (\xi - x_1)\dots(\xi - x_n)$ from a sharing $[\![x]\!]$ of x and will compare the result with $Q(\xi)$.

Given an evaluation point ξ, let us denote

$$\begin{aligned} s_1 &:= (\xi - x_1) \\ s_2 &:= (\xi - x_1)(\xi - x_2) \\ &\vdots \\ s_n &:= (\xi - x_1)(\xi - x_2)\dots(\xi - x_n). \end{aligned}$$

The MPC protocol could proceed as follows:

1. The parties get a random evaluation point $\xi \in \mathbb{F}_{q^{\eta_1}}$.
2. The parties get as hints $[\![s_1]\!]$, ..., $[\![s_{n-1}]\!]$ (which depend on ξ).
3. The parties execute a multiplication checking protocol to check that

$$\forall i \in \{1, \dots, n-1\},\ s_i \cdot x_{i+1} = s_{i+1}$$

where $s_n := Q(\xi)$.

However, all existing multiplication checking protocols induce a communication cost which depends on the bitsize of the multiplication triples. In what follows, we assume that n is even. Let us define

$$\begin{aligned} t_1 &:= x_1 \cdot x_2 \\ t_2 &:= x_3 \cdot x_4 \\ &\vdots \\ t_{n/2} &:= x_{n-1} \cdot x_n \end{aligned}$$

To save communication, the MPC protocol we consider will proceed as follows:

1. The parties get a random evaluation point $\xi \in \mathbb{F}_{q^{\eta_1}}$.
2. The parties get as hints $[\![t_1]\!]$, ..., $[\![t_{n/2}]\!]$ which live in $\mathbb{F}_q$.
3. The parties get as hints $[\![s_4]\!]$,$[\![s_6]\!]$, ..., $[\![s_{n-2}]\!]$ which live in $\mathbb{F}_{q^{\eta_1}}$.
4. The parties execute a multiplication checking protocol to check that

$$\forall i \in \{1, \dots, n/2\},\ x_{2i-1} \cdot x_{2i} = t_i.$$

5. The parties execute a multiplication checking protocol to check that

$$\forall i \in \{2, \dots, n/2\},\ s_{2i-2} \cdot \left(\xi^2 - (x_{2i-1} + x_{2i})\xi + t_i\right) = s_{2i}$$

where $s_2 := (\xi^2 - (x_1 + x_2)\xi + t_1)$ and $s_n := Q(\xi)$.

Since t_i's bitsize is η_1 times smaller than s_i's, the communication cost of this MPC protocol is smaller than the previous one.

The MPC protocol is completely described in Figure 7. As batch multiplication checking protocol, we use the MPC protocol Π_{BMC} described in Figure 8 (inspired from [BdK+21]).

Proof of Knowledge. Using the MPC-in-the-Head paradigm (see [FR22, Theorem 2]), we transform the above MPC protocol into an interactive 7*-round* zero-knowledge proof of knowledge which enables to convince a verifier that a prover knows the solution of a permuted kernel problem. The soundness error of the resulting protocol is

$$\varepsilon := \frac{1}{N} + \left(1 - \frac{1}{N}\right) p_{\eta_1,\eta_2}$$

Public values: $H \in \mathbb{F}_q^{m \times n}$, $y \in \mathbb{F}_q^m$ and $v \in \mathbb{F}_q^n$.

Inputs: Each party takes a share of the following sharings as inputs: $\llbracket x \rrbracket$ where $x \in \mathbb{F}_q^n$.

MPC Protocol:

1. The parties get a random $\xi \in \mathbb{F}_{q^{\eta_1}}$.
2. The parties get as hints $\llbracket t_1 \rrbracket, \dots, \llbracket t_{n/2} \rrbracket$ where
$$\forall i \in \{1, \dots, \frac{n}{2}\},\ t_i = x_{2i-1} \cdot x_{2i}.$$
3. The parties get as hints $\llbracket s_4 \rrbracket$, $\llbracket s_6 \rrbracket$, ..., $\llbracket s_{n-2} \rrbracket$ where
$$\forall i \in \{2, \dots, \frac{n}{2} - 1\},\ s_{2i} = s_{2i-2} \cdot (\xi - x_{2i-1}) \cdot (\xi - x_{2i})$$
with $s_2 := (\xi - x_1)(\xi - x_2)$.
4. The parties execute in parallel the MPC protocols
$$\llbracket v_1 \rrbracket \leftarrow \mathit{\Pi}_{\mathrm{BMC}}^{\eta_1 \cdot \eta_2} \begin{pmatrix} \llbracket x_1 \rrbracket, & \llbracket x_2 \rrbracket, & \llbracket t_1 \rrbracket \\ \llbracket x_3 \rrbracket, & \llbracket x_4 \rrbracket, & \llbracket t_2 \rrbracket \\ & \vdots & \\ \llbracket x_{n-1} \rrbracket, & \llbracket x_n \rrbracket, & \llbracket t_{\frac{n}{2}} \rrbracket \end{pmatrix}$$
and
$$\llbracket v_2 \rrbracket \leftarrow \mathit{\Pi}_{\mathrm{BMC}}^{\eta_2} \begin{pmatrix} \llbracket s_2 \rrbracket, & \xi^2 - (\llbracket x_3 \rrbracket + \llbracket x_4 \rrbracket) \cdot \xi + \llbracket t_2 \rrbracket, & \llbracket s_4 \rrbracket \\ \llbracket s_4 \rrbracket, & \xi^2 - (\llbracket x_5 \rrbracket + \llbracket x_6 \rrbracket) \cdot \xi + \llbracket t_3 \rrbracket, & \llbracket s_6 \rrbracket \\ & \vdots & \\ \llbracket s_{n-4} \rrbracket, & \xi^2 - (\llbracket x_{n-3} \rrbracket + \llbracket x_{n-2} \rrbracket) \cdot \xi + \llbracket t_{\frac{n}{2}-1} \rrbracket, & \llbracket s_{n-2} \rrbracket \\ \llbracket s_{n-2} \rrbracket, & \xi^2 - (\llbracket x_{n-1} \rrbracket + \llbracket x_n \rrbracket) \cdot \xi + \llbracket t_{\frac{n}{2}} \rrbracket, & Q(\xi) \end{pmatrix}$$
where $\llbracket s_2 \rrbracket = \xi^2 - (\llbracket x_1 \rrbracket + \llbracket x_2 \rrbracket) \cdot \xi + \llbracket t_1 \rrbracket$.
5. The parties open v_1 and v_2.
6. The parties locally set $\llbracket v_0 \rrbracket = H \llbracket x \rrbracket$, and they open v_0.
7. The parties outputs Accept if $v_0 = y$, $v_1 = 0$ and $v_2 = 0$, and Reject otherwise.

