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 paradigm, allowing 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 its 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 she 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 [22] introduced the MPC-in-the-head paradigm to build zero-knowledge proofs of knowledge from *multi-party computation* (MPC) protocols. 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 [9]. It is defined by an integer r and a set of matrices $M_0, M_1, \dots, M_k$ 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 appeared first in cryptology in the cryptanalysis of the HFE cryptosystem [24]. There, 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. The hardness of MinRank seems to be a reasonable assumption to make when designing secure *post-quantum schemes*. First, due to its relevance in cryptanalysis [7,8,20,28], several classical algorithms solving this problem have been extensively studied [1,2,3,12,15,16,24,29]. 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 that solve the MinRank problem.

Despite the believed hardness of MinRank, 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 [12], MR-DSS [11], and the scheme by Santoso, Ikematsu, Nakamura and Yasuda [27].

Our Contributions. We propose a provable secure signature scheme based on the hardness of the MinRank problem by introducing the first MPC protocol specifically designed to prove knowledge of a solution of an instance of the MinRank problem. Our scheme is highly inspired by the work of Feneuil, Joux, and Rivain [17] by following their MPC-in-the-head paradigm, then obtaining an honest-verifier zero-knowledge 5-pass identification scheme, and finally using the generalized Fiat-Shamir transformation to turn the identification scheme into a digital signature scheme. To make our adaption from [17] work, we needed to bring the following novelties.

– Generalize the simultaneous verification of two scalar multiplication triples of [4]to matrix multiplication triples.
– Employ the Kipnis-Shamir modeling to translate MinRank instances for the verification of matrix multiplication triples. This considerably improves the communication cost of our signature scheme over the usual rank decomposition modeling.
– Sample the second challenge in our 5-pass identification scheme and signature scheme from the so-called *exceptional sets of matrices* to ease the soundness and unforgeabilty analysis.

We obtain a soundness error of $\frac{1}{N} + \frac{N-1}{q^n N}$ for our MinRank-based identification scheme, where N is the number of parties in the MPC protocol, n the number of columns of every M_i ($0 \leq i \leq k$), and q the size of the base field.

Assuming the hardness of MinRank, we prove that our scheme is *existentially unforgeable under chosen messages attack* (EUF-CMA). We propose several parameter sets for the new scheme. Each of them targets security levels 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] while achieving public keys of similar sizes. We stress that our work extends naturally to ring signatures in the same way as MR-DSS, see [5, Section 5] for more details.

Related works. MinRank-based signature schemes built on zero-knowledge proofs started with Courtois' scheme in 2001 [12], where the identification scheme 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 [27] also proposed a MinRank-based signature scheme with proofs of soundness error 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 finite field have been proposed by Baum and Nof in [4] (see also [26]). To design a new signature scheme based on the syndrome decoding problem, Feneuil, Joux, and Rivain used in [17] the Baum-Nof MPC protocol to verify multiplication triples of shares of univariate polynomials. As mentioned above, 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 $[n] = \{1, 2, \dots, n\}$. 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 $(M_0, M_1, \dots, M_k) \in (\mathbb{F}_q^{m \times n})^{k+1}$. For a given vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_k)^T \in \mathbb{F}_q^k$, 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, and $a \overset{\$}{\leftarrow} \mathcal{S}$ means 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*

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

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 $p(\lambda) > \lambda$ and the distributions $\{G(s) \mid s \xleftarrow{\$} \{0,1\}^\lambda\}$ and $\{r \mid r \xleftarrow{\$} \{0,1\}^{p(\lambda)}\}$ are (t,ε)-indistinguishable.*

In the description of our signing algorithm (Fig. 4), we make call of a function $\mathsf{TreePRG}$, which, on input of a root seed in $\{0,1\}^\lambda$ and a positive integer N, generates a binary tree of N leaves using a PRG : $\{0,1\}^\lambda \to \{0,1\}^{2\lambda}$, such that each nodes has seed value in $\{0,1\}^\lambda$ (we refer to [6, Section 2.6]).

2.1 Commitment Schemes

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

Definition 4 (Computational hiding). *A commitment scheme Com is said to be (t,ε)-hiding if for every pair of messages (m, m'), the distributions $\{c \mid c \leftarrow \mathsf{Com}(m,\rho), \rho \xleftarrow{\$} \{0,1\}^\lambda)\}$ and $\{c \mid c \leftarrow \mathsf{Com}(m',\rho), \rho \xleftarrow{\$} \{0,1\}^\lambda)\}$ are (t,ε)-indistinguishable. Com is said 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 Signature Schemes

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

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* Sign *takes a message* $\mathsf{msg} \in \{0,1\}^*$ *and a private key* sk*, and 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.*

A signature scheme is defined to be secure if it has the following properties.

Correctness: if 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 (EUF-CMA): if for all probabilistic polynomial-time adversaries $\mathcal{A}$, the probability

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

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

An identification scheme (IDS) is an interactive protocol between a *prover* P and a *verifier* V, where P wants to prove its knowledge of a secret value sk to V, with $(\mathsf{pk}, \mathsf{sk})$ satisfying a given relation, for a public value pk.

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

1. $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. P *and* V *follow the protocol in Fig. 1, and at the end of this,* V *outputs 1, if it* ***accepts*** *that* P *knows* sk*, otherwise it* ***rejects*** *outputting 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. We require an IDS to fulfill the following security properties.

Correctness: if for any $\lambda \in \mathbb{N}$ and $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ it holds, where $(\mathsf{com}, \mathsf{ch}_1, \mathsf{rsp}_1, \mathsf{ch}_2, \mathsf{rsp}_2)$ is the transcript of an execution of the protocol between $\mathsf{P}(\mathsf{pk}, \mathsf{sk})$ and $\mathsf{V}(\mathsf{pk})$.

Soundness (with soundness error ε): if, given a key pair $(\mathsf{pk}, \mathsf{sk})$, for every polynomial-time adversary $\tilde{\mathcal{P}}$ such that

$$\tilde{\varepsilon} = \Pr\left[1 \leftarrow \mathsf{V}(\mathsf{pk}, \mathsf{com}_{\tilde{\mathcal{P}}}, \mathsf{ch}_1, \mathsf{rsp}_{1,\tilde{\mathcal{P}}}, \mathsf{ch}_2, \mathsf{rsp}_{2,\tilde{\mathcal{P}}})\right] > \varepsilon,$$

where $(\mathsf{com}_{\tilde{\mathcal{P}}}, \mathsf{ch}_1, \mathsf{rsp}_{1,\tilde{\mathcal{P}}}, \mathsf{ch}_2, \mathsf{rsp}_{2,\tilde{\mathcal{P}}})$ is the transcript of an execution of the protocol between $\tilde{\mathcal{P}}(\mathsf{pk})$ and $\mathsf{V}(\mathsf{pk})$, then there exists an *extractor* $\mathcal{E}$ which, given rewindable black-box access to $\tilde{\mathcal{P}}$, outputs in time polynomial in $(\lambda, \tilde{\varepsilon} - \varepsilon)$ a *witness* $\tilde{\mathsf{sk}}$ such that $(\mathsf{pk}, \tilde{\mathsf{sk}})$ verifies the same relation as $(\mathsf{pk}, \mathsf{sk})$.

Honest-verifier zero-knowledge: if there exists a probabilistic polynomial-time *simulator* $\mathcal{S}(\mathsf{pk})$ that 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 $\mathsf{P}(\mathsf{pk}, \mathsf{sk})$ and $\mathsf{V}(\mathsf{pk})$.

