MinRank in the Head: Short Signatures from Zero-Knowledge Proofs

Gora Adj, Luis Rivera-Zamarripa, Javier Verbel

Technology Innovation Institute, UAE
{gora.adj,luis.zamarripa,javier.verbel}@tii.ae

Abstract. In recent years, many digital signature scheme proposals have been built from the so-called MPC-in-the-head paradigm. This has shown to be an outstanding way to design efficient signatures with security based on hard problems.
MinRank is an NP-complete problem extensively studied due to its applications to cryptanalysis since its introduction in 1999. However, only a few schemes base their security on its intractability, and their signature size is large compared with other proposals based on NP problems. This paper introduces the first MinRank-based digital signature scheme that uses the MPC-in-the-head, enabling it to achieve small signature sizes and running times. For NIST's category I parameter set, we obtain signatures of 6.5KB, which is competitive with the shortest proposals in the literature that are based on non-structured problems.

Keywords: MinRank · zero-knowledge · proof of knowledge · MPC-in-the-Head

1 Introduction

Signature schemes form an essential part of almost every secure digital communication protocol, allowing the receiver of a message to verify authenticity (the identity of the sender) and integrity (no unauthorized modifications of the message).

One way to design signature schemes is via *zero-knowledge proofs of knowledge*, which consists of executing an interactive *identification protocol* between a prover and a verifier. In this, the prover tries to convince the verifier that he knows a witness of a public statement without revealing any information beyond the fact that the statement is true. An essential property of zero-knowledge proofs is the *soundness error*, defined as the probability that an adversary successfully convinces the verifier about the truth of the public statement without knowing a witness.

In 2007, Ishai, Kushilevitz, Ostrovsky, and Sahai introduced the MPC-in-the-head paradigm to build zero-knowledge proofs of knowledge from *multi-party computation* (MPC) protocols [23]. In an MPC protocol, a set of N parties jointly compute the image of a function on their local and private inputs without revealing any information beyond the computed image. In principle, one can use

the MPC-in-the-head paradigm to prove, in zero-knowledge, the knowledge of a solution to any problem that is verifiable using a logical circuit. However, due to the potentially large size of the resulting verification circuit, such a generic approach is not always the optimal one, and more efficient options might be found by exploiting the specific structure of the problem.

The MinRank problem is an NP-complete problem introduced by Buss, Frandsen, and Shallit in [8]. It is defined by an integer r and a set of matrices $M_0, M_1, \ldots, M_k$ over a finite field $\mathbb{F}_q$. In its decisional version, the goal is to decide whether or not there exists a combination $M_0 + \sum_{i=1}^{k} \alpha_i M_i$ of the matrices having rank at most r, where the $\alpha_i \in \mathbb{F}_q$.

The MinRank problem first appeared in cryptology in the cryptanalysis of the HFE cryptosystem [25]. Here, Kipnis and Shamir proposed the so-called *Kipnis-Shamir modeling*, where an instance of the MinRank problem is modeled as a system of bilinear equations. Given $M_0, \ldots, M_k \in \mathbb{F}_q^{n \times n}$, such a bilinear system is defined by the entries of the following matrix equation

$$\left(M_0 + \sum_{i=1}^{k} x_i M_i\right) \cdot \begin{bmatrix} I_{n-r} \\ K \end{bmatrix} = O,$$

where the sets of variables are the x_i and the entries of $K \in \mathbb{F}_q^{r \times (n-r)}$, and I_{n-r} is the identity matrix of size $n - r$.

The hardness of the MinRank seems to be a reasonable assumption to be made while designing secure *post-quantum schemes*. First, due to its relevance in cryptanalysis [6,7,20,30], several classical algorithms solving this problem have been extensively studied [1,2,3,11,15,16,25,31]. Second, random instances of the MinRank problem are expected to be hard [1, Sec. 5.6]. Finally, it is known no quantum algorithm that improves over classical algorithms while solving the MinRank problem.

Despite the hardness of MinRank seems to be a reasonable assumption to build secure cryptographic schemes, only three cryptographic schemes based on this problem have been proposed in the literature, and all of them are signature schemes built from zero-knowledge proofs: Courtois' scheme [11], MR-DSS [10], and the scheme by Santoso, Ikematsu, Nakamura and Yasuda [29].

Our Contributions In this paper, we propose a provable secure signature scheme based on the hardness of the MinRank problem. To this end, we introduce the first MPC protocol specifically designed to prove knowledge of a solution to an instance of the MinRank problem. Our MPC protocol builds over the Kipnis-Shamir modeling to generate shares of $(\alpha_1, \ldots, \alpha_k)$ and a matrix K satisfying Section 1. For this to work, we extend the simultaneous verification of two scalar multiplication triples from [4] (see also [27]) to matrix multiplication triples. Then, following the MPC-in-the-head paradigm, we obtain an honest-verifier zero-knowledge identification scheme for the MinRank problem. We prove that the protocol thus constructed is sound with soundness error of $\frac{1}{N} + \frac{N-1}{q^n N}$, where

N is the number of parties in the MPC protocol, n the number of columns of the M_i, and q the size of the field.

Finally, using the generalized Fiat-Shamir transformation, we turn our identification scheme into a digital signature scheme. Assuming the hardness of MinRank, we prove that our scheme is *existentially unforgeable under chosen messages attack* (EU-CMA). We propose several parameter sets for the new scheme. Each of them targets security above either 143, 207, or 273 bits.

In terms of signature size, our scheme is smaller than MR-DSS by a factor greater than 3.75 for the parameters suggested in [5]. In terms of public key size, our scheme achieves the same sizes as MR-DSS.

Related works MinRank-based signature schemes built on zero-knowledge proofs have been previously proposed in the literature. It first started with the scheme introduced by Courtois in 2001 [11], where the underlying zero-knowledge proof has a soundness error of 2/3. More than 20 years later, Bellini, Esser, Sanna, and Verbel introduced MR-DSS [5], improving the soundness error of Courtois' protocol to 1/2. Santoso, Ikematsu, Nakamura, and Yasuda proposed another MinRank-based signature scheme [29]. They also obtained proof with a soundness error of 1/2 but with larger signatures than Courtois and MR-DSS for comparable parameter sets (see [5, Appendix C]).

MPC protocols to verify multiplication triples of shares of elements in a ring have been proposed before. Baum and Nof introduced in [4] an MPC protocol that simultaneously verifies two multiplication triples of shares of elements in a finite field. A generalization of this approach was introduced in [28], where the authors showed how more than two multiplication triples of shares could be verified at the same time by lifting them over an extension field. In [17], Feneuil, Joux, and Rivain showed an MPC protocol to verify multiplication triples of shares of univariate polynomials. They used this protocol to design a signature scheme based on the syndrome decoding problem. Our work in this paper has been mostly inspired by the ideas in [17].

Organization of the paper. The remainder of the paper is organized as follows. In Section 2, we provide the relevant background on zero-knowledge proofs of knowledge, the MPC-in-the-head paradigm, the MinRank problem, and digital signatures. In Section 4, we adapt the verification of multiplication triples from [4] to the case of matrix multiplications. Our zero-knowledge protocol for the MinRank problem is presented in Section 5, and the subsequent signature scheme in Section 6. In Section 7, we compare our signature scheme with some relevant ones in the literature. We draw our conclusions in Section 8.

2 Preliminaries

This section presents the preliminary concepts and notations that are used throughout the paper.

Notation. All over this paper, we use λ to denote the security parameter, $\mathbb{F}_q$ denotes a finite field of q elements. For any positive integer n, we denote $[n] = \{1, 2, \ldots, n\}$. For positive integers m, n, the notation $\mathbb{F}_q^{m \times n}$ refers to the set of matrices over $\mathbb{F}_q$ of m rows and n columns. For $k > 0$, we use $\boldsymbol{M}$ to denote a tuple of $k+1$ matrices $(M_0, M_1, \ldots, M_k) \in (\mathbb{F}_q^{m \times n})^{k+1}$. For a given vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)^T \in \mathbb{F}_q^k$, we define $\boldsymbol{M}_{\boldsymbol{\alpha}} = \sum_{i=1}^{k} \alpha_i M_i$.

The notation $a \leftarrow \mathcal{A}(x)$ indicates that a is the output of an algorithm $\mathcal{A}$ on input x. Moreover, $a \xleftarrow{\$} \mathcal{S}$ indicates that a is sampled uniformly at random from a set $\mathcal{S}$.

Definition 1 (Collision-Resistant Hash Functions). *We say that a function $h : \{0,1\}^* \to \{0,1\}^{p(\lambda)}$, with $p(\cdot)$ a polynomial, is a collision-resistant hash function if it can be computed in polynomial time and for any probabilistic polynomial algorithm $\mathcal{A}$, there exists a negligible function ε such that*

$$\Pr[(x_1, x_2) \leftarrow \mathcal{A}(1^\lambda, h) \mid x_1 \neq x_2, h(x_1) = h(x_2)] < \varepsilon(\lambda).$$

We consider in this paper hash functions $\mathrm{Hash}_0, \mathrm{Hash}_1$ and $\mathrm{Hash}_2 : \{0,1\}^* \to \{0,1\}^{2\lambda}$ that we assume to be collision resistant.

Definition 2 (Indistinguishability). *Two distributions $\{D_\lambda\}_\lambda$ and $\{E_\lambda\}_\lambda$ are said to be $(t(\lambda), \varepsilon(\lambda))$-indistinguishable for functions t and ε if for any probabilistic algorithm $\mathcal{A}$ running in time at most $t(\lambda)$, we have*

$$\left| \Pr\left[1 \leftarrow \mathcal{A}(x) \mid x \leftarrow D_\lambda\right] - \Pr\left[1 \leftarrow \mathcal{A}(x) \mid x \leftarrow E_\lambda\right] \right| \leq \varepsilon(\lambda).$$

When ε is negligible for any t polynomial in λ, we just say that the two distributions are indistinguishable.

Definition 3 (Pseudorandom Generator (PRG)). *Let $G : \{0,1\}^* \to \{0,1\}^*$ be a function such that for any $s \in \{0,1\}^\lambda$ we have $G(s) \in \{0,1\}^{p(\lambda)}$, where $p(\cdot)$ is a polynomial. We say that G is a (t, ε)-secure pseudorandom generator if the following two conditions hold:*

- *Expansion: $p(\lambda) > \lambda$;*
- *Pseudorandomness: the distributions*

$$\{G(s) \mid s \xleftarrow{\$} \{0,1\}^\lambda\} \text{ and } \{r \mid r \xleftarrow{\$} \{0,1\}^{p(\lambda)}\}$$

are (t, ε)-indistinguishable.

2.1 Commitment Schemes

In our *identification scheme*, which is presented in Section 5, we make use of a commitment scheme $\mathsf{Com} : \{0,1\}^* \times \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ that is assumed *computationally hiding* and *computationally binding*. We formally define these concepts.

Definition 4 (Computational hiding). *We say that a commitment scheme* Com *is* (t, ε)*-hiding if for every pair of messages* (m, m')*, the two following distributions are* (t, ε)*-indistinguishable:*

$$\{c \mid c \leftarrow \mathsf{Com}(m, \rho), \rho \xleftarrow{\$} \{0,1\}^\lambda)\} \text{ and } \{c \mid c \leftarrow \mathsf{Com}(m', \rho), \rho \xleftarrow{\$} \{0,1\}^\lambda)\}.$$

We say that Com *is computationally hiding if the two distributions are indistinguishable.*

Definition 5 (Computational binding). *We say that* Com *is computationally binding if for all algorithms* $\mathcal{A}$ *running in time polynomial in* λ*, the probability that* $\mathcal{A}$ *outputs two different messages committing to the same value is negligible, i.e., for a negligible* $\varepsilon(\lambda)$*, we have*

$$\Pr\left[\mathsf{Com}(m, \rho) = \mathsf{Com}(m', \rho') \mid (m, \rho, m', \rho') \leftarrow \mathcal{A}(1^\lambda)\right] \leq \varepsilon(\lambda).$$

2.2 Digital Signatures Schemes

Since our main goal in this paper is to build a digital signature scheme, we define the concept in the following.