Fig. 7: An MPC protocol which verifies that the given input corresponds to a solution of a permuted kernel problem.

where
$$p_{\eta_1,\eta_2} := 1 - \left(1 - \frac{n}{q^{\eta_1}}\right)\left(1 - \frac{n-1}{q^{\eta_1 \cdot \eta_2}}\right)\left(1 - \frac{1}{q^{\eta_1 \cdot \eta_2}}\right).$$

By repeating the protocol τ times, we get a soundness error of ε^τ. To obtain a soundness error of λ bits, we can take $\tau = \left\lceil \frac{-\lambda}{\log_2 \varepsilon} \right\rceil$. We can transform the interactive protocol into a non-interactive proof / signature thanks to the Fiat-Shamir transform [FS87]. According to [KZ20] (adapted for 7-round proof), the security of the resulting scheme is

$$\mathrm{cost}_{\mathrm{forge}} := \min_{\tau_1 + \tau_2 + \tau_3 = \tau} \left\{ \frac{1}{\mathrm{SPMF}(\tau, \tau_1, p_1)} + \frac{1}{\mathrm{SPMF}(\tau - \tau_1, \tau_2, p_2)} + N^{\tau_2} \right\}$$

where $\mathrm{SPMF}(\tau, \tau', p) := \sum_{i=\tau'}^{\tau} \binom{\tau}{i} p^i (1-p)^{\tau - i}$ and

$$p_1 := \frac{n}{q^{\eta_1}},$$
$$p_2 := 1 - \left(1 - \frac{n-1}{q^{\eta_1 \cdot \eta_2}}\right)\left(1 - \frac{1}{q^{\eta_1 \cdot \eta_2}}\right).$$

Inputs: Each party takes a share of the following sharings as inputs:

$$\begin{pmatrix} [\![r_1]\!],\ [\![s_1]\!],\ [\![t_1]\!] \\ \vdots \\ [\![r_n]\!],\ [\![s_n]\!],\ [\![t_n]\!] \end{pmatrix}$$

MPC Protocol:

1. The parties get as hints $[\![a]\!]$, $[\![b]\!]$ and $[\![c]\!]$ where a and b are uniformly random in $\mathbb{K}$ and $c = a \cdot b$.
2. The parties locally build the polynomials $[\![R]\!]$ and $[\![S]\!]$ such that
$$\forall i \in [n], \begin{cases} [\![R]\!](\gamma_i) = [\![r_i]\!] \\ [\![S]\!](\gamma_i) = [\![s_i]\!] \end{cases} .$$
3. The parties get as hints $[\![t_{n+1}]\!], \ldots, [\![t_{2n-1}]\!]$ where
$$\forall i \in [n-1], t_{n+i} = (R \cdot S)(\gamma_{n+i}).$$
4. The parties locally build the polynomial $[\![T]\!]$ such that
$$\forall i \in [2n-1], [\![T]\!](\gamma_i) = [\![t_i]\!].$$
5. The parties get random $r, \varepsilon \in \mathbb{K}$.
6. The parties locally set $[\![\alpha]\!] = \varepsilon \cdot [\![R]\!](r) + [\![a]\!]$ and $[\![\beta]\!] = [\![S]\!](r) + [\![b]\!]$.
7. The parties open $\alpha, \beta \in \mathbb{K}$.
8. The parties locally set $[\![v]\!] = \varepsilon \cdot [\![T]\!](r) - [\![c]\!] + \alpha \cdot [\![b]\!] + \beta \cdot [\![a]\!] - \alpha \cdot \beta$.

Fig. 8: An MPC protocol $\Pi^{\eta}_{\mathrm{BMC}}$ which verifies that, for all $i \in [n]$, $r_i \cdot s_i = t_i$, where all (r_i, s_i, t_i)'s belong to a field $\mathbb{F}$. Let us denote $\mathbb{K}$ the field extension of degree η. $\gamma_1, \ldots, \gamma_{2n-1}$ are *distinct* elements of $\mathbb{F}$ (we assume that $|\mathbb{F}| \geq 2n-1$).