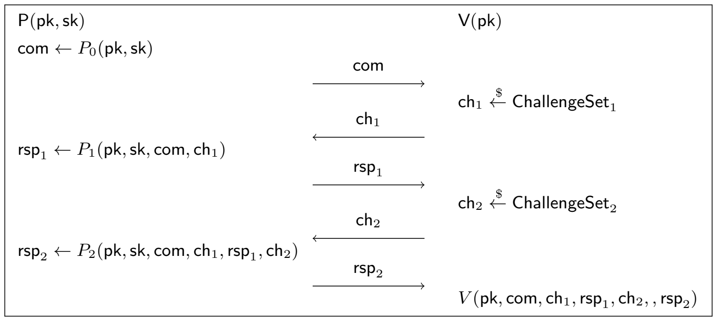
$\mathsf{P}(\mathsf{pk}, \mathsf{sk})$
$\mathsf{V}(\mathsf{pk})$
$\mathsf{com} \leftarrow P_0(\mathsf{pk}, \mathsf{sk})$
com
$\mathsf{ch}_1 \xleftarrow{\$} \mathsf{ChallengeSet}_1$
ch_1
$\mathsf{rsp}_1 \leftarrow P_1(\mathsf{pk}, \mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$
rsp_1
$\mathsf{ch}_2 \xleftarrow{\$} \mathsf{ChallengeSet}_2$
ch_2
$\mathsf{rsp}_2 \leftarrow P_2(\mathsf{pk}, \mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{rsp}_1, \mathsf{ch}_2)$
rsp_2
$V(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{rsp}_1, \mathsf{ch}_2, , \mathsf{rsp}_2)$

Fig. 1. Canonical 5-pass IDS.

2.4 The MinRank problem

This section details the underlying hard problem of our signature scheme.

Problem 1 (The MinRank problem). Let $\mathbb{F}_q$ be a finite field with q elements, and m, n, k, r be positive integers. The MinRank problem with parameters (q, m, n, k, r) is defined as: given a $(k+1)$-tuple $\boldsymbol{M} = (M_0, M_1, \ldots, M_k) \in (\mathbb{F}_q^{m \times n})^{k+1}$, find $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)^T \in \mathbb{F}_q^k$ such that $\mathsf{Rank}\big(M_0 + \sum_{i=1}^{k} \alpha_i M_i\big) \leq r$.

The Kipnis-Shamir modeling [25] states that if a vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)^T \in \mathbb{F}_q^k$ and a matrix $K \in \mathbb{F}_q^{r \times (n-r)}$ are such that

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

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

To solve an instance of the MinRank problem, Kipnis and Shamir proposed to solve the bilinear system where the 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 enables N parties to collaboratively compute the image $z = f(x_1, \dots, x_N)$ for a given function f, where each secret value x_i is only known by the i-th party. An MPC protocol is considered secure and correct if, upon completion, every party i knows z, while at the same time, preserving the confidentiality of the input values x_i.

A core concept in MPC protocols is that of *sharing*. We say that a tuple $[\![A]\!] := ([\![A]\!]_1, [\![A]\!]_2, \dots, [\![A]\!]_N)$, where each $[\![A]\!]_i$ is called a *share* of A, is a sharing of a value A with *threshold* $t < N$ if A can be efficiently recovered from any t'-sized subset of elements in $[\![A]\!]$ with $t' > t$. When A is an element in a ring, we say that $[\![A]\!]$ is an additive sharing of A if $A = \sum_{i=1}^{N} [\![A]\!]_i$. In this paper we only deal with additive sharings, which have threshold $t = N - 1$.

The operations $[\![A + B]\!]$, $[\![A]\!] \cdot C$ and $C \cdot [\![A]\!]$, for shared values A, B and non-shared C, mean that every party $i \in [N]$ computes $[\![A]\!]_i + [\![B]\!]_i$, $[\![A]\!]_i \cdot C$ and $C \cdot [\![A]\!]_i$, respectively. However, the operations $[\![A]\!] + C$, for a shared value A and non-shared C, means that only the party $i = 1$ computes $[\![A]\!]_1 + C$.

3 Exceptional Sets of Matrices over a Finite Fields

In our zero-knowledge protocol, Section 5, the first challenge space consists of an *exceptional set* of square matrices. Following an approach similar to [14, Prop. 3], we build such sets of matrices over finite fields.

Proposition 1 (Exceptional set of non-singular matrices). *Let $\mathbb{F}_q$ be a finite field with q elements, f an irreducible polynomial of degree $n \geq 1$. Let C_f be the companion matrix of f. Then,*

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

is an exceptional set of q^n matrices, i.e, for all $R, R' \in \mathcal{E}_f$ with $R \neq R'$, we have that $(R - R')$ is invertible.

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 $\{1, x, x^2, \dots, x^{n-1}\}$ 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, i.e, a triple of matrix sharings $([\![X]\!], [\![Y]\!], [\![Z]\!])$ such that $Z = X \cdot Y$.

Similarly to the protocol in [4], to verify a matrix-multiplication triple $([\![X]\!], [\![Y]\!], [\![Z]\!])$ we use a *random* triple $([\![A]\!], [\![B]\!], [\![C]\!])$. Assuming that all the matrix products are well defined, the following MPC protocol verifies the correctness of both triples, $([\![X]\!], [\![Y]\!], [\![Z]\!])$ and $([\![A]\!], [\![B]\!], [\![C]\!])$ in an MPC manner.

1. The parties sample a matrix $R \xleftarrow{\$} \mathcal{E}_f$, where $\mathcal{E}_f$ is defined as in Proposition 1.
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. The first party computes $[\![V]\!] = R\,[\![Z]\!] - [\![C]\!] + S_1 \cdot [\![B]\!] + [\![A]\!] \cdot S_2 - S_1 \cdot S_2$, and each other party computes $[\![V]\!] = R\,[\![Z]\!] - [\![C]\!] + S_1 \cdot [\![B]\!] + [\![A]\!] \cdot S_2$.
5. The parties reveal $[\![V]\!]$ to have V. They **accept** if $V = 0$, or **reject** otherwise.

Proposition 2. *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 $\frac{1}{|\mathcal{E}_f|} = \frac{1}{q^n}$.*

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., $R \cdot \Delta_Z - \Delta_C = 0$, and 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 $\Delta_Z \neq 0$, the equality $R \cdot \Delta_Z = \Delta_C$ happens for at most one $R \in \mathcal{E}_f$, since $\mathcal{E}_f$ is an exceptional set, whence the bound probability $\frac{1}{q^n}$. $\square$

5 A Zero-Knowledge Protocol on the MinRank Problem

We present in this section a zero-knowledge protocol based the MinRank problem, using the Kipnis-Shamir modeling. For simplicity, we describe the protocol for square matrices, but it can be generalized to non-square matrices.

We let $\boldsymbol{M} = (M_0, M_1, \ldots, M_k) \subset (\mathbb{F}_q^{n \times n})^{k+1}$ be an instance of the MinRank problem with parameters (q, n, k, r). Notice that in Eq. (1) of the Kipnis-Shamir modeling, if we fix I as the identity matrix of size $n - r$, then it holds

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

denoted $\boldsymbol{M}_{\boldsymbol{\alpha}}^L = \boldsymbol{M}_{\boldsymbol{\alpha}}^R K$, where $M_i^L \in \mathbb{F}_q^{n \times (n-r)}$, $M_i^R \in \mathbb{F}_q^{n \times r}$ satisfy $M_i = [M_i^L | M_i^R]$, for all $i \in [0, k]$. The pair $(\boldsymbol{\alpha}, K)$ is called a witness of $\boldsymbol{M}$.