Definition 6 (Signature scheme). *A digital signature scheme is a tuple of three probabilistic polynomial-time algorithms* $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verf})$ *that fulfill the following properties:*

1. *The key-generation algorithm* KeyGen *takes as input a security parameter* 1^λ *and outputs a pair of public/private keys* $(\mathsf{pk}, \mathsf{sk})$*.*
2. *The signing algorithm takes a message* $\mathsf{msg} \in \{0,1\}^*$ *and a private key* sk*, and it outputs a signature* σ*.*
3. *The verification algorithm* Verf *is deterministic. It takes as input a message* $\mathsf{msg} \in \{0,1\}^*$*, a signature* σ*, and a public key* pk*. It outputs 1 to mean that it* ***accepts*** σ *as a valid signature for* msg*, otherwise it* ***rejects*** *outputting 0.*

Correctness We require a signature scheme to be correct. That is, for every security parameter λ, every $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and every message $\mathsf{msg} \in \{0,1\}^*$ it holds that

$$1 \leftarrow \mathsf{Verf}\big(\mathsf{pk}, \mathsf{msg}, \mathsf{Sign}(\mathsf{sk}, \mathsf{msg})\big)$$

Existential Unforgeability under Adaptive Chosen Message Attacks We define digital signature schemes to be secure if it is existential unforgeability under adaptive chosen-message attacks (EU-CMA)[21]. This means that for all probabilistic polynomial-time adversaries $\mathcal{A}$, the probability

$$\Pr\left[1 \leftarrow \mathsf{Verf}(\mathsf{pk}, \mathsf{msg}^*, \sigma^*) \;\middle|\; \begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{pk})}\end{matrix}\right]$$

is a negligible function in λ, where $\mathcal{A}$ is given access to a signing oracle $\mathcal{O}_{\mathsf{Sign}(\mathsf{sk}, \cdot)}$, and msg^* has not been queried to $\mathcal{O}_{\mathsf{Sign}(\mathsf{sk}, \cdot)}$.

2.3 5-Pass identification schemes

In general, an identification scheme (IDS) is a two-party interactive protocol executed by a *prover* P and a *verifier* V. Here P wants to prove to V its knowledge of a private value sk such that $(\mathsf{pk}, \mathsf{sk})$ satisfies a given relation, where pk is a given public value. We say that an IDS scheme is a *canonical 5-pass* if it follows the structure described in Fig. 1. Formally we have the following.

Definition 7 (5-pass identification scheme). *A 5-pass identification scheme is a tuple of three probabilistic polynomial-time algorithms* (KeyGen, P, V) *such that*

1. *The key-generation algorithm* KeyGen *takes as input a security parameter* 1^λ *and outputs a pair of key pair* $(\mathsf{pk}, \mathsf{sk})$.
2. P *and* V *are interactive algorithms that follow a protocol as the one described in Fig. 1. The prover* P *has as input a key pair* $(\mathsf{pk}, \mathsf{sk})$*, and the verifier* V *takes as input the public key* pk*. At the end of the protocol,* V *outputs 1, indicating that it* ***accepts*** *the proof that* P *knows* sk*, otherwise* V ***rejects*** *and outputs 0.*

A *transcript* of a 5-pass IDS is a tuple $(\mathsf{com}, \mathsf{ch}_1, \mathsf{rsp}_1, \mathsf{ch}_2, \mathsf{rsp}_2)$ (as in Fig. 1) referring to all the messages exchanged between P and V in one execution of the the IDS. Here, we require an IDS to fulfill the following security properties.

Correctness An IDS is said to be correct if for any $\lambda \in \mathbb{N}$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ it holds

$$\Pr\left[1 \leftarrow V(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{rsp}_1, \mathsf{ch}_2, \mathsf{rsp}_2)\right] = 1,$$

where $(\mathsf{com}, \mathsf{ch}_1, \mathsf{rsp}_1, \mathsf{ch}_2, \mathsf{rsp}_2)$ is the transcript of an honest execution of the protocol between the prover P with input $(\mathsf{pk}, \mathsf{sk})$ and the verifier V on input pk.

Soundness (with soundness error ε) We say that an IDS is sound with soundness error ε if for every polynomial-time adversary $\mathcal{A}$ the difference

$$\Pr\left[\begin{matrix}(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ 1 \leftarrow V(\mathsf{pk}, \mathsf{com}_{\mathcal{A}}, \mathsf{ch}_1, \mathsf{rsp}_{1,\mathcal{A}}, \mathsf{ch}_2, \mathsf{rsp}_{2,\mathcal{A}})\end{matrix}\right] - \varepsilon$$

is a negligible function in λ, where $(\mathsf{com}_{\mathcal{A}}, \mathsf{ch}_1, \mathsf{rsp}_{1,\mathcal{A}}, \mathsf{ch}_2, \mathsf{rsp}_{2,\mathcal{A}})$ is the transcript of one execution of the protocol between $\mathcal{A}$ and V both with input pk.

Honest-verifier zero-knowledge We say that an IDS is honest-verifier zero-knowledge if there exists a probabilistic polynomial-time algorithm $\mathcal{S}$, called the simulator, such that on input pk, it outputs a transcript $(\mathsf{com}, \mathsf{ch}_1, \mathsf{rsp}_1, \mathsf{ch}_2, \mathsf{rsp}_2)$ from a distribution that is computationally indistinguishable from the distribution of transcripts of an honest execution of the protocol between a prover $\mathsf{P}(\mathsf{pk}, \mathsf{sk})$ and an honest verifier $\mathsf{V}(\mathsf{pk})$.

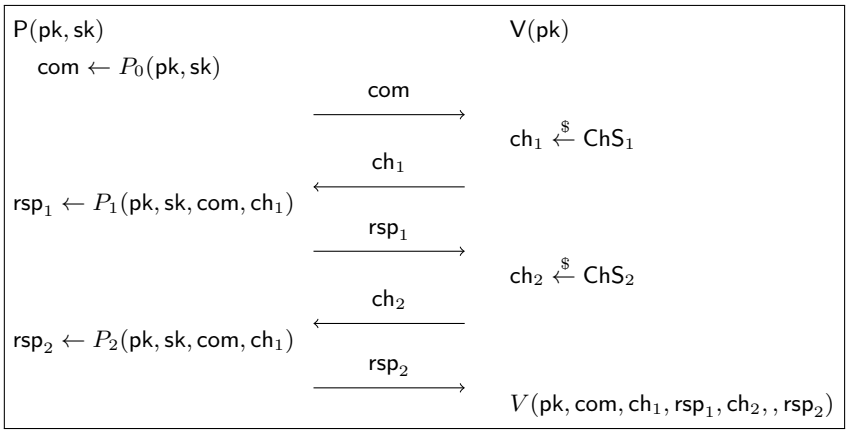

Fig. 1. Canonical 5-pass IDS.

2.4 The MinRank problem

In this section, we introduce the MinRank problem, which is the underlying hard problem of the signature scheme proposed in this paper.

Problem 1 (The MinRank problem). Let $\mathbb{F}_q$ be a finite field with q elements, and let m, n, k and r be positive integers. The MinRank problem with parameters $(q, m \times n, k, r)$ is defined as:

Input: A $(k+1)$-tuple $\boldsymbol{M} = (M_0, M_1, \ldots, M_k) \in (\mathbb{F}_q^{m \times n})^{k+1}$.

Output: $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)^T \in \mathbb{F}_q^k$ such that $\mathsf{Rank}\big(M_0 + \sum_{i=1}^{k} \alpha_i M_i\big) \leq r$.

The Kipnis-Shamir modeling The Kipnis-Shamir modeling was introduced in [26] and is based on the following fact: if there is a vector $\boldsymbol{\alpha} \in \mathbb{F}_q^k$ and a matrix $K \in \mathbb{F}_q^{r \times (n-r)}$ such that

$$\big(M_0 + \sum_{i=1}^{k} \alpha_i M_i\big) \cdot \begin{bmatrix} I \\ K \end{bmatrix} = O, \tag{1}$$

where $O \in \mathbb{F}_q^{n \times (n-r)}$ denotes the zero matrix, and $I \in \mathbb{F}_q^{(n-r) \times (n-r)}$ is a non-singular matrix, then the vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)^T$ is a solution to the MinRank problem with matrices $(M_0, M_1, \ldots, M_k)$.

We define the following to notation.

Definition 8 (Left- and Right-Side of a Matrix). *Let $A \in \mathbb{F}_q^{n \times n}$, and $1 \leq r < n$ be an integer. We define the left-side $A^{L_r} \in \mathbb{F}_q^{n \times (n-r)}$ and the right-side $A^{R_r} \in \mathbb{F}_q^{n \times r}$ of A, with respect to r, as the matrices satisfying*

$$A = [A^{L_r} | A^{R_r}].$$

In the rest of the paper, r always refers to the target rank of our MinRank instances. Hence, we only write A^L and A^R for simplicity.

Now, in the Kipnis-Shamir modeling, notice that if we fix I to be the identity matrix of size $n - r$ in Eq. (1), then the following holds

$$M_0^L + \sum_{i=1}^{k} \alpha_i M_i^L = -\left(M_0^R + \sum_{i=1}^{k} \alpha_i M_i^R \right) \cdot K. \tag{2}$$

To solve an instance of the MinRank problem, Kipnis and Shamir proposed to solve the bilinear system where equations are given by the entries of the left-hand side matrix of Eq. (1). The sets of variables in such a system are the $\alpha_1, \ldots, \alpha_k$ and the entries of K.

2.5 Multi-party computation

An MPC protocol allows N parties to jointly compute $z = f(x_1, \ldots, x_N)$ for a given function f, where each secret value x_i is known only by the i-th party. We say that an MPC protocol is correct and secure if, at the end of the protocol, every party i knows z but nothing other than the information it can deduce from z and x_i.

A core concept in MPC protocols is the one of *sharing*. We say that a tuple $[\![\mathfrak{S}]\!] := ([\![\mathfrak{S}]\!]_1, [\![\mathfrak{S}]\!]_2, \ldots, [\![\mathfrak{S}]\!]_N)$ is a sharing of a value $\mathfrak{S}$ with threshold $t < N$ if $\mathfrak{S}$ can be efficiently recovered from any t'-sized subset of elements in $[\![\mathfrak{S}]\!]$ with $t' > t$. A coordinate $[\![\mathfrak{S}]\!]_i$ is called a *share* of $\mathfrak{S}$.

Definition 9 (Additive sharing). *Let $\mathfrak{S}$ be an element in an additive group $\mathfrak{G}$. We say that a tuple $[\![\mathfrak{S}]\!] := ([\![\mathfrak{S}]\!]_1, [\![\mathfrak{S}]\!]_2, \ldots, [\![\mathfrak{S}]\!]_N)$ is an additive sharing of $\mathfrak{S}$ if $\mathfrak{S} = \sum_{i=1}^{N} [\![\mathfrak{S}]\!]_i$.*

This paper only deals with additive sharings, which have threshold $t = N-1$. Hence, from here on, we write sharing to mean an additive sharing of an element in a group. We denote by N the number of parties in an MPC protocol. Also, we assume that the i-th party holds the i-th coordinate of a given sharing $[\![\mathfrak{S}]\!]$. Finally, we say that every party executes an action A on $[\![\mathfrak{S}]\!]$ to mean that, for $i = 1, \ldots, N$, the i-th party executes A on its own share $[\![\mathfrak{S}]\!]_i$.

2.6 Scalar-Multiplication Triple Verification

Baum and Nof describe in [4] a secure MPC protocol to verify two scalar-multiplication triples $([\![x]\!], [\![y]\!], [\![z]\!])$ and $([\![a]\!], [\![b]\!], [\![c]\!])$ over a finite field $\mathbb{F}_q$.

Definition 10 (Scalar-Multiplication triple). *Let x, y, and z be elements of a finite field $\mathbb{F}_q$. We say that the triple of sharings $([\![x]\!], [\![y]\!], [\![z]\!])$ over $\mathbb{F}_q$ is a correct scalar-multiplication triple if $z = x \cdot y$. Otherwise we say that $([\![x]\!], [\![y]\!], [\![z]\!])$ is incorrect.*

At the beginning of the protocol, every party i starts with a 6-tuple of shares $(\llbracket x \rrbracket_i, \llbracket y \rrbracket_i, \llbracket z \rrbracket_i, \llbracket a \rrbracket_i, \llbracket b \rrbracket_i, \llbracket c \rrbracket_i)$. At the end, the parties output **accept** if they are convinced about the correctness of both triples $(\llbracket x \rrbracket, \llbracket y \rrbracket, \llbracket z \rrbracket)$ and $(\llbracket a \rrbracket, \llbracket b \rrbracket, \llbracket c \rrbracket)$. Otherwise, they output **reject**.

The authors of [4] proved that if both triples are indeed multiplicative, then the parties always output **accept**. Additionally, they established the following.

Proposition 1. *([4, Lemma 2]) If $(\llbracket x \rrbracket, \llbracket y \rrbracket, \llbracket z \rrbracket)$ or $(\llbracket a \rrbracket, \llbracket b \rrbracket, \llbracket c \rrbracket)$ is an incorrect scalar-multiplication triple, then the parties output **accept** in the aforementioned protocol with probability $\frac{1}{q}$.*

3 Exceptional Sets of Matrices over a Finite Fields

In the 5-pass zero-knowledge protocol presented in this work (Section 5, Fig. 2), the first challenge space consists of an *exceptional set* of square matrices. This concept appears with different names in the literature, but we stick with the one in [13,14].

Definition 11 (Exceptional sets). *Let R be a ring with $1 \neq 0$, and $\mathcal{E} \subset R$. We say that $\mathcal{E}$ is an exceptional set if for all $a, b \in \mathcal{E}$ with $a \neq b$, we have that $(a-b)$ is invertible in R, i.e., there exists $c \in R$ such that $c \cdot (a-b) = (a-b) \cdot c = 1$.*

Escudero and Soria proved in [14, Prop. 3] that there is an exceptional set of matrices $\mathbb{Z}_{p^k}^{n \times n}$ of size p^n, for a prime p. Following a similar approach, we show how to build an exceptional set $\mathcal{E} \subset \mathbb{F}_q^{n \times n}$ of size q^n, where $\mathbb{F}_q$ is a finite field with q elements.