The communication cost[7] of the scheme (in bits) is

$$4\lambda + \tau \cdot \left(\underbrace{(n-m)}_{x_A} \cdot \log_2 q + \mu_{\mathrm{misc}} + \underbrace{\lambda \cdot \log_2 N + 2\lambda}_{\mathrm{MPCitH}} \right)$$

with
$$\mu_{\mathrm{misc}} := (\underbrace{\frac{n}{2} + \eta_1 \left(\frac{n}{2} - 2\right)}_{t_1,\ldots,t_{\frac{n}{2}}, s_4, \ldots, s_{n-2}} + \underbrace{\left(\frac{n}{2} - 1\right) + \eta_1 \left(\frac{n}{2} - 2\right)}_{T} + \underbrace{5\eta_1\eta_2}_{\alpha,\beta,c}) \cdot \log_2 q$$

where λ is the security level, (η_1, η_2) are scheme parameters and τ is computed such that the soundness error is of λ bits in the interactive case and such that $\mathrm{cost}_{\mathrm{forge}}$ is of λ bits in the non-interactive case.

Performances and comparison. In what follows, we compare our scheme with the state of the art on the permuted kernel instance [Beu20]:
$$(q, n, m) = (997, 61, 28).$$

We provide in Tables 9 and 10 a comparison of our scheme with the state of the art. To get a more complete comparison, we include the schemes [Ste94], [Vér96] and [FJR21] which can be easily adapted for the permuted kernel problem.

The first schemes [Ste94] and [Vér96] can achieve signature sizes of around 20 − 25 KB (let us remark that some optimization tricks have been used to achieve these sizes). Then, using a *protocol with helper*, Beullens [Beu20] reduces the sizes to around 15 KB (12−18 KB). Thanks to their MPC-in-the-Head technique

---

[7] The formula of this cost assumes that the matrix is in standard form (as in Section 5.3). We omit this detail in Figure 7 for the sake of simplicity.

of the "shared permutation", [FJR21] achieves similar performances. [BG22] then succeeds to remove the helper from [FJR21] by leveraging the linearity of the permuted kernel problem and thus currently has the best sizes from the state of the art (9 – 10 KB). Our scheme has similar signature sizes than [Beu20] and [FJR21], and is outperformed by [BG22]. However, our scheme presents several advantages:

- instead of using permutations, our scheme works on polynomials, which is easier to securely implement;
- our scheme is more parallelizable since all the parties run computation *in parallel*, whilst the parties in [FJR21] and [BG22] run computation *in series*;
- our scheme is more compatible with existing MPC-in-the-Head techniques. As example, our scheme is compatible with techniques from [FR22] (like fast signature verification) and from [AMGH$^+$22] (see next section), while the previous schemes based on PKP were not.

Scheme Name	Security	Signature Size
[Sha90]	$\mathrm{KZ}(\frac{1}{q}, \frac{1}{2})$	$\tau\,[2\mu_{\mathrm{dig}} + \mu_{\mathrm{mask}} + \mu_{\mathrm{small}}]$
[Ste94]	$(3/2)^{\tau}$	$\mu_{\mathrm{dig}} + \tau\left[\frac{1}{3}(2\mu_{\mathrm{mask}} + \mu_{\mathrm{small}} + 2\mu_{\mathrm{seed}}) + \mu_{\mathrm{dig}}\right]$
[Vér96]	$(3/2)^{\tau}$	$\mu_{\mathrm{dig}} + \tau\left[\frac{1}{3}(\mu_{\mathrm{mask}} + \mu_{\mathrm{ptx}} + \mu_{\mathrm{small}} + 2\mu_{\mathrm{seed}}) + \mu_{\mathrm{dig}}\right]$
PKP-DSS [BFK$^+$19]	$\mathrm{KZ}(\frac{1}{q-1}, \frac{1}{2})$	$\mu_{\mathrm{dig}} + \tau \cdot [\mu_{\mathrm{mask}} + \mu_{\mathrm{dig}} + 2\mu_{\mathrm{seed}}]$
SushyFish [Beu20]	$\varepsilon_{\mathrm{helper}}(\tau, M, \frac{1}{q'})^{-1}$	$\mu_{\mathrm{dig}} + \tau\,[\mu_{\mathrm{mask}} + \mu_{\mathrm{small}} + 2\mu_{\mathrm{seed}} + \mu_{\mathrm{dig}} \cdot \log_2(q') + \mu_{\mathrm{helper}}]$
[FJR21]	$\varepsilon_{\mathrm{helper}}(\tau, M, \frac{1}{N})^{-1}$	$\mu_{\mathrm{dig}} + \tau\,[\mu_{\mathrm{mask}} + \mu_{\mathrm{ptx}} + \mu_{\mathrm{small}} + \mu_{\mathrm{MPCitH}} + \mu_{\mathrm{helper}}]$
[BG22]	$\mathrm{KZ}(\frac{1}{q-1}, \frac{1}{N})$	$\mu_{\mathrm{dig}} + \tau\,[\mu_{\mathrm{mask}} + \mu_{\mathrm{small}} + \mu_{\mathrm{MPCitH}} + \mu_{\mathrm{helper}}]$
Our scheme	$\mathrm{KZ}_3(p_1, p_2, \frac{1}{N})$	$2\mu_{\mathrm{dig}} + \tau\,[\mu_{\mathrm{ptx}} + \mu_{misc} + \mu_{\mathrm{MPCitH}}]$ where $\mu_{misc} := ((n-1)(\eta_1+1) + \eta_1(5\eta_2-3))\log_2 q$

Table 9: Sizes of the signatures relying on the permuted kernel problem (restricting to the schemes using the FS heuristics). The used notations are: $\mu_{\mathrm{mask}} := n \log 2q$, $\mu_{\mathrm{small}} := n \log_2 n$, $\mu_{\mathrm{ptx}} := (n-m)\log_2 q$, plus all the notations defined in Section 3.