5.1 Description of the protocol

Our proposed zero-knowledge protocol is described in Fig. 2, where $\mathcal{E}_f$ is as in Proposition 1. 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 $([\![\boldsymbol{\alpha}]\!], [\![K]\!])$ (with $\boldsymbol{\alpha}, K$ satisfying Eq. (2)) and a triple of shares $([\![A]\!], [\![B]\!], [\![C]\!])$ of a random matrix-multiplication triple. The protocol proceeds as follows:

1. The parties locally compute $[\![\boldsymbol{M}_{\boldsymbol{\alpha}}^L]\!]$ and $[\![\boldsymbol{M}_{\boldsymbol{\alpha}}^R]\!]$.

```
$\mathsf{ZKProof}(\mathsf{Prover}(\boldsymbol{M}, \boldsymbol{\alpha}, K),\ \mathsf{Verifier}(\boldsymbol{M}))$
Round 1: Prover sets up the inputs for the MPC protocol
  1 : $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda$,   $(\mathsf{seed}_i, \rho_i)_{i \in [N]} \leftarrow \mathrm{TreePRG}(\mathsf{seed})$
  2 : For each party $i \in [N-1]$
        $[\![A]\!]_i, [\![B]\!]_i, [\![\boldsymbol{\alpha}]\!]_i, [\![C]\!]_i, [\![K]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$
        $\mathsf{state}_i \leftarrow \mathsf{seed}_i$
  3 : $[\![A]\!]_N, [\![B]\!]_N \leftarrow \mathrm{PRG}(\mathsf{seed}_N)$,   $[\![\boldsymbol{\alpha}]\!]_N \leftarrow \boldsymbol{\alpha} - \sum_{i \neq N} [\![\boldsymbol{\alpha}]\!]_i$
  4 : $[\![K]\!]_N \leftarrow K - \sum_{i \neq N} [\![K]\!]_i$,   $[\![C]\!]_N \leftarrow A \cdot B - \sum_{i \neq N} [\![C]\!]_i$
  5 : $\mathsf{aux} \leftarrow ([\![\boldsymbol{\alpha}]\!]_N, [\![K]\!]_N, [\![C]\!]_N)$,   $\mathsf{state}_N \leftarrow \mathsf{seed}_N \parallel \mathsf{aux}$
  6 : Commit to each party's state: $\mathsf{com}_i \leftarrow \mathsf{Com}(\mathsf{state}_i, \rho_i)$, for all $i \in [N]$
  7 : Prover computes $h \leftarrow \mathrm{Hash}_1(\mathsf{com}_1, \ldots, \mathsf{com}_N)$ and sends it to Verifier
Round 2: Verifier samples $R \xleftarrow{\$} \mathbb{F}_q^{n \times n}$ and sends it to Prover
Round 3: Prover simulates the MPC protocol:
  8 : The parties locally compute $[\![\boldsymbol{M}_{\boldsymbol{\alpha}}^L]\!]$ and $[\![\boldsymbol{M}_{\boldsymbol{\alpha}}^R]\!]$ using $[\![\boldsymbol{\alpha}]\!]$
  9 : They locally set $[\![S_1]\!] = R \cdot [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^R]\!] + [\![A]\!]$, and $[\![S_2]\!] = [\![K]\!] + [\![B]\!]$
  10 : The parties open $[\![S_1]\!]$ and $[\![S_2]\!]$ to obtain $S_1$ and $S_2$
  11 : The parties locally set
        $[\![V]\!] = R \cdot [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^L]\!] - [\![C]\!] + S_1 \cdot [\![B]\!] + [\![A]\!] \cdot S_2 - S_1 \cdot S_2$
  12 : Prover sets $h' = \mathrm{Hash}_2\left([\![S_1]\!]_1, [\![S_2]\!]_1, [\![V]\!]_1, \ldots, [\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N\right)$
  13 : Prover sends $h'$ to Verifier
Round 4: Verifier uniformly samples $i^* \xleftarrow{\$} [N]$ and sends it to Prover
Round 5: Prover sends $\mathsf{rsp} = \{(\mathsf{state}_i, \rho_i)_{i \neq i^*}, \mathsf{com}_{i^*}, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}\}$ to Verifier
Verification: Verifier accepts if and only if the following checks succeed:
  14 : $\mathsf{com}_i \leftarrow \mathsf{Com}(\mathsf{state}_i, \rho_i)$, for each $i \neq i^*$
  15 : Check that $h = \mathrm{Hash}_1(\mathsf{com}_1, \ldots, \mathsf{com}_N)$
  16 : Compute $\{[\![S_1]\!]_i, [\![S_2]\!]_i\}_{i \neq i^*}$ from $\{\mathsf{state}_i\}_{i \neq i^*}$, and $\{[\![V]\!]_i\}_{i \neq i^*}$
  17 : $[\![V]\!]_{i^*} \leftarrow -\sum_{i \neq i^*} [\![V]\!]_i$
  18 : Check that $h' = \mathrm{Hash}_2\left([\![S_1]\!]_1, [\![S_2]\!]_1, [\![V]\!]_1, \ldots, [\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N\right)$
```

Fig. 2. Zero-knowledge proof for MinRank.

2. The parties follow the MPC protocol described in Section 4 to verify the multiplication triple $\left([\![K]\!], [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^R]\!], [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^L]\!]\right)$ by using the random triple.

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

5.2 Security proofs

The security properties of the zero-knowledge protocol are stated in the following theorems. Since the proofs of Theorem 2 and Theorem 3 are similar to the corresponding ones from [17], we provide them in Appendix A and Appendix B.

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). *Assume the commitment scheme* Com *is binding and the hash function* Hash_1 *is collision-resistant. Suppose there exists an efficient prover* $\tilde{\mathcal{P}}$ *that, on input* $\boldsymbol{M}$*, convinces the honest verifier* V *on input* $\boldsymbol{M}$ *to accept with probability*

$$\tilde{\varepsilon} = \Pr[\mathsf{V}(\boldsymbol{M}, h_{\tilde{\mathcal{P}}}, R, h'_{\tilde{\mathcal{P}}}, i^*, \mathsf{rsp}_{\tilde{\mathcal{P}}}) \to 1] > \varepsilon,$$

where $\varepsilon = \frac{1}{q^n} + (1 - \frac{1}{q^n}) \cdot \frac{1}{N}$*, then there exists an efficient probabilistic extraction algorithm* $\mathcal{E}$ *that, given rewindable black-box access to* $\tilde{\mathcal{P}}$*, produces a witness* $(\boldsymbol{\alpha}, K)$ *verifying* $\boldsymbol{M}^L_{\boldsymbol{\alpha}} = \boldsymbol{M}^R_{\boldsymbol{\alpha}} K$ *by making an average number of calls to* $\tilde{\mathcal{P}}$ *upper bounded by*

$$\frac{4}{\tilde{\varepsilon} - \varepsilon}\left(1 + \tilde{\varepsilon} \cdot \frac{2 \cdot \ln(2)}{\tilde{\varepsilon} - \varepsilon}\right).$$

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.*

5.3 Complexity of the MinRank problem

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

Table 1 shows our 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, targeting 143, 207, and 273 bits of security, respectively.

This is done by first estimating the complexity, in terms of multiplications in $\mathbb{F}_q$, of the most efficient algorithms to solve the MinRank problem. Then, we assume that every multiplication over $\mathbb{F}_q$ costs $(\log_2 q)^2$ bit operations, and we set $w = 2$ in the estimates below.

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

Kernel-search. This algorithm was introduced in [21], and consists of guessing $\lceil k/n \rceil$ linearly independent vectors in the kernel of the unknown rank r matrix E. The expected number of $\mathbb{F}_q$multiplications of this algorithm is

$$\mathcal{O}\left(q^{r \cdot \lceil k/n \rceil} \cdot k^{\omega}\right), \text{ where } 2 \leq \omega \leq 3 \text{ is a constant.}$$

Support-minors modelling (SM). This is an algebraic method to solve the MinRank problem, by Bardet et al. [1]. It models an instance as a bilinear system of equations that are then solved with an XL-like algorithm.