Proposition 2 (Exceptional set of non-singular matrices). *Let $\mathbb{F}_q$ be a finite field with q elements, $f(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} + x^n \in \mathbb{F}_q[x]$ an irreducible polynomial, and $n \geq 1$ an integer. Let C_f be the companion matrix of f, i.e.,*

$$C_f := \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1}. \end{bmatrix} \in \mathbb{F}_q^{n \times n}.$$

Then, the set

$$\mathcal{E}_f := \left\{ \sum_{i=0}^{n-1} c_i \cdot C_f^i \mid (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n \right\} \subset \mathbb{F}_q^{n \times n}$$

is an exceptional set of q^n matrices containing the zero matrix $O \in \mathbb{F}_q^{n \times n}$. In particular, every non-zero matrix in $\mathcal{E}_f$ is non-singular.

Proof. Since $\mathbb{E} := \mathbb{F}_q[x]/\langle f \rangle$ is a field, any non-zero $g := \sum_{i=0}^{n-1} c_i x \in \mathbb{E}$ defines an invertible $\mathbb{F}_q$-linear map $L_g(a) = a \cdot g$ from $\mathbb{E}$ to $\mathbb{E}$. Hence, the matrix representation $G \in \mathbb{F}_q^{n \times n}$ of L_g in the ordered basis $\{1, x, x^2, \ldots, x^{n-1}\}$ is an invertible matrix. Finally, we note that $G = \sum_{i=0}^{n-1} c_i \cdot C_f^i$. □

4 Matrix-Multiplication Triple Verification

In this section, we present a generalization of the protocol by Baum and Nof [4] that verifies multiplication triples of matrices.

Definition 12 (Matrix-multiplication triple). *Let X, Y, and Z be matrices such that $Z = X \cdot Y$. Then, we say that the triple of sharings $([\![X]\!], [\![Y]\!], [\![Z]\!])$ is a matrix-multiplication triple.*

Similarly to the protocol in [4], in order to verify a matrix-multiplication triple $([\![X]\!], [\![Y]\!], [\![Z]\!])$ we use an additional *random* triple $([\![A]\!], [\![B]\!], [\![C]\!])$, with "random" meaning here that A and B are sampled uniformly at random.

Let n denote the number of rows of the matrix X. Assuming that all the matrix products are well defined, the following MPC protocol verifies the correctness of both triples, $([\![X]\!], [\![Y]\!], [\![Z]\!])$ and $([\![A]\!], [\![B]\!], [\![C]\!])$, without revealing any information on any of them.

1. The parties sample a matrix $R \xleftarrow{\$} \mathcal{E}_f$, where $\mathcal{E}_f$ is defined as in Proposition 2.
2. Each party computes $[\![S_1]\!] = R\,[\![X]\!] + [\![A]\!]$ and $[\![S_2]\!] = [\![Y]\!] + [\![B]\!]$.
3. The parties publish $[\![S_1]\!]$ and $[\![S_2]\!]$ so that all of them can obtain S_1 and S_2.
4. Each party computes $[\![V]\!] = R\,[\![Z]\!] - [\![C]\!] + S_1 \cdot [\![B]\!] + [\![A]\!] \cdot S_2 - S_1 \cdot S_2$.
5. The parties publish $[\![V]\!]$ to obtain V, and they output **accept** if $V = 0$. Otherwise, they output **reject**.

Notice that both matrix-multiplication triples are correct then

$$V = R \cdot Z - C + S_1 \cdot B + A \cdot S_2 - S_1 \cdot S_2 = 0.$$

The following lemma comes directly from the fact that $\mathcal{E}_f$ is an exceptional set.

Lemma 1. *Let $\mathcal{E}_f$ be as in Proposition 2, and $Z, C \in \mathbb{F}_q^{n \times n'}$ with $Z \neq O$. Then, there exist at most one matrix $R \in \mathcal{E}_f$ such that $R \cdot Z = C$.*

Proposition 3. *If $([\![X]\!], [\![Y]\!], [\![Z]\!])$ or $([\![A]\!], [\![B]\!], [\![C]\!])$ is an incorrect multiplication triple, then the parties output **accept** in the above protocol with probability at most $\frac{1}{q^n}$.*

Proof. Let $\Delta_Z = Z - X \cdot Y$ and $\Delta_C = C - A \cdot B$. If the parties output **accept**, this means that $V = 0$, i.e.,

$$\begin{aligned} V &= R \cdot Z - C + S_1 \cdot B + A \cdot S_2 - S_1 \cdot S_2 \\ &= R \cdot (X \cdot Y + \Delta_Z) - (A \cdot B + \Delta_C) + (R \cdot X + A) \cdot B + A \cdot (Y + B) \\ &\qquad - (R \cdot X + A) \cdot (Y + B) \\ &= R \cdot \Delta_Z - \Delta_C = 0. \end{aligned}$$

Now we have the following cases:

- If $\Delta_Z = 0$ and $\Delta_C \neq 0$, then $V \neq 0$, which is in contradiction with the **accept** assumption;
- In the case $\Delta_Z \neq 0$, by Lemma 1, the equality $R \cdot \Delta_Z = \Delta_C$ happens for at most one $R \in \mathcal{E}_f$, whence the upper bound probability $\frac{1}{q^n}$. □

5 A Zero-Knowledge Protocol for MinRank Problem

In this section, we describe a zero-knowledge protocol for the MinRank problem, which is based on the Kipnis-Shamir modeling (see Section 2.4). For simplicity, we describe the protocol for square matrices, but it can be easily generalized for non-square matrices.

We let $\boldsymbol{M} = (M_0, M_1, \ldots, M_k) \subset (\mathbb{F}_q^{n\times n})^{k+1}$ defining an instance of the MinRank problem with parameters $(q, n \times n, k, r)$, and let $\mathcal{E}_f$ be the exceptional set constructed in Proposition 2 (with respect to $\mathbb{F}_q$ and n).

5.1 Description of the protocol

A witness of the statement $\boldsymbol{M} = (M_0, M_1, \ldots, M_k)$ in our proof of knowledge protocol consists of a vector $\boldsymbol{\alpha} \in \mathbb{F}_q^k$ and a matrix $K \in \mathbb{F}_q^{r\times(n-r)}$ such that:

$$M_0^L + \sum_{i=1}^{k} \alpha_i M_i^L = -\left(M_0^R + \sum_{i=1}^{k} \alpha_i M_i^R\right) \cdot K, \tag{3}$$

i.e., $\boldsymbol{M}_{\boldsymbol{\alpha}}^L = \boldsymbol{M}_{\boldsymbol{\alpha}}^R K$. As discussed in Section 2.4, this proves the knowledge of a solution of the MinRank instance defined by $\boldsymbol{M}$.

Our proposed zero-knowledge protocol is described in Fig. 2. This is highly inspired by the Feneuil, Joux, and Rivain protocol presented in [17] for the syndrome decoding problem. It follows the MPC-in-the-head paradigm over the MPC protocol depicted below to verify solutions of instances of the MinRank problem.

An MPC protocol for the MinRank problem In this protocol, we assume that all the parties know the set of input matrices $\boldsymbol{M}$. Also, each party holds a pair of additive shares $(\llbracket \boldsymbol{\alpha} \rrbracket, \llbracket K \rrbracket)$ (where $\boldsymbol{\alpha}$ and K satisfy Eq. (3)) and a triple of shares $(\llbracket A \rrbracket, \llbracket B \rrbracket, \llbracket C \rrbracket)$ of a random matrix-multiplication triple. The protocol proceeds as follows:

1. The parties locally compute $\llbracket \boldsymbol{M}_{\boldsymbol{\alpha}}^L \rrbracket$ and $\llbracket \boldsymbol{M}_{\boldsymbol{\alpha}}^R \rrbracket$.
2. The parties follow the MPC protocol described in Section 4 to verify the multiplication triple $(\llbracket K \rrbracket, \llbracket \boldsymbol{M}_{\boldsymbol{\alpha}}^R \rrbracket, \llbracket \boldsymbol{M}_{\boldsymbol{\alpha}}^L \rrbracket)$ by using the random triple:
   - The parties sample a random matrix $R \xleftarrow{\$} \mathcal{E}_f$.
   - Each party locally sets $\llbracket S_1 \rrbracket = R \llbracket K \rrbracket + \llbracket A \rrbracket$ and $\llbracket S_2 \rrbracket = \llbracket \boldsymbol{M}_{\boldsymbol{\alpha}}^R \rrbracket + \llbracket B \rrbracket$.
   - The parties publish $\llbracket S_1 \rrbracket$ and $\llbracket S_2 \rrbracket$, so that they all obtain S_1 and S_2.
   - Each party locally computes
   $$\llbracket V \rrbracket = R \llbracket \boldsymbol{M}_{\boldsymbol{\alpha}}^L \rrbracket - \llbracket C \rrbracket + S_1 \cdot \llbracket B \rrbracket + \llbracket A \rrbracket \cdot S_2 - S_1 \cdot S_2.$$
   - The parties publish $\llbracket V \rrbracket$ to obtain V.
3. If $V = 0$ the parties output **accept**, otherwise they output **reject**.

Clearly, this MPC protocol proves the knowledge of a matrix $K \in \mathbb{F}_q^{r\times(n-r)}$ and a vector $\boldsymbol{\alpha} \in \mathbb{F}_q^k$ satisfying $\boldsymbol{M}_{\boldsymbol{\alpha}}^L = \boldsymbol{M}_{\boldsymbol{\alpha}}^R K$, i.e., K and $\boldsymbol{\alpha}$ satisfy Eq. (3), whence $\boldsymbol{\alpha}$ is a solution of the MinRank instance given by $\boldsymbol{M}$.

Inputs: Both the prover and the verifier know the MinRank instance $\boldsymbol{M}$. The prover has in addition a vector $\boldsymbol{\alpha} \in \mathbb{F}_q^k$ and a matrix $K \in \mathbb{F}_q^{r \times (n-r)}$ such that $\boldsymbol{M}_{\boldsymbol{\alpha}}^L = \boldsymbol{M}_{\boldsymbol{\alpha}}^R K$.

Round 1: The prover prepares the inputs for the multi-party computation as follows:

1. $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda$.
2. Compute parties' seeds and randomnesses $(\mathsf{seed}_i, \rho_i)_{i \in [N]}$ with $\mathrm{TreePRG}(\mathsf{seed})$.
3. For each party $i \in [N]$,
   - $[\![A]\!]_i, [\![B]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$, where $[\![A]\!]_i \in \mathbb{F}_q^{n \times r}$ and $[\![B]\!]_i \in \mathbb{F}_q^{r \times n-r}$
   - If $i \neq N$,
     - $[\![\boldsymbol{\alpha}]\!]_i, [\![C]\!]_i, [\![K]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$
     - $\mathsf{state}_i = \mathsf{seed}_i$
   - Else,
     - $[\![\boldsymbol{\alpha}]\!]_N = \boldsymbol{\alpha} - \sum_{i \neq N} [\![\boldsymbol{\alpha}]\!]_i$
     - $[\![K]\!]_N = K - \sum_{i \neq N} [\![K]\!]_i$
     - $[\![C]\!]_N = A \cdot B - \sum_{i \neq N} [\![C]\!]_i$
     - $\mathsf{aux} = ([\![\boldsymbol{\alpha}]\!]_N, [\![K]\!]_N, [\![C]\!]_N)$
     - $\mathsf{state}_N = \mathsf{seed}_N \parallel \mathsf{aux}$
   - Commit each party's state: $\mathsf{com}_i = \mathsf{Com}(\mathsf{state}_i, \rho_i)$.

The prover computes $h = \mathrm{Hash}_1(\mathsf{com}_1, \ldots, \mathsf{com}_N)$ and sends it to the verifier.

Round 2: The verifier samples $R \xleftarrow{\$} \mathcal{E}_f$ and sends it to the prover.

Round 3: The prover simulates the MPC protocol:

- The parties locally compute $[\![\boldsymbol{M}_{\boldsymbol{\alpha}}^L]\!]$ and $[\![\boldsymbol{M}_{\boldsymbol{\alpha}}^R]\!]$ using $[\![\boldsymbol{\alpha}]\!]$.
- They locally set $[\![S_1]\!] = R \cdot [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^R]\!] + [\![A]\!]$, and $[\![S_2]\!] = [\![K]\!] + [\![B]\!]$
- The parties open $[\![S_1]\!]$ and $[\![S_2]\!]$ to obtain S_1 and S_2.
- The parties locally set

$$[\![V]\!] = R \cdot \left[\!\!\left[\boldsymbol{M}_{\boldsymbol{\alpha}}^L\right]\!\!\right] - [\![C]\!] + S_1 \cdot [\![B]\!] + [\![A]\!] \cdot S_2 - S_1 \cdot S_2.$$

The prover builds $h' = \mathrm{Hash}_2\left([\![S_1]\!]_1, [\![S_2]\!]_1, [\![V]\!]_1, \ldots, [\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N\right)$ and sends it to the verifier.

