Sneaking up the ranks: Partial key exposure attacks on rank-based schemes

Designs, Codes and Cryptography (2026) 94:15
https://doi.org/10.1007/s10623-025-01738-1

Sneaking up the ranks: Partial key exposure attacks on rank-based schemes

Giuseppe D'Alconzo[1] · Andre Esser[2] · Andrea Gangemi[1] · Carlo Sanna[1]

Received: 24 December 2024 / Revised: 24 December 2024 / Accepted: 26 November 2025

Abstract
A partial key exposure attack is a key recovery attack where an adversary obtains a priori partial knowledge of the secret key, e.g., through side-channel leakage. While for a long time post-quantum cryptosystems, unlike RSA, have been believed to be resistant to such attacks, recent results by Esser, May, Verbel, and Wen (CRYPTO '22), and by Kirshanova and May (SCN '22), have refuted this belief. In this work, we focus on partial key exposure attacks in the context of rank-metric-based schemes, particularly targeting the RYDE, MIRA, and MiRitH digital signatures schemes, which are active candidates in the NIST post-quantum cryptography standardization process. We demonstrate that, similar to the RSA case, the secret key in RYDE can be recovered from a constant fraction of its bits. Specifically, for NIST category I parameters, our attacks remain efficient even when less than 25% of the key material is leaked. Interestingly, our attacks lead to a natural improvement of the best generic attack on RYDE *without partial knowledge*, reducing security levels by up to 9 bits. For MIRA and MiRitH our attacks remain efficient as long as roughly 57–60% of the secret key material is leaked. Additionally, we initiate the study of partial exposure of the witness in constructions following the popular MPCitH (MPC-in-the-Head) paradigm. We show a generic reduction from recovering RYDE and MIRA's witness to the MinRank problem, which again leads to efficient key recovery from constant fractions of the secret witness in both cases.

Keywords Erasure/error model · MinRank · Rank syndrome decoding · Post-quantum

Communicated by D. Jungnickel.

✉ Giuseppe D'Alconzo
giuseppe.dalconzo@polito.it

Andre Esser
andre.esser@tii.ae

Andrea Gangemi
andrea.gangemi@polito.it

Carlo Sanna
carlo.sanna@polito.it

[1] Department of Mathematical Sciences, Polytechnic of Turin, Turin, Italy

[2] Technology Innovation Institute, Abu Dhabi, UAE

1 Introduction

Cryptographic systems are typically designed to guarantee security as long as the secret key remains hidden, while security trivially breaks if the secret key is revealed. However, the security of the system under *partial* exposure of the key material, referred to as *leakage robustness* or resilience, is usually not guaranteed by design. A prominent example is the RSA cryptosystem, which has been proven vulnerable to *partial key exposure attacks*: A well-known result by Coppersmith [10] shows that half of the bits of a prime factor p are sufficient to factor the public modulus $N = pq$ in polynomial time. Similar results have later been shown under partial exposure of the private exponent as well as for the private CRT exponents [14, 25, 26, 32].

In contrast, for most other schemes and hardness assumptions, such as the discrete logarithm problem, it is widely believed that a partial exposure of the secret does not allow for polynomial-time recovery [20, 30]. Similar beliefs hold up in the case of post-quantum cryptography, where certain schemes have been proven to enjoy a specific form of leakage robustness [11]. Furthermore, previous works on partial key exposure attacks on different PQC schemes, including McEliece [34], NTRU [31], BLISS [33] and LUOV [35], all seemingly support the assumption that the best attack under key leakage is based on enumeration strategies of the missing key material. As such, those attacks are usually inferior to generic key recovery attacks, unless most of the key material is leaked.

Only recently Esser, May, Verbel, and Wen [16] have shown that in the case of the Rainbow [12], NTRU [9] and BIKE [5] cryptosystems the structure as well as redundancy in the secret keys can be exploited to mount more efficient partial key exposure attacks. For example, in the case of BIKE it is shown that knowledge of half of the secret-key bits allows one to recover the entire secret key. Later Kirshanova and May [24] showed that in the case of the McEliece system [27, 28] already a quarter of the secret key is sufficient for full-key recovery. To the best of our knowledge, no other post-quantum scheme has yet been shown to be vulnerable to partial key exposure attacks.

In this work, we analyze the leakage resilience of rank-based schemes and in particular apply our findings to RYDE [6], MiRitH [1] and MIRA [7], three active candidates of the NIST standardization process for post-quantum secure digital signatures. More precisely, RYDE is based on the rank-metric syndrome decoding (Rank-SD) problem, while both MIRA and MiRitH are based on the MinRank problem. Note that all three schemes have been selected by NIST to move forward to the second round of evaluation [3], with MiRitH and MIRA being merged into a single submission named Mirath [2]. Furthermore, these schemes are constructed following the MPCitH (MPC-in-the-Head) paradigm [23] from an authentication protocol, which is transformed into a non-interactive signature scheme via the Fiat–Shamir transform [18]. Within the authentication protocol, the prover proves knowledge of a witness to the verifier, which guarantees that the prover knows a solution to the given problem. In RYDE as well as MIRA this witness is a specific polynomial constructed from the solution. As knowledge of this polynomial allows for polynomial-time key recovery in both cases, the witness can be seen as part of the secret key.

In the case of Rank-SD, we show that already a constant fraction of the secret allows for polynomial-time recovery. This translates to polynomial-time attacks on RYDE parameters if about 40% of the key material is leaked. Furthermore, we show that our partial-key exposure attack leads to a natural improvement of the best generic Rank-SD attack against RYDE, the GRS attack [4, 19], reducing RYDE security levels *without any key leakage* by up to 9 bits.