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

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

$$\text{where} \quad A(b,n') := \binom{n'}{r}\binom{k+b-1}{b},$$

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

In [3], it is shown that a guess-and-solve (or hybrid) approach for the MinRank is more efficient than directly solving the problem.

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 [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 $(q, m \times (n-a), k - am - l_v, r)$. The complexity of this hybrid approach is given by

$$q^{a \cdot r + l_v}\left(\mathsf{MR\_Complexity}(q, n \times (n-a), k - an - l_v, r) + \left(\min\{k, an\}\right)^\omega\right), \quad (3)$$

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 in Table 1, the minimum bit complexity was found for $l_v = 0$. The kernel-search algorithm for $a = 8$ minimizes the complexity for Ib, and IIIb. The SM modeling gives the optimal complexity for the remaining parameters sets; $a = 5$ for Ia, $a = 6$ for IIIa, $a = 9$ for Va and $a = 11$ for Vb.

6 The Signature Scheme

This section presents our signature scheme. We let $\mathrm{Hash}_0, \mathrm{Hash}_1$, Hash_2 be collision-resistant hash functions, and PRG a secure 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, 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. In the non-interactive protocol, the prover simulates τ executions (or rounds) of the canonical 5-pass IDS, which results in a transcript of the form $(h, \{R^{[\ell]}\}_{\ell\in[\tau]}, h', \{i^{*,[\ell]}\}_{\ell\in[\tau]}, \{\mathsf{rsp}^{[\ell]}\}_{\ell\in[\tau]})$, where $\mathsf{rsp}^{[\ell]}$, from the ℓ-th execution, corresponds to the reponse of the prover in Round 5 of Fig. 2.

More precisely, for a given random value $\mathsf{salt} \overset{\$}{\leftarrow} \{0,1\}^{2\lambda}$ 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 Fig. 2. 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 Fig. 2 to compute

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

The set of second challenges is computed as $i^{*,1}, \ldots, i^{*,\tau} \leftarrow \mathrm{PRG}(h')$, where $h' = \mathrm{Hash}_0\left(\mathsf{msg}, \mathsf{salt}, h, \mathsf{rsp}_{1,1}, \ldots, \mathsf{rsp}_{1,\tau}\right)$. 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 Fig. 2.

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 [23] 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:

$$C(\tau, q, n, N) = \min_{0\le k\le\tau}\left\{\frac{1}{\sum_{i=k}^{\tau}\left(\frac{1}{q^n}\right)^i\left(1-\frac{1}{q^n}\right)^{\tau-i}\binom{\tau}{i}} + N^{\tau-k}\right\}. \tag{4}$$

6.2 Description of the signature scheme

Fig. 3 proposes a KeyGen algorithm generating a public key and a secret key that decompress into a MinRank instance $\boldsymbol{M}$ and a witness $(\boldsymbol{\alpha}, K)$ satisfying Eq. (1). Another variant providing smaller public keys is proposed by Di Scala and Sanna [13].

The signing algorithm Sign is detailed in Fig. 4. It receives as inputs the secret $(\boldsymbol{\alpha}, K)$, the corresponding public key $\boldsymbol{M}$, and a message $\mathsf{msg} \in \{0,1\}^*$. Then, it outputs a signature σ for the message msg.

The verification algorithm Verf, Fig. 4, takes as input the 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**.

6.3 EUF-CMA security of the signature scheme

Theorem 4 provides the security guarantees of our signature scheme in the random oracle model. Its proof, presented in Appendix C, closely follows the one of [17, Theorem 5], which in turn is highly inspired by that of [10, Theorem 6.2].

```
KeyGen$(1^\lambda)$
1 : $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda, \quad \mathsf{seed}_{\mathsf{pk}}, \mathsf{seed}_{\mathsf{sk}} \leftarrow \mathrm{PRG}(\mathsf{seed})$
2 : $(\boldsymbol{\alpha}, K, E^R) \leftarrow \mathrm{PRG}(\mathsf{seed}_{\mathsf{sk}})$, where $\boldsymbol{\alpha} \in \mathbb{F}_q^k, K \in \mathbb{F}_q^{r\times(n-r)}$, and $E^R \in \mathbb{F}_q^{m\times r}$
3 : $E \leftarrow [E^R \cdot K \mid E^R]$
4 : $(M_1, \ldots, M_k) \leftarrow \mathrm{PRG}(\mathsf{seed}_{\mathsf{sk}}, \quad M_0 \leftarrow E - \sum_{i=1}^{k} \alpha_i M_i$
5 : Output $(\mathsf{pk}, \mathsf{sk}) : \mathsf{pk} \leftarrow (M_0, \mathsf{seed}_{\mathsf{pk}})$ and $\mathsf{sk} \leftarrow \mathsf{seed}_{\mathsf{sk}}$
```

Fig. 3. Key generation algorithm for our MinRank-based signature scheme.

Theorem 4. *Suppose that 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 Section 6.2. Then modelling* Hash_0, Hash_1 *and* Hash_2 *as random oracles, 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 to* Hash_1, *and q_2 to* Hash_2 *succeeds in outputting a valid forgery with probability at most*

$$\Pr[\mathsf{Forge}] \leq \frac{(q_0 + \tau N q_s)^2}{2 \cdot 2^{2\lambda}} + \frac{q_s(q_s + q_0 + q_1 + q_2)}{2^{2\lambda}} + \tau q_s \varepsilon_{\mathrm{PRG}} + \varepsilon_{\mathrm{MR}} + \max_{s \in [\tau]} P(s),$$

where $P(s) = \left(1 - \left[1 - \left(\frac{1}{q^n}\right)^s \left(1 - \frac{1}{q^n}\right)^{\tau - s} \binom{\tau}{s}\right]^{q_1}\right)\left(1 - \left[1 - \frac{1}{N^{\tau-s}}\right]^{q_2}\right).$

6.4 Parameters and signature size

The maximum bit size of our signature scheme is given as

$$6\lambda + \tau\left(\underbrace{(k + n(n-r) + 2r(n-r) + nr) \cdot \log_2 q}_{[\![\boldsymbol{\alpha}]\!]_N, [\![C]\!]_N, [\![K]\!]_N, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}} + \underbrace{\lambda \cdot \log_2 N}_{\{\mathsf{seed}_i\}_{i \neq i^*}} + \underbrace{2\lambda}_{\mathsf{com}_{i^*}}\right).$$

Table 1 shows the proposed parameter sets and corresponding signature sizes. The values of N and τ denote the number of parties in the MPC protocol and the number of repetitions in the IDS, respectively. These two values are set to achieve a soundness error, in the non-interactive protocol, smaller than $2^{-\lambda}$.