Round 4: The verifier uniformly samples $i^* \xleftarrow{\$} [N]$ and sends it to the prover.

Round 5: The prover sends $\mathsf{rsp} = \{(\mathsf{state}_i, \rho_i)_{i \neq i^*}, \mathsf{com}_{i^*}, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}\}$.

Verification: The verifier accepts if and only if all the following checks succeed:

1. For each $i \neq i^*$, she computes all the commitments of the parties' states: $\mathsf{com}_i = \mathsf{Com}(\mathsf{state}_i, \rho_i)$, and she checks that $h = \mathrm{Hash}(\mathsf{com}_1, \ldots, \mathsf{com}_N)$ holds.
2. Using $\{\mathsf{state}_i\}_{i \neq i^*}$, she recomputes all the shares except for the i^*-th. She first checks that $h' = \mathrm{Hash}\left([\![S_1]\!]_1, [\![S_2]\!]_1, [\![V]\!]_1, \ldots, [\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N\right)$ holds, where $[\![V]\!]_{i^*} = -\sum_{i \neq i^*} [\![V]\!]_i$.

Fig. 2. A zero-knowledge proof for MinRank.

5.2 Security proofs

Theorem 1 (Correctness). *The protocol in Fig. 2 is perfectly correct, i.e., a prover with a witness of the underlying MinRank instance always succeeds in convincing a verifier.*

Proof. The correctness of the protocol shown in Fig. 2 follows from the correctness of the protocol to verify matrix-multiplication triples shown in Section 4. It is easy to see that when the prover appropriately executes all the steps in the protocol, all the checks by the verifier pass. □

Theorem 2 (Soundness). *If the commitment scheme* Com *is binding and the hash function* Hash_1 *is collision-resistant, then the protocol in Fig. 2 is sound with soundness error* $\varepsilon = \frac{1}{N} + \frac{N-1}{q^n N}$.

Proof. We first present an adversary $\mathcal{A}$ that cheats with probability at most $\varepsilon = \frac{1}{N} + \frac{N-1}{q^n N}$. $\mathcal{A}$ starts by sampling $(\boldsymbol{\alpha}', K') \xleftarrow{\$} \mathbb{F}_q^k \times \mathbb{F}_q^{r\times(n-r)}$ and makes a random guess $R' \xleftarrow{\$} \mathcal{E}_f$ of the first challenge. Then $\mathcal{A}$ follows every step consistently with Round 1 of the protocol using $(\boldsymbol{\alpha}', K')$, except for the computation of $[\![C]\!]_N$, which is as

$$[\![C]\!]_N = R' \cdot \boldsymbol{M}^L_{\boldsymbol{\alpha}'} - \sum_{i\neq N} [\![C]\!]_i + S'_1 \cdot B + A \cdot S'_2 - S_1 \cdot S_2,$$

where S'_1 and S'_2 are computed as in Round 3 using R'. It is clear that if $\mathcal{A}$ commits on the values generated in Round 1 this way, then whenever the challenge R sent by the verifier in Round 2 coincides with the guessed challenge R', the verification will succeed whatever second challenge $i^* \in [N]$ the verifier sends in Round 4. The event that the verifier's challenge in Round 2 coincides with the guessed challenge occurs with probability $1/q^n$.

Let us assume that $\mathcal{A}$ received from the verifier a challenge $R \neq R'$, then $\mathcal{A}$ can still cheat by making a guess for i^* and change the continuation of the protocol accordingly. Indeed, for a guess $i^* \xleftarrow{\$} [N]$, $\mathcal{A}$ performs all the computations as in Round 3 of the protocol except for $[\![V]\!]_{i^*}$, setting $[\![V]\!]_{i^*} = -\sum_{i\neq i^*} [\![V]\!]_i$. This will pass the verification only if the verifier's second challenge is exactly the guessed i^*. Thus, the overall probability for $\mathcal{A}$ to successfully cheat is

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

Now, we show that doing significantly better than $\mathcal{A}$ means that a witness can be extracted if the commitment scheme Com is binding and the hash function Hash_1 is collision-resistant. This means that if a prover convinces a verifier with a significantly greater probability than ε, then we can construct an extractor that rewinds the prover to efficiently retrieve a witness of the underlying MinRank problem. So, suppose there exists an efficient prover $\tilde{\mathcal{P}}$ such that on input $\boldsymbol{M}$ of a MinRank instance, we have

$$\tilde{\varepsilon} = \Pr\left[\begin{matrix}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ V(\mathsf{pk}, h, R, h', i^*, \mathsf{rsp}) = 1\end{matrix}\right] - \varepsilon$$

is polynomial in λ. Then, rewinding $\tilde{\mathcal{P}}$ with the same internal random tape, an extractor can obtain with non-negligible probability $\tilde{\varepsilon}$ four accepting transcripts $\{(h, R_j, h'_j, i^*_{j,k}, \mathsf{rsp}_{j,k})\}_{(j,k)\in[2]\times[2]}$, with $R_1, R_2 \in \mathcal{E}_f$, $R_1 \neq R_2$.

From the collision-resistance property of Hash_1, for every $i \in [N]$, the commitments com_i are the same in all the four transcripts, and from the binding property of Com, the same holds for state_i and ρ_i.

Let us fix $j \in [2]$ and consider the verifier's first challenge R_j and the corresponding response h'_j from $\tilde{\mathcal{P}}$. Then, there are two different second challenges $i^*_{j,1}$ and $i^*_{j,2}$ for which $\tilde{\mathcal{P}}$ produces two accepting responses, respectively. Since $i^*_{j,1} \neq i^*_{j,2}$, the missing share in one response is obtained in the other. Thus, an extractor easily recovers all the committed values $(\mathsf{state}_i, \rho_i)_{i\in[1,N]}$, and therefore the values of α, K, A, B, C, which are the same for all $j \in [2]$. This implies

$$R_j \cdot \boldsymbol{M}^L_{\boldsymbol{\alpha}} - C + (R_j \cdot \boldsymbol{M}^R_{\boldsymbol{\alpha}} + A) \cdot B + A \cdot (K + B) - (R_j \cdot \boldsymbol{M}^R_{\boldsymbol{\alpha}} + A) \cdot (K + B) = O,$$

for $j \in [2]$. Then, equating the left-hand sides of the above equation for $j = 1$ and for $j = 2$, we have

$$\begin{aligned} &R_1 \cdot \boldsymbol{M}^L_{\boldsymbol{\alpha}} + (R_1 \cdot \boldsymbol{M}^R_{\boldsymbol{\alpha}} + A) \cdot B + A \cdot (K + B) - (R_1 \cdot \boldsymbol{M}^R_{\boldsymbol{\alpha}} + A) \cdot (K + B) = \\ &R_2 \cdot \boldsymbol{M}^L_{\boldsymbol{\alpha}} + (R_2 \cdot \boldsymbol{M}^R_{\boldsymbol{\alpha}} + A) \cdot B + A \cdot (K + B) - (R_2 \cdot \boldsymbol{M}^R_{\boldsymbol{\alpha}} + A) \cdot (K + B). \end{aligned}$$

Hence

$$(R_1 - R_2) \cdot (\boldsymbol{M}^L_{\boldsymbol{\alpha}} - \boldsymbol{M}^R_{\boldsymbol{\alpha}} \cdot K) = O,$$

and since $R_1 - R_2$ is invertible, we necessarily have $\boldsymbol{M}^L_{\boldsymbol{\alpha}} = \boldsymbol{M}^R_{\boldsymbol{\alpha}} \cdot K$, whence a witness for $\boldsymbol{M}$ is extracted. □

Theorem 3 (Honest-Verifier Zero-Knowledge). *In the protocol in Fig. 2, if the PRG is $(t, \varepsilon_{\mathrm{PRG}})$-secure and the commitment scheme $(t, \varepsilon_{\mathsf{com}})$-hiding, then there exists an efficient simulator that outputs transcripts $(t, \varepsilon_{\mathrm{PRG}}+\varepsilon_{\mathsf{com}})$-indistinguishable from real transcripts of the protocol.*

Proof. Here we follow the approach by Feneuil Joux and Rivain in [17]. First, we describe in Fig. 3 an internal HVZK simulator $\mathcal{S}$ and show that its responses are $(t, \varepsilon_{\mathrm{PRG}})$-indistinguishable from the responses of an honest prover for the same challenge i^*. Then we describe a global HVZK simulator that uses $\mathcal{S}$ to output transcripts $(t, \varepsilon_{\mathrm{PRG}} + \varepsilon_{\mathsf{com}})$-indistinguishable from real transcripts of the protocol.

To show the indistinguishability of outputs of simulator $\mathcal{S}$ from outputs of the protocol, we describe the following sequence of simulators.

Simulator 0 (Actual protocol). This simulator, described in Fig. 4, outputs $(\mathsf{rsp}_1, \mathsf{rsp}_2)$ from the transcript of a genuine execution of the protocol with a prover that knowns a witness $(\boldsymbol{\alpha}, K)$ and receives challenges (R, i^*).

Simulator 1. Same as Simulator 0 but uses true randomness instead of seed-derived randomness for party i^*. If $i^* = N$, the values $[\![\boldsymbol{\alpha}]\!]_N$, $[\![K]\!]_N$ and $[\![C]\!]_N$ are computed as described in the protocol (only $[\![A]\!]_N$ and $[\![B]\!]_N$ are generated from true randomness). It is easy to see that the probability of distinguishing Simulator 1 and Simulator 0 in running time t is no more than $\varepsilon_{\mathrm{PRG}}$.

1. Sample a root seed in $\{0,1\}^\lambda$.
2. Compute parties' seeds and randomness $(\mathsf{seed}_i, \rho_i)_{i\in[N]}$ with TreePRG(seed).
3. For each party $i \in [N]\backslash\{i^*\}$,
   - $[\![A]\!]_i, [\![B]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$, where $[\![A]\!]_i \in \mathbb{F}_q^{n\times r}$ and $[\![B]\!]_i \in \mathbb{F}_q^{r\times n-r}$
   - If $i \neq N$,
     - $[\![\boldsymbol{\alpha}]\!]_i, [\![C]\!]_i, [\![K]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$
     - $\mathsf{state}_i = \mathsf{seed}_i$
   - Else,
     - $[\![\boldsymbol{\alpha}]\!]_N \xleftarrow{\$} \mathbb{F}_q^k$
     - $[\![K]\!]_N \xleftarrow{\$} \mathbb{F}_q^{r\times(n-r)}$
     - $[\![C]\!]_N \xleftarrow{\$} \mathbb{F}_q^{n\times(n-r)}$
     - $\mathsf{aux} = ([\![\boldsymbol{\alpha}]\!]_N, [\![K]\!]_N, [\![C]\!]_N)$
     - $\mathsf{state}_N = \mathsf{seed}_N \parallel \mathsf{aux}$
   - Simulate the computation of the party i to obtain $\big([\![S_1]\!]_i, [\![S_2]\!]_i, [\![V]\!]_i\big)$.
4. For party i^*,
   - $\mathsf{com}_{i^*} \leftarrow \mathsf{Com}(\mathsf{state}_{i^*}, \rho_{i^*})$
   - $[\![S_1]\!]_{i^*} \xleftarrow{\$} \mathbb{F}_q^{n\times r}$
   - $[\![S_2]\!]_{i^*} \xleftarrow{\$} \mathbb{F}_q^{r\times n-r}$
   - $[\![V]\!]_{i^*} = -\sum_{i\neq i^*} [\![V]\!]_i$
5. Output the responses
   - $\mathsf{rsp}_1 = \mathrm{Hash}\big([\![S_1]\!]_1, [\![S_2]\!]_1, [\![V]\!]_1, \ldots, [\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N\big)$.
   - $\mathsf{rsp}_2 = \big((\mathsf{state}_i, \rho_i)_{i\neq i^*}, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}\big)$

Fig. 3. Internal HVZK simulator $\mathcal{S}$ on input of a challenge (R, i^*).

Simulator 2. Replace $[\![\boldsymbol{\alpha}]\!]_N$, $[\![K]\!]_N$ and $[\![C]\!]_N$ in Simulator 1 by uniformly random elements of the same type and compute $[\![V]\!]_{i^*} = -\sum_{i\neq i^*} [\![V]\!]_i$. We note that the obtained simulator is independent of the witness $(\boldsymbol{\alpha}, K)$ and solely takes the challenges (R, i^*) as input. Now we show that the output distributions of Simulator 1 and Simulator 2 are identical for $i^* = N$ or $i^* \neq N$.