Instance	Protocol Name	Variant	Parameters					Signature Size
			N	M	τ	η_1	η_2	
$q = 997$ $n = 61$ $m = 38$	Shamir [Sha90]	-	-	-	149	-	-	27 746 B
	Stern [Ste94]	-	-	-	219	-	-	23 848 B
	Véron [Vér96]	-	-	-	219	-	-	21 272 B
	PKP-DSS [BFK$^+$19]	-	-	-	149	-	-	20 961 B
	SushyFish [Beu20]	Fast	4	191	68	-	-	18 448 B
		Short	128	916	20	-	-	12 145 B
	[FJR21]	Fast	8	187	49	-	-	15 420 B
		Short	32	389	28	-	-	11 947 B
	[BG22]	Fast	32	-	42	-	-	9 896 B
		Short	256	-	31	-	-	**8 813 B**
	Our scheme	Fast	32	-	41	2	2	16 373 B
		Short	256	-	24	3	2	12 816 B

Table 10: Sizes of the signatures relying on the permuted kernel problem (restricting to the schemes using the FS heuristics). Numerical comparison.

7 Performances

To provide a fairer comparison of our work with the state of the art, we need to give an estimation of the computational performances of our proposals. The best way to proceed would be to have optimized implementations for them, but producing such implementations requires dedicated work for each of the proposed signature schemes (and is left for future works). However, we can roughly estimate what would be the running time for our schemes by analyzing the existing implementations of the recent signature scheme proposed by [FJR22] that we refer to as SDitH (for *Syndrome Decoding in the Head*).

Until recently, the only way to implement an MPCitH-based proof system was by emulating all the parties of the underlying MPC protocol, implying that we would need to emulate N times a party per repetition. The recent work [AMGH$^+$22] changes this drastically. The authors suggest generating the input shares of the parties in a correlated way using a hypercube approach. This optimization enables to emulate only $\log_2(N)$ parties per repetition. For example, in Section 4, we propose to take $\tau = 25$ and $N = 256$ for the "short" trade-off of our scheme. Without the optimization of [AMGH$^+$22], we would need to emulate $\tau \cdot N = 6400$ times a party per signing. With it, we just need to emulate $\tau \cdot \log_2 N = 200$ times a party, reducing the computational cost of the MPC emulation by a factor of 32. In the implementation of SDitH in [AMGH$^+$22], there are two computational bottlenecks:

- **Generating the input shares and preparing the commitments.** The computational cost of this part does not depend on the MPC protocol, it just depends on:
  - the security level λ, the number N of parties, and the number τ of repetitions,
  - the size of the input shares, since we need to expand these shares from seeds using a pseudo-random generator,
  - the addition law since we are working on additive sharing.

  Since we repeat the same computation for each repetition, the cost scales linearly into τ. In Table 11, we give the size of the input shares for all the signature schemes we proposed in this work. We can observe that the sizes of the input shares of our proposals are all smaller than for SDitH. In [AMGH$^+$22], for 256 parties, generating the input shares and preparing the commitments takes around 5 milliseconds with $\tau_{\text{SDitH}} := 17$ repetitions. Without a more refined benchmark, we can not know how the sizes of the shares impact these 5 ms, so we estimate the running time for the first bottleneck in our schemes *assuming that the sizes of our shares are equal to the size of the shares of SDitH* (this assumption will overestimate the computational cost).
- **Emulating the MPC protocol.** The computational cost of this part is dominated by the cost of all the multiplications, thus we could use the number of involved multiplications in the MPC protocols as metric to estimate the cost. Unfortunately, since the schemes are not relying on the same base field, it would be hard to interpret the obtained numbers. For example, for schemes based on the rank syndrome decoding problem, one proposal is defined over $\mathbb{F}_2$ and uses a large number of multiplications, while the other is defined over $\mathbb{F}_{2^{31}}$ and uses a much smaller number of multiplications. To estimate the computation cost of emulating the MPC protocol, we make the *conservative assumption that the MPC protocols we proposed in this work are three times heavier* than the MPC protocol involved in SDitH (which was already quite heavy since it uses polynomial interpolations/evaluations). In [AMGH$^+$22], emulating the MPC protocol takes around 2 milliseconds for $\tau_{\text{SDitH}} := 17$ repetitions.

Gathering the estimations for the two computational bottlenecks, we can estimate the running times of our proposals using

$$\textsc{Running Time} \le \tau \cdot \left(\frac{5 \text{ ms}}{\tau_{\text{SDitH}}} + 3 \cdot \frac{2 \text{ ms}}{\tau_{\text{SDitH}}} \right)$$

where τ is the number of repetitions. The obtained timings are given in Table 11. Let us stress that those estimations are overestimated for the reasons given above. In practice, we can expect that (for $N = 256$)

- signing would take *between 10 and 15 milliseconds* for the scheme based on the permuted kernel problem, and
- signing would take *between 5 and 10 milliseconds* for all the other schemes (which are lighter).

Scheme	Size of an input share (in bits)	τ	Signing time
SDitH [FJR22]	2664	17	7 ms [AMGH+22]
MQ over $\mathbb{F}_4$	1244	25	$\leq$ 15.5 ms
MQ over $\mathbb{F}_{256}$	976	25	$\leq$ 15.5 ms
MinRank (with RC)	1800	19	$\leq$ 12.3 ms
MinRank (with LP)	1144	18	$\leq$ 11.6 ms
Rank SD (with RC)	1950	21	$\leq$ 13.6 ms
Rank SD (with LP)	1054	20	$\leq$ 12.9 ms
PKP	2570	24	$\leq$ 15.5 ms

Table 11: Estimation of the computational costs (for $N = 256$) compared to existing implementations of SDitH.