7 Comparisons with other Signatures Schemes

In this section, we compare our signature scheme with some proposals in the literature. Table 2 shows the signature and public key sizes of Courtois's scheme [12], MR-DSS [5], and our scheme for the sets of parameters proposed in [5]. For those parameter sets, minimizing 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 these schemes

```
$\mathsf{Sign}(\boldsymbol{M}, \boldsymbol{\alpha}, K, \mathsf{msg})$
  $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$
Phase 1: Set up the views for the MPC protocols
  for $\ell \in [\tau]$ do
    1 : $\mathsf{seed}^{[\ell]} \xleftarrow{\$} \{0,1\}^{\lambda}$,   $(\mathsf{seed}_i^{[\ell]})_{i\in[N]} \leftarrow \mathrm{TreePRG}(\mathsf{salt}, \mathsf{seed}^{[\ell]})$
    2 : for $i \in [N-1]$ do
          $[\![A^{[\ell]}]\!]_i, [\![B^{[\ell]}]\!]_i, [\![\boldsymbol{\alpha}^{[\ell]}]\!]_i, [\![C^{[\ell]}]\!]_i, [\![K^{[\ell]}]\!]_i \leftarrow \mathrm{PRG}(\mathsf{salt}, \mathsf{seed}_i^{[\ell]})$
          $\mathsf{state}_i^{[\ell]} \leftarrow \mathsf{seed}_i^{[\ell]}$
    3 : $[\![A^{[\ell]}]\!]_N, [\![B^{[\ell]}]\!]_N \leftarrow \mathrm{PRG}(\mathsf{salt}, \mathsf{seed}_N^{[\ell]})$,   $[\![\boldsymbol{\alpha}^{[\ell]}]\!]_N \leftarrow \boldsymbol{\alpha} - \sum_{i\neq N} [\![\boldsymbol{\alpha}^{[\ell]}]\!]_i$
    4 : $[\![K^{[\ell]}]\!]_N \leftarrow K - \sum_{i\neq N} [\![K^{[\ell]}]\!]_i$,   $[\![C^{[\ell]}]\!]_N \leftarrow A^{[\ell]} \cdot B^{[\ell]} - \sum_{i\neq N} [\![C^{[\ell]}]\!]_i$
    5 : $\mathsf{aux}^{[\ell]} \leftarrow ([\![\boldsymbol{\alpha}^{[\ell]}]\!]_N, [\![K^{[\ell]}]\!]_N, [\![C^{[\ell]}]\!]_N)$,   $\mathsf{state}_N^{[\ell]} \leftarrow \mathsf{seed}_N^{[\ell]} \parallel \mathsf{aux}^{[\ell]}$
    6 : $\mathsf{com}_i^{[\ell]} \leftarrow \mathrm{Hash}_0\left(\mathsf{salt}, \ell, i, \mathsf{state}_i^{[\ell]}\right)$, for all $i \in [N]$
Phase 2: First challenges
    7 : $h_1 \leftarrow \mathrm{Hash}_1\left(\mathsf{msg}, \mathsf{salt}, \mathsf{com}_1^{[1]}, \ldots, \mathsf{com}_N^{[1]}, \mathsf{com}_1^{[2]}, \ldots, \mathsf{com}_N^{[\tau]}\right)$
    8 : $R^{[1]}, \ldots, R^{[\tau]} \leftarrow \mathrm{PRG}(h_1)$
Phase 3: Simulation of the MPC protocols
  for $\ell \in [\tau]$ do
    9 : Compute $[\![\boldsymbol{M}_{\boldsymbol{\alpha}}^{L,[\ell]}]\!], [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^{R,[\ell]}]\!]$ from $[\![\boldsymbol{\alpha}^{[\ell]}]\!]$
    10 : $[\![S_1^{[\ell]}]\!] \leftarrow R^{[\ell]} \cdot [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^{R,[\ell]}]\!] + [\![A^{[\ell]}]\!]$,   $[\![S_2^{[\ell]}]\!] \leftarrow [\![K^{[\ell]}]\!] + [\![B^{[\ell]}]\!]$
    11 : $S_1^{[\ell]} \leftarrow \sum_i [\![S_1^{[\ell]}]\!]_i$,   $S_2^{[\ell]} \leftarrow \sum_i [\![S_2^{[\ell]}]\!]_i$
    12 : $[\![V^{[\ell]}]\!] \leftarrow R^{[\ell]} \cdot [\![\boldsymbol{M}_{\boldsymbol{\alpha}}^{L,[\ell]}]\!] - [\![C^{[\ell]}]\!] + S_1^{[\ell]} \cdot [\![B^{[\ell]}]\!] + [\![A^{[\ell]}]\!] \cdot S_2^{[\ell]} - S_1^{[\ell]} \cdot S_2^{[\ell]}$
Phase 4: Second challenges
    13 : $h_2 \leftarrow \mathrm{Hash}_2\left(\mathsf{msg}, \mathsf{salt}, h_1, ([\![S_1^{[\ell]}]\!]_i, [\![S_2^{[\ell]}]\!]_i, [\![V_1^{[\ell]}]\!]_i)_{i\in[N],\, \ell\in[\tau]}\right)$
    14 : $i^{*,[1]}, \ldots, i^{*,[\tau]} \leftarrow \mathrm{PRG}(h_2)$
Phase 5: Assembling the signature $\sigma$
    15 : $\sigma \leftarrow \mathsf{salt} \mid h_1 \mid h_2 \mid \left( \left(\mathsf{state}_i^{[\ell]}\right)_{i\neq i^{*,[\ell]}} \mid \mathsf{com}_{i^{*,[\ell]}}^{[\ell]} \mid [\![S_1^{[\ell]}]\!]_{i^{*,[\ell]}} \mid [\![S_2^{[\ell]}]\!]_{i^{*,[\ell]}} \right)_{\ell\in[\tau]}$
```

```
$\mathsf{Verf}(\boldsymbol{M}, \mathsf{msg}, \sigma)$
    1 : $R^{[1]}, \ldots, R^{[\tau]} \leftarrow \mathrm{PRG}(h_1)$,   $i^{*,[1]}, \ldots, i^{*,[\tau]} \leftarrow \mathrm{PRG}(h_2)$
    2 : for all $\ell \in [\tau]$, all $i \in [N] \backslash i^{*,[\ell]}$ do
          $\mathsf{com}_i^{[\ell]} \leftarrow \mathrm{Hash}_0\left(\mathsf{salt}, \ell, i, \mathsf{state}_i^{[\ell]}\right)$,
          Compute $[\![V^{[\ell]}]\!]_i$ as in $\mathsf{Sign}()$ using $\left(\mathsf{state}_i^{[\ell]}\right)_{i\neq i^{*[\ell]}}, [\![S_1^{[\ell]}]\!]_{i^{*,[\ell]}}, [\![S_2^{[\ell]}]\!]_{i^{*,[\ell]}}$
    3 : $[\![V^{[\ell]}]\!]_{i^{*,[\ell]}} \leftarrow -\sum_{i\neq i^{*,[\ell]}} [\![V^{[\ell]}]\!]_i$
    4 : $h_1' \leftarrow \mathrm{Hash}_1\left(\mathsf{msg}, \mathsf{salt}, \mathsf{com}_1^{[1]}, \ldots, \mathsf{com}_N^{[1]}, \mathsf{com}_1^{[2]}, \ldots, \mathsf{com}_{N-1}^{[\tau]}, \mathsf{com}_N^{[\tau]}\right)$
    5 : $h_2' \leftarrow \mathrm{Hash}_2\left(\mathsf{msg}, \mathsf{salt}, h_1, ([\![S_1^{[\ell]}]\!]_i, [\![S_2^{[\ell]}]\!]_i, [\![V_1^{[\ell]}]\!]_i)_{i\in[N],\, \ell\in[\tau]}\right)$
    6 : Output accept if $h_1' = h_1$ and $h_2' = h_2$, otherwise output reject
```

Fig. 4. Signing and verification algorithms for our MinRank-based signature scheme.

since they all rely on the hardness of random instances of the MinRank problem. Finally, our public keys are as short as MR-DSS.

Our signature scheme remains competitive when compared with schemes selected by NIST for standardization as shown in Table 3.

The optimal performance anylysis for each proposed variant is left for future work. However, given the similarity between our scheme and the one in [17], we

Set	Variant	λ (bits)	q	n	k	r	N	τ	Bit security	Maximum signature size
Ia	Fast	128	16	15	79	6	16	34	144	10364
	Short						256	18		6695
Ib	Fast	128	16	16	142	4	16	34	155	11758
	Short						256	18		7422
IIIa	Fast	192	16	19	115	8	16	51	207	24114
	Short						256	27		14832
IIIb	Fast	192	16	19	167	6	16	51	229	24930
	Short						256	27		15858
Va	Fast	256	16	21	192	7	16	67	273	40827
	Short						256	35		25934
Vb	Fast	256	16	22	254	6	16	67	295	44211
	Short						256	35		27667

Table 1. Parameter sets and signature sizes of the proposed signature scheme.