If $i^* = N$, the changes only impact the shares $[\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N$ in the simulated responses. We can see that the distributions of those shares are identical in Simulator 2 as in Simulator 1. Indeed, in both cases, the shares $[\![S_1]\!]_N$ and $[\![S_2]\!]_N$ are uniformly distributed because of the uniformly sampled (in Simulator 1) additive terms $[\![A]\!]_N$ and $[\![B]\!]_N$, respectively, and independent of the rest. The share $[\![V]\!]_N$, as in Simulation 1, verifies $[\![V]\!]_N = -\sum_{i\neq N} [\![V]\!]_i$.

If $i^* \neq N$, the changes only impact $[\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N$, derived from $\mathsf{aux} = ([\![\boldsymbol{\alpha}]\!]_N, [\![K]\!]_N, [\![C]\!]_N)$, in the simulated response. But aux was already uniformly random in Simulator 1. Indeed, the shares in aux are computed by adding share values from parties $i \neq N$, including party i^* (which is uniformly random in Simulator 1). Therefore, the output distributions of Simulator 1 and Simulator 2 are identical.

1. Sample a root seed in $\{0,1\}^\lambda$.
2. Compute parties' seeds and randomness $(\mathsf{seed}_i, \rho_i)_{i\in[N]}$ with TreePRG(seed).
3. For each party $i \in [N]$,
   - $[\![A]\!]_i, [\![B]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$, where $[\![A]\!]_i \in \mathbb{F}_q^{n\times r}$ and $[\![B]\!]_i \in \mathbb{F}_q^{r\times n-r}$
   - If $i \neq N$,
     - $[\![\boldsymbol{\alpha}]\!]_i, [\![C]\!]_i, [\![K]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$
     - $\mathsf{state}_i = \mathsf{seed}_i$
   - Else,
     - $[\![\boldsymbol{\alpha}]\!]_N = \boldsymbol{\alpha} - \sum_{i\neq N} [\![\boldsymbol{\alpha}]\!]_i$
     - $[\![K]\!]_N = K - \sum_{i\neq N} [\![K]\!]_i$
     - $[\![C]\!]_N = A \cdot B - \sum_{i\neq N} [\![C]\!]_i$
     - $\mathsf{aux} = ([\![\boldsymbol{\alpha}]\!]_N, [\![K]\!]_N, [\![C]\!]_N)$
     - $\mathsf{state}_N = \mathsf{seed}_N \parallel \mathsf{aux}$
   - Perform the computation of the party i to obtain $\left([\![S_1]\!]_i, [\![S_2]\!]_i, [\![V]\!]_i\right)$.
4. Output the responses
   - $\mathsf{rsp}_1 = \mathrm{Hash}\left([\![S_1]\!]_1, [\![S_2]\!]_1, [\![V]\!]_1, \ldots, [\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N\right)$.
   - $\mathsf{rsp}_2 = \left((\mathsf{state}_i, \rho_i)_{i\neq i^*}, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}\right)$

Fig. 4. Simulator 0 on input of a challenge (R, i^*).

Simulator 3 (Internal HVZK simulator). The only difference between Simulator 2 and the internal HVZK simulator $\mathcal{S}$ in Fig. 3 is that the latter directly draws $[\![S_1]\!]_{i^*}$ and $[\![S_2]\!]_{i^*}$ uniformly at random. As explained above, this does not impact the output distribution.

To sum up, we have shown that the internal simulator $\mathcal{S}$ outputs responses $(\mathsf{rsp}_1, \mathsf{rsp}_2)$ which are $(t, \varepsilon_{\mathrm{PRG}})$-indistinguishable from the responses of the real protocol on same challenges of an honest verifier. To obtain a global HVZK simulator, we proceed as in Fig. 5:

Applying the hiding property of the commitment scheme on com_{i^*}, we then have that the global HVZK simulator outputs a transcript which is $(t, \varepsilon_{\mathrm{PRG}} + \varepsilon_{\mathsf{com}})$-indistinguishable from a real transcript of the protocol. □

5.3 Parameters

This section shows our choices of parameters for the MinRank problem underlying the signature scheme presented in this paper. Here, we assume that the MinRank instance is given by $\boldsymbol{M} \in \left(\mathbb{F}_q^{n\times n}\right)^{k+1}$ and the target rank is r. Also, we denote by E the rank-r matrix in the vector space generated by the matrices in $\boldsymbol{M}$.

Table 1 shows our estimates of the bit security estimates for the proposed parameter sets. We suggest a pair of parameter sets for each of the NIST's security categories I, III, and V, i.e., targeting 143, 207, or 273 bits of security, respectively.

1. Sample challenges
   - $R \xleftarrow{\$} \mathcal{E}_f$
   - $i^* \xleftarrow{\$} [N]$

   uniformly at random (as an honest verifier).
2. Run the simulator $\mathcal{S}(R, i^*)$ to obtain
   - $\mathsf{rsp}_1 = \text{Hash}\left(\llbracket S_1 \rrbracket_1, \llbracket S_2 \rrbracket_1, \llbracket V \rrbracket_1, \ldots, \llbracket S_1 \rrbracket_N, \llbracket S_2 \rrbracket_N, \llbracket V \rrbracket_N\right)$.
   - $\mathsf{rsp}_2 = \left((\mathsf{state}_i, \rho_i)_{i \neq i^*}, \llbracket S_1 \rrbracket_{i^*}, \llbracket S_2 \rrbracket_{i^*}\right)$
3. Compute the initial commitment Com as follows
   - For each party $i \neq i^*$, compute the commitment $\mathsf{com}_i = \mathsf{Com}(\mathsf{state}_i, \rho_i)$
   - For party i^*, sample a random commitment $\mathsf{com}_{i^*} \xleftarrow{\$} \{0,1\}^{2\lambda}$
   - Set $h = \text{Hash}(\mathsf{com}_1, \ldots, \mathsf{com}_N)$
   - Update $\mathsf{rsp}_2 = \left((\mathsf{state}_i, \rho_i)_{i \neq i^*}, \mathsf{com}_{i^*}, \llbracket S_1 \rrbracket_{i^*}, \llbracket S_2 \rrbracket_{i^*}\right)$
4. Output the transcript $T = (h, R, \mathsf{rsp}_1, i^*, \mathsf{rsp}_2)$

Fig. 5. The global HVZK simulator.

To estimate the bit-security of the proposed parameters, we first estimate the complexity, in terms of multiplications in $\mathbb{F}_q$, of the most efficient algorithms to solve the MinRank problem. To derive the bit-complexity we assume that every multiplication over $\mathbb{F}_q$ costs $(\log_2 q)^2$ bit operations, and we set $w = 2$ in the below estimates.

To directly solve instances of the MinRank problem, one uses the kernel-search algorithm and the support-minors modeling:

Kernel-search The kernel-search algorithm was introduced in [22]. It consists of guessing $\lceil k/m \rceil$ linearly independent vectors in the kernel of the unknown rank r matrix E. The expected complexity of this algorithm, in terms of multiplications in $\mathbb{F}_q$, is

$$\mathcal{O}\left(q^{r \cdot \lceil k/m \rceil} \cdot k^{\omega}\right),$$

where $2 \leq \omega \leq 3$ is a constant.

Support-minors modelling The support-minors (SM) modeling is an algebraic method to solve the MinRank problem, which was introduced by Bartdet et al. in [1]. It models an instance of the MinRank problem as a bilinear system of equations that are then solved by an XL-like algorithm.

For $q > 2$, the complexity of solving the SM equations is computes as

$$\min\left\{3 \cdot k(r+1) \cdot A(b,n')^2, 7 \cdot A(b,n')^{\omega} \;\middle|\; \begin{smallmatrix} 1 \leq b \leq r+1, \\ r+b \leq n' \leq n, \\ A(b,n')-1 \leq B(b,n') \end{smallmatrix}\right\},$$

Set	q	m	n	k	r	bit security
Ia	16	15	15	79	6	144
Ib	16	16	16	142	4	155
IIIa	16	19	19	115	8	207
IIIb	16	19	19	167	6	229
Va	16	21	21	192	7	273
Vb	16	22	22	254	6	295

Table 1. Proposed parameters for the MinRank problem and their bit security.

where

$$A(b, n') := \binom{n'}{r}\binom{k+b-1}{b},$$
$$B(b, n') := \sum_{j=1}^{b}\sum_{i=1}^{j}\left\{(-1)^{i+1}\binom{n'}{r+i}\binom{m+i-1}{i}\binom{k}{j-i}\right\}.$$

In [3], the authors showed a new guess-and-solve (or hybrid) approach for the MinRank, which is more efficient than directly solving a given instance.

Hybrid approaches We consider two hybridization approaches to estimate the number of multiplications in $\mathbb{F}_q$ required to solve a given MinRank instance. The first approach guesses l_v coefficients of the solution vector $\boldsymbol{\alpha}$. The second approach, introduced in [3], guesses a vectors (with a specific structure) in the right-kernel of the secret E. When both approaches are combined, one derives MinRank instances $\tilde{\boldsymbol{M}}$ with parameters $(q, m \times (n-a), k - am - l_v, r)$. Hence the complexity of this hybrid approach is given by

$$q^{a \cdot r + l_v}\left(\mathsf{MR\_Complexity}(q, m \times (n-a), k - am - l_v, r) + \left(\min\{k, am\}\right)^{\omega}\right), \quad (4)$$

where $\mathsf{MR\_Complexity}(\cdot)$ returns the complexity to solve a random instance of the MinRank problem defined by the input parameters.

For every parameter set, the minimum bit complexity was found for $l_v = 0$. The kernel-search algorithm along with $a = 8$ minimizes the complexity for Ib, and IIIb. The support-minors modeling gives the optimal for the remaining paramters sets; $a = 5$ for Ia, $a = 6$ for IIIa, $a = 9$ for Va and $a = 11$ for Vb.

5.4 Comparison with other MinRank-based zero-knowledge proofs

To the best of our knowledge, the protocol presented in Fig. 2 is the first zero-knowledge protocol based on an MPC protocol, with $N \geq 2$ parties, to prove

Set	Courtois ($\varepsilon = \frac{2}{3}$)	MR-DSS ($\varepsilon = \frac{1}{2}$)	Our ($\varepsilon \approx \frac{1}{256}$)
Ia	153	210	430
Ib	176	220	471
IIIa	229	328	662
IIIb	247	322	678
Va	315	408	863
Vb	339	427	913

Table 2. Communication cost (in bytes) of one round and the soundness error ε of MinRank-based zero-knowledge proofs.

knowledge of a solution to the MinRank problem. This multi-party protocol allows us to obtain a small soundness error compared with previous MinRank-based zero-knowledge proof protocols, that have soundness error of at least $\frac{1}{2}$.

The maximum number of bits sent in one round of our zero-knowledge protocol is given as

$$\underbrace{4\lambda}_{h,h'} + \Big(\underbrace{k + n(n-r) + 2r(n-r) + nr}_{[\![\boldsymbol{\alpha}]\!]_N, [\![C]\!]_N, [\![K]\!]_N, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}} \Big) \cdot \log_2 q + \underbrace{\lambda \cdot \log_2 N}_{\{\mathsf{seed}_i\}_{i \neq i^*}} + \underbrace{2\lambda}_{\mathsf{com}_{i^*}}.$$

As stated in Theorem 2, the soundness error of our protocol is $\varepsilon = \frac{1}{N} + \frac{N-1}{q^n N}$, which is approximately $\frac{1}{N}$ for most of our parameter sets.

The average bit size of the response in the IDS with helper from MR-DSS [5], which has a soundness error of $\frac{1}{2}$, is

$$\frac{\left(n^2 + 2rn\right)\log_2 q}{2} + \frac{\lambda + k\log_2 q}{2}.$$

If we count the communication cost of one round of the protocol without the helper (which is removed by using cut-and-choose), then the bit communication cost of the IDS used in MR-DSS is ([10, Sec.4.5])

$$5\lambda + 2 + \frac{\left(n^2 + 2rn\right)\log_2 q}{2} + \frac{\lambda + k\log_2 q}{2}.$$

The communication cost of Courtois' scheme [11] is given by

$$6\lambda + 2 + \frac{2}{3}n^2 \log_2 q + \frac{2}{3}(\lambda + k\log_2 q).$$

In Table 2, we compare the communication of our scheme with previous MinRank-based zero-knowledge IDS, namely Courtois' scheme and MR-DSS. On the one hand, we observed that the communication cost for one round of

our scheme is about twice larger as MR-DSS and nearly three times larger than Courtois'. On the other hand, our IDS offers a soundness error about 170 times smaller than Courtois' and 128 times smaller than MR-DSS. Therefore, our scheme turns out to be more efficient in terms of the product of the soundness error times the communication cost per round.

6 The Signature Scheme

This section presents our signature scheme. We assume here that $\mathrm{Hash}_0, \mathrm{Hash}_1$, Hash_2 are hash functions, and PRG is a pseudorandom generator.