8 Conclusion

In this work, we studied how the MPC-in-the-Head paradigm behaves for the multivariate quadratic problem, the MinRank problem, the rank syndrome decoding problem and the permuted kernel problem.

While a straight application of this paradigm to the *permuted kernel problem* seems to produce schemes with limited performances, it enables to reduce communication cost when considering the *multivariate quadratic problem* on larger fields as $\mathbb{F}_{256}$.

The main contribution of this work is to reduce the task of proving the low rank of a matrix to proving that some field elements are roots of a q-polynomial. Such polynomials are MPC-friendly thanks to the linearity of the Frobenius endomorphism. Using this reduction, we can produce signatures relying on the *MinRank problem* and on the *rank syndrome decoding problem* with sizes below 6 KB.

References

AMGH+22. Carlos Aguilar-Melchor, Nicolas Gama, James Howe, Andreas Hülsing, David Joseph, and Dongze Yue. The return of the sdith. Cryptology ePrint Archive, Paper 2022/1645, 2022. `https://eprint.iacr.org/2022/1645`.

ARS+15. Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 430–454. Springer, Heidelberg, April 2015.

ARZV22. Gora Adj, Luis Rivera-Zamarripa, and Javier Verbel. Minrank in the head: Short signatures from zero-knowledge proofs. Cryptology ePrint Archive, Paper 2022/1501, 2022. `https://eprint.iacr.org/2022/1501`.

BdK+21. Carsten Baum, Cyprien de Saint Guilhem, Daniel Kales, Emmanuela Orsini, Peter Scholl, and Greg Zaverucha. Banquet: Short and fast signatures from AES. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 266–297. Springer, Heidelberg, May 2021.

BESV22. Emanuele Bellini, Andre Esser, Carlo Sanna, and Javier Verbel. Mr-dss – smaller minrank-based (ring-)signatures. In Jung Hee Cheon and Thomas Johansson, editors, *Post-Quantum Cryptography*, pages 144–169, Cham, 2022. Springer International Publishing.

Beu20. Ward Beullens. Sigma protocols for MQ, PKP and SIS, and Fishy signature schemes. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 183–211. Springer, Heidelberg, May 2020.

BFK+19. Ward Beullens, Jean-Charles Faugère, Eliane Koussa, Gilles Macario-Rat, Jacques Patarin, and Ludovic Perret. PKP-based signature scheme. In Feng Hao, Sushmita Ruj, and Sourav Sen Gupta, editors, *INDOCRYPT 2019*, volume 11898 of *LNCS*, pages 3–22. Springer, Heidelberg, December 2019.

BG22. Loïc Bidoux and Philippe Gaborit. Compact post-quantum signatures from proofs of knowledge leveraging structure for the pkp, sd and rsd problems, 2022.

BMSV22. Emanuele Bellini, Rusydi H. Makarim, Carlo Sanna, and Javier Verbel. An estimator for the hardness of the mq problem. In *Progress in Cryptology - AFRICACRYPT 2022*, pages 323–347, Berlin, Heidelberg, 2022. Springer-Verlag.

BN20. Carsten Baum and Ariel Nof. Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 495–526. Springer, Heidelberg, May 2020.

CHR$^+$16. Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. From 5-pass MQ-based identification to MQ-based signatures. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 135–165. Springer, Heidelberg, December 2016.

Cou01. Nicolas Courtois. Efficient zero-knowledge authentication based on a linear algebra problem MinRank. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 402–421. Springer, Heidelberg, December 2001.

DKR$^+$21. Christoph Dobraunig, Daniel Kales, Christian Rechberger, Markus Schofnegger, and Greg Zaverucha. Shorter signatures based on tailor-made minimalist symmetric-key crypto. Cryptology ePrint Archive, Report 2021/692, 2021. `https://eprint.iacr.org/2021/692`.

FJR21. Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. Shared permutation for syndrome decoding: New zero-knowledge protocol and code-based signature. Cryptology ePrint Archive, Report 2021/1576, 2021. `https://eprint.iacr.org/2021/1576`.

FJR22. Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 541–572. Springer, Heidelberg, August 2022.

FMRV22. Thibauld Feneuil, Jules Maire, Matthieu Rivain, and Damien Vergnaud. Zero-knowledge protocols for the subset sum problem from MPC-in-the-head with rejection. Cryptology ePrint Archive, Report 2022/223, 2022. `https://eprint.iacr.org/2022/223`.

FR22. Thibauld Feneuil and Matthieu Rivain. Threshold linear secret sharing to the rescue of mpc-in-the-head. Cryptology ePrint Archive, Report 2022/1407, 2022. `https://eprint.iacr.org/2022/1407`.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987.

IKOS07. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In David S. Johnson and Uriel Feige, editors, *39th ACM STOC*, pages 21–30. ACM Press, June 2007.

KKW18. Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 525–537. ACM Press, October 2018.

KZ20. Daniel Kales and Greg Zaverucha. An attack on some signature schemes constructed from five-pass identification schemes. In Stephan Krenn, Haya Shulman, and Serge Vaudenay, editors, *CANS 20*, volume 12579 of *LNCS*, pages 3–22. Springer, Heidelberg, December 2020.

KZ22. Daniel Kales and Greg Zaverucha. Efficient lifting for shorter zero-knowledge proofs and post-quantum signatures. Cryptology ePrint Archive, Report 2022/588, 2022. `https://eprint.iacr.org/2022/588`.