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 2. Size comparison with other MinRank-based schemes.

made estimates of the computational cost for our category I parameters based on their performance. It results then that the signing time for Ia-short and Ib-short in our scheme could be ≤ 30 milliseconds, while the signing time for SPHINCS-SHA2-128s-simple is approximately 207 milliseconds. On the other hand, our scheme offers signatures 1.45 times shorter than SPHINCS$^+$, but 4.8 times larger than Dilithium. However, 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. Thus, we demonstrated that our scheme is highly competitive in terms of size and computational cost compared to similar schemes.

8 Conclusions and Future work

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

	This work		SPHINCS$^+$		Dilithium
	Fast	Short	Fast	Short	
Security level	1	1	1	1	2
Bit security	155	155	128	133	192
Public key size (bytes)	73	73	32	32	1,312
Signature size (bytes)	11,758	7,422	17,088	7,856	2,420

Table 3. Comparison with SPHINCS$^+$ and Dilithium schemes.

Future efforts will be considered to provide an efficient implementation of our scheme. This will help to assess the concrete efficiency of it 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 the implementations of algorithms for solving the MinRank problem scale in practice; this shall help to understand the security of the here-proposed signature scheme in practice.

Acknowledgements: We thank the anonymous referees for their comments, and Mukul Kulkarni for responding to our questions about the security proofs. We also thank Carlo Sanna, Andre Esser, Stefano Barbero and Emanuele Bellini for their valuable comments on the paper.

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., Katsumata, S., Pintore, F.: Calamari and Falafl: logarithmic (linkable) ring signatures from isogenies and lattices. In: Advances in cryptology—ASIACRYPT 2020. Part II, Lecture Notes in Comput. Sci., vol. 12492, pp. 464–492. Springer, Cham (2020)

7. 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)
8. Beullens, W.: Breaking rainbow takes a weekend on a laptop. IACR Cryptol. ePrint Arch. p. 214 (2022), https://eprint.iacr.org/2022/214
9. 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
10. 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
11. 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)
12. 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)
13. Di Scala, A.J., Sanna, C.: Smaller public keys for MinRank-based schemes. arXiv preprint (2023), https://arxiv.org/abs/2302.12447
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. Goubin, L., Courtois, N.T.: Cryptanalysis of the TTM Cryptosystem, pp. 44–57. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)
22. 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

23. 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)
24. 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)
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. 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
27. 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
28. 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)
29. 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)

A Proof of Theorem 2 (soundness)

Proof. We follow the soundness proof by Feneuil, Joux and Rivain in [17]. For simplicity, we assume that the commitment scheme is perfectly binding, since otherwise, if is was computationally binding, we would have to deal with cases of commitment collisions. For any set of successful transcripts corresponding to the same commitment, with at least two different challenges i^*:

– either the revealed shares of $[\![\boldsymbol{\alpha}]\!], [\![K]\!]$ are not consistent, and then a hash collision is found, since the commitment scheme is assumed perfectly binding;
– or the openings are unique, and then $([\![\boldsymbol{\alpha}]\!], [\![K]\!])$ is uniquely defined.

In the second case, this witness can be recovered from any two successful transcripts T_1 and T_2 corresponding to the same commitment and for which $i_1^* \neq i_2^*$. Let us call a witness $([\![\boldsymbol{\alpha}]\!], [\![K]\!])$ a *good witness* whenever $\boldsymbol{M}^L_{\boldsymbol{\alpha}} = \boldsymbol{M}^R_{\boldsymbol{\alpha}} K$, i.e., $\boldsymbol{\alpha}$ is a solution of the underlying MinRank problem.

Let $\tilde{\mathcal{P}}$, $\tilde{\varepsilon}$ and ε be as in Theorem 2. In figure Fig. 5, we describe an extractor $\mathcal{E}$ to find two valid transcripts T_1 and T_2 with a different second challenge. In what follows, we consider that $\mathcal{E}$ only receives transcripts with consistent shares since otherwise the extractor would find a hash collision.

Now, we want to estimate the number of calls $\mathcal{E}$ makes to $\tilde{\mathcal{P}}$ before returning (T_1, T_2) at step 7. We denote $\mathsf{succ}_{\tilde{\mathcal{P}}}$ the event that $\tilde{\mathcal{P}}$ succeeds in convincing a honest verifier V. By hypothesis, we have $\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}] = \tilde{\varepsilon} > \varepsilon$.

```
1. Repeat $+\infty$ times
2.   Run $\tilde{\mathcal{P}}$ with the honest verifier $\mathsf{V}$ to obtain a transcript $T_1$
3.   If $T_1$ is not a successful transcript, go to step 2
4.   Do $Z$ times:
5.     Run $\tilde{\mathcal{P}}$ with $\mathsf{V}$ to obtain a transcript $T_2$
6.     If $T_2$ is a successful transcript, $i_2^* \neq i_1^*$ and $(T_1, T_2)$ reveals a good witness,
7.       Return $(T_1, T_2)$
```

Fig. 5. Extractor $\mathcal{E}$.

Let $\alpha \in (0,1)$ be an arbitrary value such that $(1-\alpha)\tilde{\varepsilon} > \varepsilon$. Also, let X_h be the random variable that samples the randomness used by $\tilde{\mathcal{P}}$ in the generation of the initial commitment h. We say that an x_h in the sample space of X_h is *good* if

$$\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} \mid X_h = x_h] \geq (1-\alpha) \cdot \tilde{\varepsilon}.$$

By the Splitting Lemma [17, Lemma 5], we have for all realization x_h of X_h,

$$\Pr[x_h \text{ good } \mid \mathsf{succ}_{\tilde{\mathcal{P}}}] \geq \alpha.$$

Assume $\mathcal{E}$ obtains a successful transcript T_1 in Step 2 of Fig. 5, and let x_h be the underlying realization of X_h. Assume x_h is good. By definition, we have

$$\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} \mid X_h = x_h] \geq (1-\alpha) \cdot \tilde{\varepsilon} > \varepsilon > \frac{1}{N},$$

implying that there must exist a successful transcript T_2 with $i_2^* \neq i_1^*$. As explained above, this implies that there exists a unique and well-defined witness corresponding to these transcripts. Let $(\llbracket \boldsymbol{\alpha} \rrbracket, \llbracket K \rrbracket)$ be that witness. Now, we show that $(\llbracket \boldsymbol{\alpha} \rrbracket, \llbracket K \rrbracket)$ is a good witness. Assume $(\llbracket \boldsymbol{\alpha} \rrbracket, \llbracket K \rrbracket)$ is bad (i.e., $\boldsymbol{M}_{\boldsymbol{\alpha}}^L \neq \boldsymbol{M}_{\boldsymbol{\alpha}}^R K$.). By contradiction, we will show that then we have $\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} | X_h = x_h] \leq \varepsilon$, meaning that x_h is not good.

Denote FP the event that a genuine execution of the MPC protocol outputs a false positive, i.e. a zero matrix V. Then from Proposition 2, we have $\Pr[\mathsf{FP}] \leq \frac{1}{q^n}$. We now upper bound the probability that the inner loop of Fig. 5 succeeds:

$$\begin{aligned}\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} \mid X_h = x_h] &= \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}, \mathsf{FP} \mid X_h = x_h] + \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}, \overline{\mathsf{FP}} \mid X_h = x_h]. \\ &\leq \frac{1}{q^n} + (1 - \frac{1}{q^n}) \cdot \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} \mid X_h = x_h, \overline{\mathsf{FP}}].\end{aligned}$$

Having a successful transcript means that the sharing $\llbracket V \rrbracket$ in the first response of the prover must encode the zero matrix. But, the event $\overline{\mathsf{FP}}$, when we have a bad witness, implies that a genuine execution outputs a non-zero matrix V. So, to have a successful transcript, the prover must cheat for the simulation of at least one party. If the prover cheats for several parties, there is no way it can produce a successful transcript, while if the prover cheats for exactly one party (among the N parties), the probability to be successful is at most $1/N$. Thus, $\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} | X_h = x_h, \overline{\mathsf{FP}}] \leq 1/N$ and we have

$$\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}} | X_h = x_h] \leq \frac{1}{q^n} + (1 - \frac{1}{q^n}) \cdot \frac{1}{N} = \varepsilon,$$

meaning that x_h is not good. Thus, if x_h is good, then $(\llbracket \boldsymbol{\alpha} \rrbracket, \llbracket K \rrbracket)$ is good.