6.1 Non-interactive zero-knowledge proofs

We start by describing how to turn our interactive honest-verifier zero-knowledge protocol for the MinRank problem in Section 5 (see Fig. 2) into a multi-round non-interactive protocol. Specifically, we use a standard generalization of the Fiat-Shamir transformation [19] for canonical 5-pass IDS protocols, previously used in related works [10,12,17]. In this protocol, the prover simulates τ executions (or rounds) of the canonical 5-pass IDS, and a transcript of the form $(h, \{R^{[\ell]}\}_{\ell\in[\tau]}, h', \{i^{*,[\ell]}\}_{\ell\in[\tau]}, \{\mathsf{rsp}^{[\ell]}\}_{\ell\in[\tau]})$ is produced as a result of these τ executions, where ℓ indicates the ℓ-th execution, and $\mathsf{rsp}^{[\ell]}$ denotes the messages sent by the prover in Round 5 of our proof of knowledge protocol (Fig. 2).

More precisely, for a given random value $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$ and a message $\mathsf{msg} \in \{0,1\}^*$, the prover simulates τ executions of the protocol as follows: She starts by computing $\mathsf{com}^{[\ell]} := (\mathsf{com}_1^{[\ell]}, \ldots, \mathsf{com}_N^{[\ell]})$ just as in the interactive protocol. Next, she calculates $h = \mathrm{Hash}_0(\mathsf{msg}, \mathsf{salt}, \mathsf{com}^{[1]}, \ldots, \mathsf{com}^{[\tau]})$ and produces $R^{[1]}, \ldots, R^{[\tau]} \leftarrow \mathrm{PRG}(h)$. Then, the prover follows τ runs of Round 3 of the interactive protocol to compute

$$\mathsf{rsp}_{1,\ell} := \left(\left[\!\left[S_1^{[\ell]} \right]\!\right]_1, \left[\!\left[S_2^{[\ell]} \right]\!\right]_1, \left[\!\left[V^{[\ell]} \right]\!\right]_1, \ldots, \left[\!\left[S_1^{[\ell]} \right]\!\right]_N, \left[\!\left[S_2^{[\ell]} \right]\!\right]_N, \left[\!\left[V^{[\ell]} \right]\!\right]_N \right)$$

for each of the N MPC parties. The set of second challenges is compute as $[N] \in i^{*,1}, \ldots, i^{*,\tau} \leftarrow \mathrm{PRG}(h')$, where the input $h' = \mathrm{Hash}_0\left(\mathsf{msg}, \mathsf{salt}, h, \mathsf{rsp}_{1,1}, \ldots, \mathsf{rsp}_{1,\tau}\right)$. Finally, the prover uses the challenges $(i^{*,\ell})_{\ell\in[\tau]}$ to compute the responses $(\mathsf{rsp}^{[\ell]})_{\ell\in[\tau]}$ corresponding to the responses generated in Round 5 of the interactive protocol.

We point out that applying the Fiat-Shamir transformation to the interactive proof of knowledge slightly harms the soundness of the protocol. Indeed, the Kales-Zaverucha [24] forgery attack on 5-pass Fiat-Shamir has a cost lower than that of the interactive protocol for the same number τ of repetitions. Thus, the forgery cost, in terms of the number of hashes, that has to be considered in our signature scheme is the following:

$$C(\tau, q, n, N) = \min_{0\le k\le \tau} \left\{ \frac{1}{\sum_{i=k}^{\tau} \left(\frac{1}{q^n}\right)^i \left(1 - \frac{1}{q^n}\right)^{\tau-i} \binom{\tau}{i}} + N^{\tau-k} \right\}. \tag{5}$$

6.2 Description of the signature scheme

We propose two variants for our KeyGen algorithm, Variant 1 and Variant 2. These choices are described in Fig. 6. In both variants, the public key is given by a MinRank instance $\boldsymbol{M}$, while the secret key contains enough information to allow generating a vector $\boldsymbol{\alpha}$ and a matrix K satisfying Eq. (1). Variant 1 aims to be more efficient regarding the algorithm complexity and the secret key size. Instead, Variant 2 targets a smaller public key.

Variant 1:

Input: 1^λ.

1. $\mathsf{seed} \leftarrow \{0,1\}^\lambda$.
2. $\mathsf{seed}_{pk}, \mathsf{seed}_{sk} \leftarrow \mathrm{PRG}(\mathsf{seed})$.
3. $\left(\boldsymbol{\alpha}, K, E^R\right) \leftarrow \mathrm{PRG}(\mathsf{seed}_{sk})$, where $\boldsymbol{\alpha} \in \mathbb{F}_q^k$, $K \in \mathbb{F}_q^{r\times(n-r)}$ and $E^R \in \mathbb{F}_q^{n\times r}$.
4. $E \leftarrow \left[E^R \cdot K | E^R\right]$.
5. $\left(M_1, \ldots, M_k\right) \leftarrow \mathrm{PRG}(\mathsf{seed}_{pk})$.
6. $M_0 \leftarrow -\sum_{i=1}^k \alpha_i M_i + E$.
7. $pk = (M_0, \mathsf{seed}_{pk})$ and $sk = \mathsf{seed}_{sk}$.

Output: (pk, sk).

Variant 2:

Input: 1^λ.

1. Get (pk, sk) for variant 1.
2. Find $\boldsymbol{\beta} \in \mathbb{F}_q$ such that $M_0 - \sum_{i=1}^k \beta_i M_i$ has at least k zero coordinates.
3. $\tilde{M}_0 := M_0 - \sum_{i=1}^k \beta_i M_i$.
4. $pk = (\tilde{M}_0, \mathsf{seed}_{pk})$ and $sk = (\boldsymbol{\beta} + \boldsymbol{\alpha}, \mathsf{seed}_{sk})$

Output: (pk, sk).

Fig. 6. Key generation algorithm for our MinRank-based signature scheme.

The signing algorithm Sign is detailed in Fig. 7. It receives as inputs the secret objects $\boldsymbol{\alpha}, K$, the corresponding public key $\boldsymbol{M}$, and a message $\mathsf{msg} \in \{0,1\}^*$. Then, it outputs a signature σ for the message msg. This algorithm is mounted in such a way that the τ non-interactive rounds of the protocol are executed in an optimized manner.

The verification algorithm Verf, described in Fig. 8, takes as input a public key $\boldsymbol{M}$, a message $\mathsf{msg} \in \{0,1\}^*$ and a signature σ. Verf outputs **accept** if σ is considered as a valid signature of the message msg. Otherwise, it outputs **reject**.

Inputs: An instance $\boldsymbol{M} \in \left(\mathbb{F}_q^{n\times n}\right)^{k+1}$, the secret values $(\boldsymbol{\alpha}, K) \in \mathbb{F}_q^k \times \mathbb{F}_q^{r\times(n-r)}$, and a message $\mathsf{msg} \in \{0,1\}^*$.

$\mathsf{salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$.

Phase 1: Set up the views for the MPC protocols. For each $\ell \in [\tau]$:

1. $\mathsf{seed}^{[\ell]} \xleftarrow{\$} \{0,1\}^{\lambda}$.
2. Compute the parties' seeds $(\mathsf{seed}_i^{[\ell]})_{i\in[N]}$ with $\mathrm{TreePRG}(\mathsf{salt}, \mathsf{seed}^{[\ell]})$.
3. For each party $i \in [N]$,
   - $\left[\!\left[A^{[\ell]}\right]\!\right]_i \in \mathbb{F}_q^{n\times r}$, $\left[\!\left[B^{[\ell]}\right]\!\right]_i \in \mathbb{F}_q^{r\times n-r} \leftarrow \mathrm{PRG}(\mathsf{salt}, \mathsf{seed}_i^{[\ell]})$
   - If $i \neq N$,
     - $\left[\!\left[\boldsymbol{\alpha}^{[\ell]}\right]\!\right]_i$, $\left[\!\left[C^{[\ell]}\right]\!\right]_i$, $\left[\!\left[K^{[\ell]}\right]\!\right]_i \leftarrow \mathrm{PRG}(\mathsf{salt}, \mathsf{seed}_i^{[\ell]})$
     - $\mathsf{state}_i^{[\ell]} = \mathsf{seed}_i^{[\ell]}$
   - Else,
     - $\left[\!\left[\boldsymbol{\alpha}^{[\ell]}\right]\!\right]_N = \boldsymbol{\alpha} - \sum_{i\neq N} \left[\!\left[\boldsymbol{\alpha}^{[\ell]}\right]\!\right]_i$
     - $\left[\!\left[K^{[\ell]}\right]\!\right]_N = K - \sum_{i\neq N} \left[\!\left[K^{[\ell]}\right]\!\right]_i$
     - $\left[\!\left[C^{[\ell]}\right]\!\right]_N = A^{[\ell]} \cdot B^{[\ell]} - \sum_{i\neq N} \left[\!\left[C^{[\ell]}\right]\!\right]_i$
     - $\mathsf{aux}^{[\ell]} = (\left[\!\left[\boldsymbol{\alpha}^{[\ell]}\right]\!\right]_N, \left[\!\left[K^{[\ell]}\right]\!\right]_N, \left[\!\left[C^{[\ell]}\right]\!\right]_N)$
     - $\mathsf{state}_N^{[\ell]} = \mathsf{seed}_N^{[\ell]} \parallel \mathsf{aux}^{[\ell]}$
   - Commit the party's state: $\mathsf{com}_i^{[\ell]} = \mathrm{Hash}_0\left(\mathsf{salt}, \ell, i, \mathsf{state}_i^{[\ell]}\right)$.

Phase 2: First challenges.

1. $h_1 \leftarrow \mathrm{Hash}_1\left(\mathsf{msg}, \mathsf{salt}, \mathsf{com}_1^{[1]}, \mathsf{com}_2^{[1]}, \ldots, \mathsf{com}_N^{[1]}, \mathsf{com}_1^{[2]}, \ldots, \mathsf{com}_{N-1}^{[\tau]}, \mathsf{com}_N^{[\tau]}\right)$
2. $R^{[1]}, \ldots, R^{[\tau]} \leftarrow \mathrm{PRG}(h_1)$, where $R^{[\ell]} \in \mathcal{E}_f$.

Phase 3: Simulation of the MPC protocols. For each $\ell \in [\tau]$:

- The parties compute locally $\left[\!\left[\boldsymbol{M}_{\boldsymbol{\alpha}}^{L,[\ell]}\right]\!\right]$ and $\left[\!\left[\boldsymbol{M}_{\boldsymbol{\alpha}}^{R,[\ell]}\right]\!\right]$ using $\left[\!\left[\boldsymbol{\alpha}^{[\ell]}\right]\!\right]$.
- They locally set $\left[\!\left[S_1^{[\ell]}\right]\!\right] = R^{[\ell]} \cdot \left[\!\left[\boldsymbol{M}_{\boldsymbol{\alpha}}^{R,[\ell]}\right]\!\right] + \left[\!\left[A^{[\ell]}\right]\!\right]$, and $\left[\!\left[S_2^{[\ell]}\right]\!\right] = \left[\!\left[K^{[\ell]}\right]\!\right] + \left[\!\left[B^{[\ell]}\right]\!\right]$.
- The parties open $\left[\!\left[S_1^{[\ell]}\right]\!\right]$ and $\left[\!\left[S_2^{[\ell]}\right]\!\right]$ to get $S_1^{[\ell]}$ and $S_2^{[\ell]}$.
- The parties locally set

$$\left[\!\left[V^{[\ell]}\right]\!\right] = R^{[\ell]} \cdot \left[\!\left[\boldsymbol{M}_{\boldsymbol{\alpha}}^{L,[\ell]}\right]\!\right] - \left[\!\left[C^{[\ell]}\right]\!\right] + S_1^{[\ell]} \cdot \left[\!\left[B^{[\ell]}\right]\!\right] + \left[\!\left[A^{[\ell]}\right]\!\right] \cdot S_2^{[\ell]} - S_1^{[\ell]} \cdot S_2^{[\ell]}.$$

Phase 4: Second challenges.

1. $h_2 \leftarrow \mathrm{Hash}_2\left(\mathsf{msg}, \mathsf{salt}, h_1, \left(\left[\!\left[S_1^{[\ell]}\right]\!\right]_i, \left[\!\left[S_2^{[\ell]}\right]\!\right]_i, \left[\!\left[V_1^{[\ell]}\right]\!\right]_i\right)_{i\in[N],\ \ell\in[\tau]}\right)$.
2. $i^{*,[1]}, \ldots, i^{*,[\tau]} \leftarrow \mathrm{PRG}(h_2)$, where the $i^{*,[\ell]} \in [N]$.

Phase 5: Assembling the signature. Output σ defined by

$$\sigma \leftarrow \mathsf{salt} \mid h_1 \mid h_2 \mid \left(\left(\mathsf{state}_i^{[\ell]}\right)_{i\neq i^{*,[\ell]}} \mid \mathsf{com}_{i^{*,[\ell]}}^{[\ell]} \mid \left[\!\left[S_1^{[\ell]}\right]\!\right]_{i^{*,[\ell]}} \mid \left[\!\left[S_2^{[\ell]}\right]\!\right]_{i^{*,[\ell]}}\right)_{\ell\in[\tau]}$$

Fig. 7. Signing algorithm for our MinRank-based signature scheme.