LN96. Rudolf Lidl and Harald Niederreiter. *Finite Fields.* Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 1996.

Sha90. Adi Shamir. An efficient identification scheme based on permuted kernels (extended abstract) (rump session). In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 606–609. Springer, Heidelberg, August 1990.

SINY22. Bagus Santoso, Yasuhiko Ikematsu, Shuhei Nakamura, and Takanori Yasuda. Three-pass identification scheme based on minrank problem with half cheating probability, 2022.

SSH11. Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. Public-key identification schemes based on multivariate quadratic polynomials. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 706–723. Springer, Heidelberg, August 2011.

Ste94. Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 13–21. Springer, Heidelberg, August 1994.

Vér96. Pascal Véron. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996.

Wan22. William Wang. Shorter signatures from MQ. Cryptology ePrint Archive, Report 2022/344, 2022. `https://eprint.iacr.org/2022/344`.

– Supplementary Material –

A Using Shamir's Secret Sharings in the Proof of Knowledge for Rank SD

Let us focus on the MPC protocol described in Figure 6. This protocol checks that a vector corresponds to a solution of a rank syndrome decoding problem, by using a q-poynomial. In what follows, we describe how the MPC protocol behaves when replacing additive sharings by Shamir's secret sharings over $\mathbb{F}_{q^m}$.

To share a secret value $v \in \mathbb{F}_{q^m}$, the $(\ell+1, N)$-Shamir's secret sharing scheme proceeds as follows:

- sample $r_1, \ldots, r_\ell$ uniformly in $\mathbb{F}_{q^m}$,
- build the polynomial P as $P(X) = v + \sum_{i=1}^{\ell} r_i X^i$,
- build the shares $[\![v]\!]_i$ as evaluations $P(e_i)$ of P for each $i \in \{1, \ldots, N\}$, where $e_1, \ldots, e_N$ are public non-zero distinct points of $\mathbb{F}_{q^m}$.

From a sharing $[\![v]\!]$ of v, the parties can easily build a sharing of v^q: they just need to compute

$$[\![v^q]\!]_i \leftarrow [\![v]\!]_i^q$$

for all i. However, the parties' evaluation points of $[\![v^q]\!]$ are not $e_1, \ldots, e_N$, but they are $e_1^q, \ldots, e_N^q$. Indeed, we have

$$\begin{aligned} P(X)^q &= \left(v + \sum_{i=1}^{\ell} r_i X^i\right)^q \\ &= v^q + \sum_{i=1}^{\ell} r_i^q X^{q \cdot i} \\ &= v^q + \sum_{i=1}^{\ell} r_i^q (X^q)^i \\ &= P'(X^q), \quad \text{where } P' := v^q + \sum_{i=1}^{\ell} r_i^q X^i. \end{aligned}$$

Thus for all i, we get

$$[\![v]\!]_i^q = P(e_i)^q = P'(e_i^q) = [\![v^q]\!]_i$$

if P' is the polynomial which encodes $[\![v^q]\!]$.

Adding two sharings is possible if and only if *those two sharings have the same parties' evaluation points.* The MPC protocol described in Figure 6 satisfies this property, enabling us to replace the additive sharings by Shamir's secret sharings over $\mathbb{F}_{q^m}$. If we denote $e_1, \ldots, e_N$ the parties' evaluation points of $[\![x_A]\!]$, then

- for all $i \in \{0, \ldots, r-1\}$, the parties' evaluation points for $[\![w_i]\!]$, $[\![a_i]\!]$ and $[\![\alpha_i]\!]$ are $e_1^{q^i}, \ldots, e_N^{q^i}$,
- the parties' evaluation points for $[\![\beta]\!]$, $[\![z]\!]$ and $[\![c]\!]$ are $e_1^{q^r}, \ldots, e_N^{q^r}$.

B Proof of Knowledge for Sum-Rank SD

We want to build a zero-knowledge proof of knowledge for the *sum-rank syndrome decoding problem*:

Definition 5 (Sum-Rank Syndrome Decoding Problem). *Let $\mathbb{F}_{q^m}$ be the finite field with q^m elements. Let (n, k, ℓ, r) be positive integers such that $k \leq n$ and $\ell \mid n$. We define the sum-rank weight $\mathrm{wt}_{SR}(x)$ of an element of $\mathbb{F}_{q^m}^n$ as*

$$\mathrm{wt}_{SR}(x) := \sum_{i=1}^{n/\ell} \mathrm{wt}_R(x_i),$$

with $x := (x_1, \dots, x_{\frac{n}{\ell}})$. The sum-rank syndrome decoding problem with parameters (q, m, n, k, ℓ, r) is the following problem:

> *Let H, x and y be such that:*
>
> 1. *H is uniformly sampled from $\{(H'|I_{n-k}), H' \in \mathbb{F}_{q^m}^{(n-k)\times n}\}$,*
> 2. *x is uniformly sampled from $\{x \in \mathbb{F}_{q^m}^n : \mathrm{wt}_{SR}(x) \leq r\}$,*
> 3. *y is built as $y := Hx$.*
>
> *From (H, y), find x.*

The prover wants to convince the verifier that she knows such an x, *i.e.* a vector $x \in \mathbb{F}_{q^m}^n$ such that $y = Hx$ and $\mathrm{wt}_{SR}(x) \leq r$. To proceed, the prover will first share the secret vector x and then use an MPC protocol which verifies that this vector satisfies the above property.