Now, we lower bound the probability that the i-th iteration of the inner loop of Fig. 5 finds a successful transcript T_2 with $i^*_{T_1} \neq i^*_{T_2}$ with a good x_h. We have

$$\begin{aligned}
&\Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}^{T_2} \cap (i^*_{T_1} \neq i^*_{T_2}) \mid x_h \text{ good}\,] \\
&\qquad = \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}^{T_2} \mid x_h \text{ good}\,] - \Pr[\mathsf{succ}_{\tilde{\mathcal{P}}}^{T_2} \cap (i^*_{T_1} = i^*_{T_2}) \mid x_h \text{ good}\,] \\
&\qquad \geq (1-\alpha)\tilde{\varepsilon} - \Pr[i^*_{T_1} = i^*_{T_2} \mid x_h \text{ good}\,] \geq (1-\alpha)\tilde{\varepsilon} - \Pr[i^*_{T_1} = i^*_{T_2}] \\
&\qquad = (1-\alpha)\tilde{\varepsilon} - 1/N \geq (1-\alpha)\tilde{\varepsilon} - \varepsilon.
\end{aligned}$$

Define $p_0 := (1-\alpha)\cdot\tilde{\varepsilon} - \varepsilon$. By running $\tilde{\mathcal{P}}$ with the same x_h as for the good transcript Z times, we hence obtain a second non-colliding transcript T_2 with probability at least $1/2$ when

$$Z \approx \frac{\ln(2)}{\ln\left(\frac{1}{1-p_0}\right)} \leq \frac{\ln(2)}{p_0}.$$

Now, we upper bounded the average number of calls of $\mathcal{E}$ to $\tilde{\mathcal{P}}$ before finishing.

1. $\mathcal{E}$ makes an average number of calls $1/\tilde{\varepsilon}$ to obtain T_1
2. Then $\mathcal{E}$ makes at most Z calls to $\tilde{\mathcal{P}}$ using the same x_h as for T_1 to obtain a successful transcript T_2 such that $i^*_{T_1} \neq i^*_{T_2}$. The probability that such a T_2 is found is at least $\alpha/2$, since the probability that x_h is good is at least α, and whenever x_h is good the probability of finding T_2 is at least $1/2$.

Hence, the average number of calls of the extractor $\mathcal{E}$ to $\tilde{\mathcal{P}}$ is upper bounded by

$$\left(\frac{1}{\tilde{\varepsilon}} + Z\right)\cdot\frac{2}{\alpha} = \left(\frac{1}{\tilde{\varepsilon}} + \frac{\ln(2)}{(1-\alpha)\cdot\tilde{\varepsilon} - \varepsilon}\right)\cdot\frac{2}{\alpha}$$

To obtain an α-free formula, we take α such that $(1-\alpha)\cdot\tilde{\varepsilon} = \frac{1}{2}(\tilde{\varepsilon}+\varepsilon)$, implying $\alpha = \frac{1}{2}(1-\frac{\varepsilon}{\tilde{\varepsilon}})$. Hence, the average number of calls to $\tilde{\mathcal{P}}$ is at most

$$\frac{4}{\tilde{\varepsilon}-\varepsilon}\left(1 + \tilde{\varepsilon}\cdot\frac{2\cdot\ln(2)}{\tilde{\varepsilon}-\varepsilon}\right).$$

B Proof of Theorem 3 (zero-knowledge)

Proof. As in the proof of the soundness, we follow the approach by Feneuil, Joux and Rivain in [17]. First, we describe in Fig. 6 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. 7, 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^*).

```
1. $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda, \quad (\mathsf{seed}_i, \rho_i)_{i\in[N]} \leftarrow \mathrm{TreePRG}(\mathsf{seed})$
2. For each party $i \in [N]\backslash\{i^*\}$,
   – $[\![A]\!]_i, [\![B]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$
   – If $i \neq N$,
      • $[\![\boldsymbol{\alpha}]\!]_i, [\![C]\!]_i, [\![K]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i), \quad \mathsf{state}_i = \mathsf{seed}_i$
   – Else,
      • $[\![\boldsymbol{\alpha}]\!]_N \xleftarrow{\$} \mathbb{F}_q^k, \quad [\![K]\!]_N \xleftarrow{\$} \mathbb{F}_q^{r\times(n-r)}, \quad [\![C]\!]_N \xleftarrow{\$} \mathbb{F}_q^{n\times(n-r)}$
      • $\mathsf{aux} = ([\![\boldsymbol{\alpha}]\!]_N, [\![K]\!]_N, [\![C]\!]_N), \quad \mathsf{state}_N = \mathsf{seed}_N \parallel \mathsf{aux}$
3. For party $i^*$,
   – $\mathsf{com}_{i^*} \leftarrow \mathsf{Com}(\mathsf{state}_{i^*}, \rho_{i^*})$
   – $[\![S_1]\!]_{i^*} \xleftarrow{\$} \mathbb{F}_q^{n\times r}, \quad [\![S_2]\!]_{i^*} \xleftarrow{\$} \mathbb{F}_q^{r\times n-r}, \quad [\![V]\!]_{i^*} = -\sum_{i\neq i^*} [\![V]\!]_i$
4. Run Phase 3 of Fig. 2 on each party $i \in [N]\backslash\{i^*\}$ to obtain $\{[\![S_1]\!]_i, [\![S_2]\!]_i, [\![V]\!]_i\}$.
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. 6. Internal HVZK simulator $\mathcal{S}$ on input of challenges (R, i^*).

```
1. $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda, \quad (\mathsf{seed}_i, \rho_i)_{i\in[N]} \leftarrow \mathrm{TreePRG}(\mathsf{seed})$
2. For each party $i \in [N]$,
   – $[\![A]\!]_i, [\![B]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i)$
   – If $i \neq N$,
      • $[\![\boldsymbol{\alpha}]\!]_i, [\![C]\!]_i, [\![K]\!]_i \leftarrow \mathrm{PRG}(\mathsf{seed}_i), \quad \mathsf{state}_i = \mathsf{seed}_i$
   – Else,
      • $[\![\boldsymbol{\alpha}]\!]_N \leftarrow \boldsymbol{\alpha} - \sum_{i\neq N} [\![\boldsymbol{\alpha}]\!]_i, \ [\![K]\!]_N \leftarrow K - \sum_{i\neq N} [\![K]\!]_i, \ [\![C]\!]_N \leftarrow A\cdot B - \sum_{i\neq N} [\![C]\!]_i$
      • $\mathsf{aux} = ([\![\boldsymbol{\alpha}]\!]_N, [\![K]\!]_N, [\![C]\!]_N), \quad \mathsf{state}_N = \mathsf{seed}_N \parallel \mathsf{aux}$
3. Run Phase 3 of Fig. 2 on all the parties to obtain $\{[\![S_1]\!]_i, [\![S_2]\!]_i, [\![V]\!]_i\}_{i\in[N]}$.
4. 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. 7. Simulator 0 on input of 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 $\varepsilon_{\mathrm{PRG}}$.

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.