Inputs: A public key $\boldsymbol{M} \in \left(\mathbb{F}_q^{n\times n}\right)^{k+1}$, a message $\mathsf{msg} \in \{0,1\}^*$, and a signature σ formatted as

$$\sigma = \mathsf{salt} \mid h_1 \mid h_2 \mid \left(\left(\mathsf{state}_i^{[\ell]}\right)_{i\neq i^{*[\ell]}} \mid \mathsf{com}_{i^{*,[\ell]}}^{[\ell]} \mid \left[\!\left[S_1^{[\ell]} \right]\!\right]_{i^{*,[\ell]}} \mid \left[\!\left[S_2^{[\ell]} \right]\!\right]_{i^{*,[\ell]}} \right)_{\ell\in[\tau]}$$

1. Get the first challenges $R^{[1]}, \ldots, R^{[\tau]} \leftarrow \mathrm{PRG}(h_1)$, where $R^{[\ell]} \in \mathbb{F}_q^{m\times n}$.
2. Get the second challenges $i^{*,[1]}, \ldots, i^{*,[\tau]} \leftarrow \mathrm{PRG}(h_2)$, where the $i^{*,[\ell]} \in [N]$.
3. For each $\ell \in [\tau]$:
   - $\mathsf{com}_i^{[\ell]} \leftarrow \mathrm{Hash}_0\left(\mathsf{salt}, \ell, i, \mathsf{state}_i^{[\ell]}\right)$, for each $i \in [N] \setminus i^{*,[\ell]}$.
   - Use $\left(\mathsf{state}_i^{[\ell]}\right)_{i\neq i^{*[\ell]}} \mid \left[\!\left[S_1^{[\ell]} \right]\!\right]_{i^{*,[\ell]}} \mid \left[\!\left[S_2^{[\ell]} \right]\!\right]_{i^{*,[\ell]}}$ and follow Phase 3 of the signing algorithm to compute $\left[\!\left[V^{[\ell]} \right]\!\right]_i$ for all $i \in [N] \setminus i^{*,[\ell]}$.
   - $\left[\!\left[V^{[\ell]} \right]\!\right]_{i^{*,[\ell]}} \leftarrow -\sum_{i\neq i^{*,[\ell]}} \left[\!\left[V^{[\ell]} \right]\!\right]_i$.
4. $h_1' \leftarrow \mathrm{Hash}_1\left(\mathsf{msg}, \mathsf{salt}, \mathsf{com}_1^{[1]}, \mathsf{com}_2^{[1]}, \ldots, \mathsf{com}_N^{[1]}, \mathsf{com}_1^{[2]}, \ldots, \mathsf{com}_{N-1}^{[\tau]}, \mathsf{com}_N^{[\tau]}\right)$
5. $h_2' \leftarrow \mathrm{Hash}_2\left(\mathsf{msg}, \mathsf{salt}, h_1, \left(\left[\!\left[S_1^{[\ell]} \right]\!\right]_i, \left[\!\left[S_2^{[\ell]} \right]\!\right]_i, \left[\!\left[V_1^{[\ell]} \right]\!\right]_i \right)_{i\in[N],\ \ell\in[\tau]}\right)$.
6. Output **accept** if $h_1' = h_1$ and $h_2' = h_2$, otherwise output **reject**

Fig. 8. Verification algorithm for the MinRank-based signature scheme.

6.3 EU-CMA security of the signature scheme

In this section, the security proofs of our signature scheme are presented. We closely follow the proof provided in [17, Theorem 5], which in turn is highly inspired by [9, Theorem 6.2].

Theorem 4. *Suppose the used PRG is $(t, \varepsilon_{\mathrm{PRG}})$-secure and any adversary running in time t has at most an advantage $\varepsilon_{\mathrm{MR}}$ against the underlying MinRank problem in the signature scheme in Section 6.2. Model* Hash_0, Hash_1 *and* Hash_2 *as as random oracles with 2λ-bit output length. Then any adaptive chosen-message adversary against the signature scheme running in time t, making q_s signing queries, q_0 queries to* Hash_0, q_1 *queries to* Hash_1, *and q_2 queries to* Hash_2 *succeeds in outputting a valid forgery with probability at most*

$$\Pr\left[\mathsf{Forge}\right] \leq \frac{(q_0 + \tau N q_s)^2}{2\cdot 2^{2\lambda}} + \frac{q_s(q_s+q_0+q_1+q_2)}{2^{2\lambda}} + \tau q_s \varepsilon_{\mathrm{PRG}} + \varepsilon_{\mathrm{MR}} + \frac{q_2}{C(\tau,q,n,n)},$$

with $C(\tau,q,n,n)$ as given in Eq. (5).

Proof. Let $\mathcal{A}$ be an adversary making q_s signing queries, and q_0, q_1, q_2 queries, respectively, to the random oracles Hash_0, Hash_1 and Hash_2. To prove the theorem, we define in the following a sequence of experiments involving $\mathcal{A}$. We let $\Pr_i[\cdot]$ refer to the probability of an event in experiment i, and t denote the running time of the entire experiment, i.e., including $\mathcal{A}$'s running time, the time required to answer signing queries and to verify $\mathcal{A}$'s output.

Note that since Hash_0, Hash_1, and Hash_2 are modeled as random oracles, $\mathcal{A}$ can know the output of one of these on a prepared input only if it queries the oracle. Hence, if $\mathcal{A}$ outputs a forgery (msg, σ) at the end of an experiment, with

$$\sigma = \mathsf{salt} \mid h_1 \mid h_2 \mid \Big(\big(\mathsf{state}_i^{[\ell]}\big)_{i \neq i^{*,[\ell]}} \mid \mathsf{com}_{i^{*,[\ell]}}^{[\ell]} \mid \Big[\!\Big[S_1^{[\ell]} \Big]\!\Big]_{i^{*,[\ell]}} \mid \Big[\!\Big[S_2^{[\ell]} \Big]\!\Big]_{i^{*,[\ell]}} \Big)_{\ell \in [\tau]},$$

then there necessarily exists, at a given moment during the experiment, a query to Hash_2 made by $\mathcal{A}$ itself with input

$$X = \Big(\mathsf{msg}, \mathsf{salt}, h_1, \Big(\Big[\!\Big[S_1^{[\ell]} \Big]\!\Big]_i, \Big[\!\Big[S_2^{[\ell]} \Big]\!\Big]_i, \Big[\!\Big[V_1^{[\ell]} \Big]\!\Big]_i \Big)_{i \in [N],\, \ell \in [\tau]}\Big),$$

such that $h_2 = \mathrm{Hash}_2(X)$.

Experiment 1. This corresponds to the interaction of $\mathcal{A}$ with the real signature scheme. In more detail: first KeyGen is run to obtain $\boldsymbol{M}, \boldsymbol{\alpha}, K$, and $\mathcal{A}$ is given the public key $\boldsymbol{M}$. At the end of this experiment, $\mathcal{A}$ outputs a message/signature pair. We let Forge denote the event that the message was not previously queried by $\mathcal{A}$ to its signing oracle, and the signature is valid. Our goal is to upper-bound $\Pr_1[\mathsf{Forge}]$.

Experiment 2. This is the previous experiment with the difference that we abort if, during the course of the experiment, a collision in Hash_0 is found. Note that the number of queries to Hash_0 throughout the experiment (by either the adversary or the signing algorithm) is $q_0 + \tau N q_s$. Thus,

$$|\Pr_1[\mathsf{Forge}] - \Pr_2[\mathsf{Forge}]| \leq \frac{(q_0 + \tau N q_s)^2}{2 \cdot 2^{2\lambda}}.$$

Experiment 3. The difference with the previous experiment is that, when signing a message m, we begin by choosing h_1 and h_2 uniformly and then expand them as the challenges $\{R^{[1]}, \ldots, R^{[\tau]}\}$ and $\{i^{*,[1]}, \ldots, i^{*,[\tau]}\}$. Phases 1, 3 and 5 of Fig. 7 remain unchanged, but in phases 2 and 4 we simply set the output of Hash_1 to h_1 and the output of Hash_2 to h_2.

A difference in the outcome of this experiment compared to the previous one occurs only when, in the course of answering a signing query, the query to Hash_1 or the query to Hash_2 was ever made before by $\mathcal{A}$. The probability of each of these two events is upper bounded by that of having the same salt in the current signing query and in the relevant previous query, which is $\frac{1}{2^{2\lambda}}$. Therefore, we have

$$|\Pr_2[\mathsf{Forge}] - \Pr_3[\mathsf{Forge}]| \leq \frac{q_s \cdot (q_1 + q_2)}{2^{2\lambda}}.$$

Experiment 4. The difference with the previous experiment is that, for each $\ell \in [\tau]$, we sample $\mathsf{com}_{i^{*,[\ell]}}^{[\ell]}$ uniformly at random (i.e., without making the corresponding query to Hash_0).

A difference between this experiment and the previous one occurs only when, in the course of answering a signing query, Hash_0 receives an input that it was

previously queried. However, such a collision cannot occur within the same signing query (since the indices i and ℓ are part of the input to Hash_0), and it occurs from a previous query (signing query or Hash_0 query made by the $\mathcal{A}$) with probability $\frac{1}{2^{2\lambda}}$ since there would be the same salt in the current signing query as in that previous query. Thus,

$$|\Pr_3[\mathsf{Forge}] - \Pr_4[\mathsf{Forge}]| \leq \frac{q_s \cdot (q_s + q_0)}{2^{2\lambda}}.$$

Experiment 5. We again modify the experiment. Now, for $\ell \in [\tau]$, the signer uses the internal HVZK simulator in Fig. 3 to generate the parties' views in one execution of Phases 1 and 3. We denote $\mathcal{S}_{\mathsf{salt}}(\cdot)$ a call to this simulator which appends salt to the sampled seed in input to TreePRG. Thus, signature queries are now answered as depicted in Fig. 9.

Phase 0.

1. Sample $h_1 \xleftarrow{\$} \{0,1\}^{2\lambda}$.
2. Compute $R^{[1]}, \ldots, R^{[\tau]} \leftarrow \mathrm{PRG}(h_1)$, where $R^{[\ell]} \in \mathcal{E}_f$.
3. Sample $h_2 \xleftarrow{\$} \{0,1\}^{2\lambda}$.
4. Compute $i^{*,[1]}, \ldots, i^{*,[\tau]} \leftarrow \mathrm{PRG}(h_2)$, where the $i^{*,[\ell]} \in [N]$.
5. Sample $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$.

Phase 1 and 3. For each $\ell \in [\tau]$:

1. $\left(\mathsf{state}_i^{[\ell]}\right)_{i \neq i^{*,[\ell]}}, \left(\left[\!\left[S_1^{[\ell]} \right]\!\right]_i, \left[\!\left[S_2^{[\ell]} \right]\!\right]_i, \left[\!\left[V_1^{[\ell]} \right]\!\right]_i \right)_{i \in [N]} \leftarrow \mathcal{S}_{\mathsf{salt}}(R^{[\ell]}, i^{*,[\ell]})$.
2. Sample $\mathsf{com}_{i^*} \xleftarrow{\$} \{0,1\}^{2\lambda}$ and for $i \neq i^*$, compute $\mathsf{com}_i = \mathsf{Com}(\mathsf{state}_i, \rho_i)$

Phase 2 and 4.

1. Set $\mathrm{Hash}_1\left(\mathsf{msg}, \mathsf{salt}, \mathsf{com}_1^{[1]}, \mathsf{com}_2^{[1]}, \ldots, \mathsf{com}_{N-1}^{[\tau]}, \mathsf{com}_N^{[\tau]}\right)$ equal to h_1.
2. Set $\mathrm{Hash}_2\left(\mathsf{msg}, \mathsf{salt}, h_1, \left(\left[\!\left[S_1^{[\ell]} \right]\!\right]_i, \left[\!\left[S_2^{[\ell]} \right]\!\right]_i, \left[\!\left[V_1^{[\ell]} \right]\!\right]_i \right)_{i \in [N],\ \ell \in [\tau]}\right)$ equal to h_2.

Phase 5: Assembling the signature. Output σ defined as

$$\sigma \leftarrow \mathsf{salt} \mid h_1 \mid h_2 \mid \left(\left(\mathsf{state}_i^{[\ell]}\right)_{i \neq i^{*,[\ell]}} \mid \mathsf{com}_{i^{*,[\ell]}}^{[\ell]} \mid \left[\!\left[S_1^{[\ell]} \right]\!\right]_{i^{*,[\ell]}} \mid \left[\!\left[S_2^{[\ell]} \right]\!\right]_{i^{*,[\ell]}} \right)_{\ell \in [\tau]}$$

Fig. 9. Experiment 5: Response to a signature query for a message msg.