MPC Protocol. As in Section 5.3, H is in standard form and we split the secret

$$x := \begin{pmatrix} x_A \\ y - H'x_A \end{pmatrix}.$$

We want to build an MPC protocol which takes as input (a sharing of) x_A and which outputs

$$\begin{cases} \textsc{Accept} \text{ if } \sum_{i=1}^{n/\ell} \mathrm{wt}_R(x_i) \leq r \text{ where } \begin{pmatrix} x_1 \\ \vdots \\ x_{\frac{n}{\ell}} \end{pmatrix} := \begin{pmatrix} x_A \\ y - H'x_A \end{pmatrix} \\ \textsc{Reject} \text{ otherwise.} \end{cases}$$

For each chunk $x_i \in \mathbb{F}_{q^m}^\ell$ with $i \in [\frac{n}{\ell}]$, let us define the binary vector $d_i \in \{0,1\}^\ell$ as

$$\forall j \in [\ell],\ (d_i)_j := \begin{cases} 0 & \text{if } (x_i)_j \in \mathrm{Vect}_{\mathbb{F}_q}((x_i)_1, \dots, (x_i)_{j-1}) \\ 1 & \text{otherwise} \end{cases}$$

and let us remark that there exists a lower triangular matrix $T_i \in \mathbb{F}_q^{\ell\times\ell}$ with the form $\begin{pmatrix} 1 & & & (0) \\ * & 1 & & \\ * & * & \ddots & \\ * & * & * & 1 \end{pmatrix}$ such that

$$d_i \circ x_i = T_i x_i$$

where $\circ$ is the component-wise multiplication. The matrix T_i corresponds to the process of removing dependencies in x_i. We have

$$\mathrm{wt}_H(d_i) \geq \mathrm{wt}_H(d_i \circ x_i) = \mathrm{wt}_R(d_i \circ x_i)$$

since each non-zero coordinates of $d_i \circ x_i$ are independent by definition of d_i. Moreover, we have

$$\mathrm{wt}_R(d_i \circ x_i) = \mathrm{wt}_R(T_i x_i) = \mathrm{wt}_R(x_i)$$

since T_i is invertible. By defining $d := (d_1, \dots, d_{\frac{n}{\ell}})$, the MPC protocol will check that $\mathrm{wt}_H(d) \leq r$, and since

$$\sum_{i=1}^{\frac{n}{\ell}} \mathrm{wt}_R(x_i) = \sum_{i=1}^{\frac{n}{\ell}} \mathrm{wt}_R(d_i \circ x_i) \leq \sum_{i=1}^{\frac{n}{\ell}} \mathrm{wt}_H(d_i) = \mathrm{wt}_H(d)$$

the desired inequality would be checked. In order to check $\mathrm{wt}_H(d) \leq w$, we will use the protocol of [FJR22].

To sum up, to check the weight inequality, the MPC protocol takes as input the vectors $d_1 \dots, d_{\frac{n}{\ell}}$ and the matrices $T_1, \dots, T_{\frac{n}{\ell}}$ (in addition to x_A) and proceeds as follows:

1. The parties locally build $[\![x]\!]$ as
$$\begin{pmatrix} [\![x_A]\!] \\ y - H'[\![x_A]\!] \end{pmatrix}.$$
2. The parties execute the [FJR22]'s protocol to check that $\mathrm{wt}_H(d) \leq r$.
3. For $i \in \{1, \dots, \frac{n}{\ell}\}$, the parties check that $d_i \circ x_i = T_i x_i$ as follows:
   - The parties locally set $[\![D_i]\!] \in \mathbb{F}_q^{\ell \times \ell}$ as a diagonal matrix for which the diagonal is the vector $[\![d_i]\!]$.
   - The parties executes the protocol Π_{MM}^{η} to check that $(D_i - T_i)x_i = 0$.

The MPC protocol is completely described in Figure 9.

Proof of Knowledge. Using the MPC-in-the-Head paradigm (see Section 2.1), we transform the above MPC protocol into an interactive zero-knowledge proof of knowledge which enables to convince a verifier that a prover knows the solution of a sum-rank syndrome decoding problem. The soundness error of the resulting protocol is

$$\varepsilon := \frac{1}{N} + \left(1 - \frac{1}{N}\right) \max\left(\frac{1}{q^{\eta}}, \delta_{\eta_1,\eta_2}\right)$$

where δ_{η_1,η_2} is the false positive rate of [FJR22]. By repeating the protocol τ times, we get a soundness error of ε^{τ}. To obtain a soundness error of λ bits, we can take $\tau = \left\lceil \frac{-\lambda}{\log_2 \varepsilon} \right\rceil$. We can transform the interactive protocol into a non-interactive proof / signature thanks to the Fiat-Shamir transform [FS87]. According to [KZ20], the security of the resulting scheme is

$$\mathrm{cost}_{\mathrm{forge}} := \min_{\tau_1,\tau_2:\tau_1+\tau_2=\tau} \left\{ \frac{1}{\sum_{i=\tau_1}^{\tau} \binom{\tau}{i} p^i (1-p)^{\tau-i}} + N^{\tau_2} \right\}$$

where $p := \max\left(\frac{1}{q^{\eta}}, \delta_{\eta_1,\eta_2}\right)$.