Simulator 3 (Internal HVZK simulator). The only difference between Simulator 2 and the internal HVZK simulator $\mathcal{S}$ in Fig. 6 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. 8:

1. $R \xleftarrow{\$} \mathcal{E}_f, \quad i^* \xleftarrow{\$} [N]$ (as an honest verifier).
2. Run the simulator $\mathcal{S}(R, i^*)$ to obtain
   - $\mathsf{rsp}_1 = \mathrm{Hash}\left([\![S_1]\!]_1, [\![S_2]\!]_1, [\![V]\!]_1, \ldots, [\![S_1]\!]_N, [\![S_2]\!]_N, [\![V]\!]_N\right)$.
   - $\mathsf{rsp}_2 = \left((\mathsf{state}_i, \rho_i)_{i\neq i^*}, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}\right)$
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 $h = \mathrm{Hash}(\mathsf{com}_1, \ldots, \mathsf{com}_N)$
   - Update $\mathsf{rsp}_2 = \left((\mathsf{state}_i, \rho_i)_{i\neq i^*}, \mathsf{com}_{i^*}, [\![S_1]\!]_{i^*}, [\![S_2]\!]_{i^*}\right)$
4. Output the transcript $T = (h, R, \mathsf{rsp}_1, i^*, \mathsf{rsp}_2)$

Fig. 8. The global HVZK simulator.

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. □

C Proof of Theorem 4 (EUF-CMA)

Proof. Let $\mathcal{A}$ be an adversary making q_s signing queries, and q_0, q_1, q_2 queries to Hash_0, Hash_1 and Hash_2, respectively. 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

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

then there necessarily exists, at a given moment during the experiment, a query to Hash_2 made by $\mathcal{A}$ itself with output h_2 input, and an input of the form

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

Experiment 1. This is 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. 4 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. 6 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. $\mathsf{salt} \xleftarrow{\$} \{0,1\}^{2\lambda}$.

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

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

1. $(\mathsf{state}_i^{[\ell]})_{i \neq i^{*,[\ell]}}, \left(\left[\!\left[S_1^{[\ell]} \right]\!\right]_i, \left[\!\left[S_2^{[\ell]} \right]\!\right]_i, \left[\!\left[V_1^{[\ell]} \right]\!\right]_i \right)_{i \in [N]} \leftarrow \mathcal{S}_{\mathsf{salt}}(R^{[\ell]}, i^{*,[\ell]})$.
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

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

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. At any point during this experiment, we say that we have a $\mathsf{correct}$ execution ℓ^* if, in a query to Hash_2 with input

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

1. there is a previous query $h_1 \leftarrow \mathrm{Hash}_1 \left(\mathsf{msg}, \mathsf{salt}, \mathsf{com}_1^{[1]}, \ldots, \mathsf{com}_N^{[\tau]}\right)$,
2. and each $\mathsf{com}_i^{[\ell^*]}$ was output by a previous query (by either $\mathcal{A}$ or the signing oracle) to Hash_0 with input $\left(\mathsf{salt}, \ell, i, \mathsf{state}_i^{[\ell^*]}\right)$,
3. and a good witness $(\boldsymbol{\alpha}, K)$ can be extracted from $\{\mathsf{state}_i^{[\ell^*]}\}_{i \in [N]}$.

In this experiment, it is checked in each query made by $\mathcal{A}$ to Hash_2 (where msg was not previously queried) 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, $\Pr_6[\mathsf{Solve}] \leq \varepsilon_{\mathrm{MR}}$. Hence,

$$\begin{aligned}\Pr_6[\mathsf{Forge}] &= \Pr_6[\mathsf{Forge} \text{ and } \mathsf{Solve}] + \Pr_6[\mathsf{Forge} \text{ and not } \mathsf{Solve}] \\ &\leq \varepsilon_{\mathrm{MR}} + \Pr_6[\mathsf{Forge} \text{ and not } \mathsf{Solve}].\end{aligned}$$

Now, suppose we have a forgery (msg, σ) and Solve does not occur. Then for every $\ell \in [\tau]$, exactly one of the three following cases must occur:

a) $\mathsf{com}^{[\ell]}_{i^*,\ell}$ was not output by Hash_0.

b)
- for all $i \in [N]$, $\mathsf{com}^{[\ell]}_i$ was output by a query to Hash_0 with input $\big(\mathsf{salt}, \ell, i, \mathsf{state}^{[\ell]}_i\big)$,
- $\left[\!\left[S^{[\ell]}_2\right]\!\right]_{i^*}$ is obtained from $\mathsf{state}^{[\ell]}_{i^*}$, $\left[\!\left[S^{[\ell]}_1\right]\!\right]_{i^*}$ is obtained from $\mathsf{state}^{[\ell]}_{i^*}$ and h_1, and $\sum_{i \neq i^*,[\ell]} \left[\!\left[V^{[\ell]}\right]\!\right]_i$ is obtained from $\mathsf{state}^{[\ell]}_{i^*}$, h_1, $S^{[\ell]}_1$, $S^{[\ell]}_2$.
- the witness $(\boldsymbol{\alpha}, K)$ extracted from $\{\mathsf{state}^{[\ell]}_i\}_{i\in[N]}$ is a bad witness.

b')
- for all $i \in [N]$, $\mathsf{com}^{[\ell]}_i$ was output by a query to Hash_0 with input $\big(\mathsf{salt}, \ell, i, \mathsf{state}^{[\ell]}_i\big)$,
- $\left[\!\left[S^{[\ell]}_2\right]\!\right]_{i^*}$ is not obtained from $\mathsf{state}^{[\ell]}_{i^*}$, or $\left[\!\left[S^{[\ell]}_1\right]\!\right]_{i^*}$ is not from $\mathsf{state}^{[\ell]}_{i^*}$ and h_1, or $\sum_{i \neq i^*,[\ell]} \left[\!\left[V^{[\ell]}\right]\!\right]_i$ is not from $\mathsf{state}^{[\ell]}_{i^*}$, h_1, $S^{[\ell]}_1$, $S^{[\ell]}_2$.
- the witness $(\boldsymbol{\alpha}, K)$ extracted from $\{\mathsf{state}^{[\ell]}_i\}_{i\in[N]}$ is a bad witness.

Clearly, if b) occurs for a round $\ell \in [\tau]$, this means that the MPC protocol in Section 4 to verify matrix-multiplication triple is honestly followed by every party $i \in [N]$. Hence, from Proposition 2, we have that the adversary had probability $1/q^n$ to have b) satisfied for this round ℓ. This probability is in fact given by obtaining from h_1 one precise first challenge $R^{[\ell]}$ out of the q^n possibilities. Therefore, the probability of having exactly $s \in [\tau]$ rounds satisfying b) is at most

$$P_b(s) = \left(\frac{1}{q^n}\right)^s \left(1 - \frac{1}{q^n}\right)^{\tau - s} \binom{\tau}{s}.$$

If b) does not occur for a round $\ell \in [\tau]$, this clearly means that any other second challenge obtained from h_2 different from $i^{*,\ell}$ would make the forgery fail. Hence, the probability of having this round ℓ not leading to rejection is at most $1/N$. Therefore, the probability of having exactly $\tau - s \in [\tau]$ rounds satisfying a) or b') is at most

$$P_{a,b'}(s) = \frac{1}{N^{\tau - s}}.$$

In view of the above, the probability of having Forge and not Solve with exactly s rounds satisfying b) after q_1 queries to Hash_1 and q_2 queries to Hash_2 is at most

$$P(s) = \Big(1 - \big(1 - P_b(s)\big)^{q_1}\Big)\Big(1 - \big(1 - P_{a,b'}(s)\big)^{q_2}\Big).$$

Thus, we have

$$\Pr_6[\mathsf{Forge} \text{ and not } \mathsf{Solve}] \leq \max_{0 \leq s \leq \tau} P(s).$$