Observe that the secret $(\boldsymbol{\alpha}, K)$ is no longer used for generating signatures. Recall that an adversary against the internal HVZK simulator has a distinguishing advantage $\varepsilon_{\mathrm{PRG}}$ (corresponding to execution time t) since commitments are built outside of the simulator. It results in

$$|\Pr_4[\mathsf{Forge}] - \Pr_5[\mathsf{Forge}]| \leq \tau \cdot q_s \cdot \varepsilon_{\mathrm{PRG}}.$$

Experiment 6. We first define the following concept. At any point during an experiment, say that the execution ℓ^* of a query to Hash_2

$$\mathrm{Hash}_2\left(\mathsf{msg}, \mathsf{salt}, h_1, \left(\left[\!\left[S_1^{[\ell]}\right]\!\right]_i, \left[\!\left[S_2^{[\ell]}\right]\!\right]_i, \left[\!\left[V_1^{[\ell]}\right]\!\right]_i\right)_{i\in[N],\, \ell\in[\tau],}\right)$$

is correct if:

1. The value h_1 was already output by a previous query to Hash_1 of the form
$$h_1 = \mathrm{Hash}_1\left(\mathsf{msg}, \mathsf{salt}, \mathsf{com}_1^{[1]}, \ldots, \mathsf{com}_N^{[\tau]}\right).$$
2. Each $\mathsf{com}_i^{[\ell^*]}$ was output by a previous query (by either $\mathcal{A}$ or the signing oracle) to Hash_0 of the form
$$\mathsf{com}_i^{[\ell]} = \mathrm{Hash}_0\left(\mathsf{salt}, \ell, i, \mathsf{state}_i^{[\ell]}\right).$$
3. A witness $(\boldsymbol{\alpha}, K)$ can be extracted from $\{\mathsf{state}_i^{[\ell^*]}\}_{i\in[N]}$.

In this experiment, it is checked for each query made by $\mathcal{A}$ to Hash_2 (where msg was not previously queried to the signing oracle) if there is a correct execution. We call this event Solve. Note that if Solve occurs then the $\{\mathsf{state}_i^{[\ell]}\}_{i\in[N]}$ (which can be determined from the oracle queries of $\mathcal{A}$) allow to easily recover a solution $(\boldsymbol{\alpha}, K)$ of the MinRank instance given by $\boldsymbol{M}$. Thus, (by hypothesis)

$$\Pr_6[\mathsf{Solve}] \leq \varepsilon_{\mathrm{MR}}.$$

Hence,

$$\begin{aligned}\Pr_6[\mathsf{Forge}] &= \Pr_6[\mathsf{Forge} \wedge \mathsf{Solve}] + \Pr_6[\mathsf{Forge} \wedge \overline{\mathsf{Solve}}] \\ &\leq \varepsilon_{\mathrm{MR}} + \Pr_6[\mathsf{Forge} \wedge \overline{\mathsf{Solve}}].\end{aligned}$$

Now we bound $\Pr_4[\mathsf{Forge} \wedge \overline{\mathsf{Solve}}]$. For this, it is easy to see that if Solve does not occur, then the best way $\mathcal{A}$ can forge a signature is to run the forgery attacker of Kales and Zaverucha [24] (see Section 6.2), and this succeeds with probability $1/C(\tau, q, n, n)$, with $C(\tau, q, n, n)$ as given in Eq. (5). Taking a union bound over all queries to Hash_2, we obtain

$$\Pr_6[\mathsf{Forge} \wedge \overline{\mathsf{Solve}}] \leq \frac{q_2}{C(\tau, q, n, n)}.$$

6.4 Parameters and signature size

The parameters of the underlying MinRank problem are shown in Table 1. The maximum bit size of our signature scheme is

$$4\lambda + \tau\left(\underbrace{(k + n(n-r) + 2r(n-r) + nr)\cdot \log_2 q}_{[\![\boldsymbol{\alpha}]\!]_N, [\![C]\!]_N, [\![K]\!]_N, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}} + \underbrace{\lambda \cdot \log_2 N}_{\{\mathsf{seed}_i\}_{i\neq i^*}} + \underbrace{2\lambda}_{\mathsf{com}_{i^*}}\right).$$

Table 3 shows the signature sizes for all the proposed parameter sets. The values of N and τ denote the number of repetitions in the IDS and the number of parties in the MPC protocol, respectively. These two values are set to achieve a soundness error, in the non-interactive protocol, smaller than $2^{-\lambda}$.

Set	Variant	λ	N	τ	Maximum signature size
Ia	Fast	128	16	34	10364
	Short		256	18	6695
Ib	Fast	128	16	34	11758
	Short		256	18	7422
IIIa	Fast	192	16	51	24114
	Short		256	27	14832
IIIb	Fast	192	16	51	24930
	Short		256	27	15858
Va	Fast	256	16	67	40827
	Short		256	35	25934
Vb	Fast	256	16	67	44211
	Short		256	35	27667

Table 3. Key and signature sizes in bytes.

7 Comparisons with other Signatures Schemes

Here, we compare our signature scheme with some proposals in the literature.

Table 4 shows the signature and public key sizes of Courtois's scheme [11], MR-DSS [5], and our scheme for the set of parameters that were proposed in [5]. For those parameter sets, which minimized the signature size of MR-DSS, our signature size is at least 3.75 (*resp.* 8.76) times smaller than MR-DSS (*resp.* Courtois' scheme). In terms of security, our scheme is comparable with previous proposals since all of them rely on the hardness of random instances of the MinRank problem. Finally, our scheme is as efficient as MR-DSS for public key sizes.

Set	Signature (KB)			Public key (B)		
	Courtois	MR-DSS	This work	Courtois	MR-DSS	This work
Ib	65	27	7.2	144	73	73
IIIb	135	60	15.4	205	121	121
Vb	248	106	27	274	147	147

Table 4. Signature size comparison with other MinRank-based schemes.

Our signature scheme is competitive when compared with schemes selected by NIST for standardization. For instance, for category I parameters, our scheme offers signatures 15% (*resp.* 39%) shorter than the short (*resp.* fast) version of

SPHINCS+. For the same category, our signatures are 2.5 times larger than Dilithium. Still, in terms of security, Dilithium is based on a structured problem from lattices, while our scheme is based on random instances of a well-known NP-complete problem that is hard in the average case.

8 Conclusions and Future work

In this paper, we proposed a digital signature scheme that we proved it is EU-CMA secure based on the hardness of random instances of the MinRank problem. The scheme follows the MPC-in-the-Head paradigm with a underlying MPC protocol that verifies a shared solution. Our proposal provides signatures significantly smaller than previous MinRank-based proposals. Moreover, our scheme improves over SPHINCS+ in terms of signature size and over Dilithium on hardness assumption.

Future efforts will be considered to provide an efficient implementation our scheme. This will help to assess the concrete efficiency of our proposed scheme in terms of signing and verification time. Additionally, it would be interesting to investigate in the future new techniques to further reduce the signature size. For instance, it is worthwhile investigating if using a *threshold linear secret sharing scheme* (as considered in [18]) yields more compact signatures. Finally, from the cryptanalytic side, it would be interesting to know how efficient implementations of algorithms for the MinRank problem scale in practice; this shall help to understand the security of the here-proposed signature scheme in practice.

References

1. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.P., Verbel, J.: Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In: Advances in cryptology—ASIACRYPT 2020. Part I, vol. 12491, pp. 507–536 (2020)
2. Bardet, M., Bertin, M.: Improvement of algebraic attacks for solving superdetermined MinRank instances. CoRR **abs/2208.01442** (2022). https://doi.org/10.48550/arXiv.2208.01442
3. Bardet, M., Briaud, P., Bros, M., Gaborit, P., Tillich, J.P.: Revisiting algebraic attacks on MinRank and on the rank decoding problem. Cryptology ePrint Archive, Paper 2022/1031 (2022), https://eprint.iacr.org/2022/1031
4. Baum, C., Nof, A.: Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) Public-Key Cryptography – PKC 2020. pp. 495–526. Springer International Publishing, Cham (2020)
5. Bellini, E., Esser, A., Sanna, C., Verbel, J.: MR-DSS – smaller MinRank-based (ring-)signatures. Cryptology ePrint Archive, Paper 2022/973 (2022), https://eprint.iacr.org/2022/973
6. Beullens, W.: Improved cryptanalysis of UOV and Rainbow. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 348–373. Springer International Publishing, Cham (2021)

7. Beullens, W.: Breaking rainbow takes a weekend on a laptop. IACR Cryptol. ePrint Arch. p. 214 (2022), https://eprint.iacr.org/2022/214
8. Buss, J.F., Frandsen, G.S., Shallit, J.O.: The computational complexity of some problems of linear algebra. Journal of Computer and System Sciences **58**(3), 572 – 596 (1999), http://www.sciencedirect.com/science/article/pii/S0022000098916087
9. Chase, M., Derler, D., Goldfeder, S., Katz, J., Kolesnikov, V., Orlandi, C., Ramacher, S., Rechberger, C., Slamanig, D., Wang, X., Zaverucha, G.: The picnic signature scheme. Design Document. Version 3.0 (2020), https://github.com/microsoft/Picnic/blob/master/spec/spec-v3.0.pdf
10. Chen, M.S., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures. In: Cheon, J.H., Takagi, T. (eds.) Advances in Cryptology – ASIACRYPT 2016. pp. 135–165. Springer Berlin Heidelberg, Berlin, Heidelberg (2016)
11. Courtois, N.T.: Efficient zero-knowledge authentication based on a linear algebra problem MinRank. In: Advances in cryptology—ASIACRYPT 2001 (Gold Coast), vol. 2248, pp. 402–421 (2001)
12. Dagdelen, O., Galindo, D., Véron, P., El Yousfi Alaoui, S.M., Cayrel, P.L.: Extended security arguments for signature schemes. Des. Codes Cryptography **78**(2), 441–461 (feb 2016), https://doi.org/10.1007/s10623-014-0009-7
13. Dalskov, A., Lee, E., Soria-Vazquez, E.: Circuit amortization friendly encodingsand their application to statistically secure multiparty computation. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 213–243. Springer International Publishing, Cham (2020)
14. Escudero, D., Soria-Vazquez, E.: Efficient information-theoretic multi-party computation over non-commutative rings. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 335–364. Springer International Publishing, Cham (2021)
15. Faugère, J., Din, M.S.E., Spaenlehauer, P.: Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology. In: Symbolic and Algebraic Computation, International Symposium, ISSAC. pp. 257–264 (2010), http://doi.acm.org/10.1145/1837934.1837984
16. Faugère, J., Levy-dit-Vehel, F., Perret, L.: Cryptanalysis of MinRank. In: Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. pp. 280–296 (2008)
17. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. Cryptology ePrint Archive, Paper 2022/188 (2022), https://eprint.iacr.org/2022/188
18. Feneuil, T., Rivain, M.: Threshold linear secret sharing to the rescue of mpc-in-the-head. Cryptology ePrint Archive, Paper 2022/1407 (2022), https://eprint.iacr.org/2022/1407
19. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. pp. 186–194. Springer Berlin Heidelberg, Berlin, Heidelberg (1987)
20. Gaborit, P., Ruatta, O., Schrek, J.: On the complexity of the rank syndrome decoding problem. IEEE Transactions on Information Theory **62**(2), 1006–1019 (2016)
21. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM J. Comput. **17**(2), 281–308 (apr 1988), https://doi.org/10.1137/0217017
22. Goubin, L., Courtois, N.T.: Cryptanalysis of the TTM Cryptosystem, pp. 44–57. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)

23. Ishai, Y., Kushilevitz, E., Ostrovsky, R., Sahai, A.: Zero-knowledge from secure multiparty computation. p. 21–30. STOC '07, Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1250790.1250794
24. Kales, D., Zaverucha, G.: An attack on some signature schemes constructed from five-pass identification schemes. In: Krenn, S., Shulman, H., Vaudenay, S. (eds.) Cryptology and Network Security. pp. 3–22. Springer International Publishing, Cham (2020)
25. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M. (ed.) Advances in Cryptology – CRYPTO 99. pp. 19–30. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
26. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M. (ed.) Advances in Cryptology — CRYPTO' 99. pp. 19–30. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
27. Lindell, Y., Nof, A.: A framework for constructing fast mpc over arithmetic circuits with malicious adversaries and an honest-majority. Association for Computing Machinery, New York, NY, USA (2017), https://doi.org/10.1145/3133956.3133999
28. Delpech de Saint Guilhem, C., Orsini, E., Tanguy, T.: Limbo: Efficient zero-knowledge mpcith-based arguments. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. p. 3022–3036. CCS '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3460120.3484595
29. Santoso, B., Ikematsu, Y., Nakamura, S., Yasuda, T.: Three-pass identification scheme based on MinRank problem with half cheating probability. CoRR **abs/2205.03255** (2022). https://doi.org/10.48550/arXiv.2205.03255
30. Tao, C., Petzoldt, A., Ding, J.: Efficient key recovery for all HFE signature variants. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021. pp. 70–93. Springer International Publishing, Cham (2021)
31. Verbel, J., Baena, J., Cabarcas, D., Perlner, R., Smith-Tone, D.: On the complexity of "superdetermined" minrank instances. In: Ding, J., Steinwandt, R. (eds.) Post-Quantum Cryptography. PQCrypto 2019. pp. 167–186 (2019)