The communication cost of the scheme (in bits) is

$$4\lambda + \tau \cdot ((\underbrace{k \cdot m}_{x_A} + \underbrace{\frac{n}{\ell}\frac{(\ell-1)\ell}{2}}_{T_1,\dots} + \underbrace{\frac{n}{\ell}(\ell-1)}_{d_1,\dots} + \underbrace{\frac{n}{\ell}(\eta \cdot \ell) + \eta \cdot \min\{m, \ell-1\}}_{c,\alpha_1,\dots} + \underbrace{2r\eta_1 + 3\eta_1\eta_2}_{\mathrm{SDitH}}) \cdot \log_2 q + \underbrace{\lambda \cdot \log_2 N + 2\lambda}_{\mathrm{MPCitH}})$$

where λ is the security level, r is a scheme parameter and τ is computed such that the soundness error is of λ bits in the interactive case and such that $\mathrm{cost}_{\mathrm{forge}}$ is of λ bits in the non-interactive case.

Public values: $H = (H'|I_{n-k}) \in \mathbb{F}_{q^m}^{(n-k)\times n}$ and $y \in \mathbb{F}_{q^m}^{n-k}$.

Inputs: Each party takes a share of the following sharings as inputs:

- $\llbracket x_A \rrbracket$ where $x \in \mathbb{F}_{q^m}^k$,
- $\llbracket d \rrbracket, \llbracket T_1 \rrbracket, \dots, \llbracket T_{\frac{n}{\ell}} \rrbracket$ where $d \in \{0,1\}^n$ and $T_1 \dots, T_{\frac{n}{\ell}} \in \mathbb{F}_q^{\ell\times\ell}$ such that

$$\forall i \in \{1, \dots, \frac{n}{\ell}\}, d_i \circ x_i = T_i x_i$$

- $\llbracket a_1 \rrbracket, \dots, \llbracket a_{\frac{n}{\ell}} \rrbracket$ where $a_1, \dots, a_{\frac{n}{\ell}} \in \mathbb{F}_q^{\eta\times\ell}$.
- $\llbracket c \rrbracket$ where $c \in \mathbb{F}_{q^m}^{\eta}$ such that $c = \sum_{i=1}^{n} a_i x_i$
- $\llbracket Q \rrbracket$ where $Q = \prod_{d_i \neq 0}(X - \gamma_i) \in \mathbb{F}_{q^{\eta_1}}[X]$
- $\llbracket P \rrbracket$ where $P \in \mathbb{F}_{q^{\eta_1}}[X]$ satisfies $SQ = FP$ with $F(X) := \prod_{i\in[n]}(X - \gamma_i)$ and S the unique polynomial of degree $n-1$ such that $S(\gamma_i) = d_i$ for all $i \in [n]$.
- $\llbracket a' \rrbracket, \llbracket b' \rrbracket, \llbracket c' \rrbracket$ where $a', b', c' \in \mathbb{F}_{q^{\eta_1\eta_2}}$ such that $c' = a' \cdot b'$.

$\llbracket T \rrbracket$ where $T \in \mathbb{F}_q^{n\times w}$ and $\llbracket R \rrbracket$ where $R \in \mathbb{F}_q^{w\times wm}$, such that $X = TR$ where X is the matrix form of x.

MPC Protocol:

1. The parties get random $\Sigma_1, \dots, \Sigma_{\frac{n}{\ell}} \in \mathbb{F}_q^{\eta\times\ell}$.
2. The parties get random $r, \varepsilon' \in \mathbb{F}_{q^{\eta_1\eta_2}}$.

3. The parties locally compute $\llbracket S \rrbracket$ by interpolation such that $\forall j, \llbracket S(\gamma_i) \rrbracket = \llbracket d_j \rrbracket \in \mathbb{F}_q$.
4. The parties locally compute $\llbracket S(r) \rrbracket$, $\llbracket Q(r) \rrbracket$ and $\llbracket P(r) \rrbracket$.
5. The parties locally set $\llbracket \alpha' \rrbracket = \varepsilon' \cdot \llbracket Q(r) \rrbracket + \llbracket a' \rrbracket$ and $\llbracket \beta' \rrbracket = \llbracket S(r) \rrbracket + \llbracket b' \rrbracket$.
6. The parties open α' and β'.
7. The parties locally set $\llbracket v' \rrbracket = \varepsilon' \cdot \llbracket (F \cdot P)(r) \rrbracket - \llbracket c' \rrbracket + \alpha' \cdot \llbracket b' \rrbracket + \beta' \cdot \llbracket a' \rrbracket - \alpha' \cdot \beta'$.

8. The parties locally set $\llbracket x_B \rrbracket = y - H' \llbracket x_A \rrbracket$.
9. The parties locally set $\llbracket x \rrbracket = (\llbracket x_A \rrbracket, \llbracket x_B \rrbracket)$.
10. For $i \in \{1, \dots, \frac{n}{\ell}\}$,
    - The parties locally write $\llbracket d_i \rrbracket$ as a diagonal matrix $\llbracket D_i \rrbracket \in \mathbb{F}_q^{\ell\times\ell}$.
    - The parties locally set $\llbracket \alpha_i \rrbracket = \Sigma_i(\llbracket D_i \rrbracket - \llbracket T_i \rrbracket) + \llbracket a_i \rrbracket$.
    - The parties open $\alpha_i \in \mathbb{F}_q^{\eta\times\ell}$.
11. The parties locally set $\llbracket v \rrbracket = \sum_{i=1}^{\frac{n}{\ell}} \alpha_i \llbracket x_i \rrbracket - \llbracket c \rrbracket$.

12. The parties outputs Accept if $v = 0$ and $v' = 0$, and Reject otherwise.

Fig. 9: An MPC Protocol that verifies that the given input corresponds to a solution of a sum-rank syndrome decoding problem. $\gamma_1, \dots, \gamma_n$ are distinct points of $\mathbb{F}_{q^{\eta_1}}$