In the case of MinRank, we show how to leverage bit knowledge on the secret key to obtain a reduced instance, which can be solved more efficiently. While this attack does not stay polynomial, it enables secret key recovery for MiRitH and MIRA whenever about 55–60% of the key material is known using less than 2^{80} operations.

We then study a partial exposure of the polynomial used as a witness in the authentication protocol of RYDE and MIRA. We show that this attack asymptotically improves significantly over a bruteforce of the missing key material, which is also reflected in its concrete complexity: A recovery of the hidden polynomial remains feasible from up to 69% (RYDE) and 76% (MIRA) of its coefficients in less than 2^{80} operations.

All our attacks apply whenever a given fraction of the bit representation of the secret key is leaked and, hence, are applicable to any distribution of leaked key material. Note that this is in strong contrast to most previous works on partial key exposure attacks. In the RSA setting key material is usually required to be leaked consecutively, while often even an exposure of the most or least significant bits is required [10]. Henninger and Shacham [22] then required erasures to appear in random positions. In [16] the attacks, with the exception of the attacks on BIKE, either require leakage of full $\mathbb{F}_q$ coefficients of the secret key or require erasures to appear again randomly when moving to leaked bit information. Similarly, in [24] leakage of elements in $\mathbb{F}_{2^m}$ is assumed.

Our results emphasize the need to counteract key leakage attacks already in the early stages of the design by embedding respective countermeasures, such as reducing key redundancy. Overall the awareness for partial key exposure attacks seems still limited, as known key-compression techniques that would reduce the impact of our attacks do not find application in the current specification of RYDE. The issue becomes even more pressing considering the growing trend towards MPCitH and VOLEitH (VOLE-in-the-Head [8]) -based constructions, which introduce additional secret material in the form of the witness. We show, for the first time, that partial exposure of this witness can equally be exploited for key recovery.

1.1 Our contributions

As a first small contribution, we give a general definition of partial key exposure attacks, which summarizes many of the different settings encountered in the literature. We focus on a setting commonly referred to as *erasure setting*, in which arbitrary, but known positions of the secret key are revealed. We then also provide generic translations of our attacks to the *error setting* in which a complete, but erroneous version of the secret key is leaked. In the following, we give a more precise summary of our technical contributions.

Rank-SD and application to RYDE The Rank-SD problem is given a parity-check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$, a syndrome $\mathbf{s} \in \mathbb{F}_{q^m}^{n-k}$ and a target rank $r < m$, find an $\mathbf{x} \in \mathbb{F}_{q^m}^{n}$ with $\mathbf{Hx} = \mathbf{s}$ where the entries of $\mathbf{x}$ generate an r-dimensional subspace $E \subset \mathbb{F}_{q^m}$. Our starting point for the partial key exposure attack is a generic attack on the Rank-SD problem, the GRS attack. On a high level, the attack guesses subspaces $F \subset \mathbb{F}_{q^m}$ of dimension $r' > r$ with the goal of finding a subspace F with $E \subseteq F$. Such a subspace then allows for recovery of E and eventually the solution $\mathbf{x}$ if r' is not much larger than r. In this context, we show that any known bit of the solution allows us to derive linear equations in the unknown generators of E. In turn, this allows us to choose subspaces F of larger dimension r' while still allowing to recover E, once $E \subseteq F$. Since the probability of $E \subseteq F$ grows with the dimension r' this overall reduces the attack complexity.

Table 1 RYDE security levels without partial knowledge, and maximum erasure rates p that makes possible a key recovery in polynomial-time or using less than 2^{80} operations

	Bit security		Erasure rate p	
	[6]	This work	Polynomial	80-Bit
RYDE-I	147	138	0.61	0.78
RYDE-III	216	210	0.59	0.70
RYDE-V	283	283	0.64	0.71

Table 2 Maximum erasure rate that allows for a key recovery with less than 2^{80} operations for MIRA and MiRitH "a" parameter sets

	MIRA	MiRitH
NIST-I	0.40	0.43
NIST-III	0.24	0.27
NIST-V	0.18	0.22

When applied to RYDE parameters this leads to polynomial-time key recovery attacks from 36 to 41% of leaked key material. Furthermore, when allowing for a reasonably high practical threshold of 2^{80} operations, key recovery remains feasible from less than a quarter of leaked key material. We summarize these results in Table 1.

Using the incorporation of additional linear equations, we then derive a natural improvement of the generic GRS attack *without key leakage*. More precisely, we show that for certain parameter regimes it is beneficial to first guess a certain amount of the secret key bits and then apply the GRS attack in the erasure case. Even though this procedure has to be repeated for every possible guess of the secret key bits, the improvement still compensates for the guessing overhead. In particular, this reduces the security level of the RYDE-I and RYDE-III parameter sets by 9 and 6 bits respectively.

MinRank and application to MiRitH and MIRA Informally, the MinRank problem asks to find a low-rank linear combination $\mathbf{E} = \mathbf{M}_0 + \sum_i \alpha_i \mathbf{M}_i$ of a given set of input matrices $\mathbf{M}_0, (\mathbf{M}_i)_{i\in\{1,\dots,k\}}$. The solution to the problem are the coefficients of the linear combination, i.e., the α_i. Knowledge of coefficients α_i allows one to reduce the input instance: Given α_i discard the matrix $\mathbf{M}_i$ and consider as the new $\mathbf{M}_0$ the matrix $\mathbf{M}_0 + \alpha_i \mathbf{M}_i$, yielding an instance with same parameters but with new $k' = k - 1$. However, since MiRitH as well as MIRA use coefficients $\alpha_i \in \mathbb{F}_{16}$, bit knowledge on the secret key does not translate directly into knowledge of the coefficients α_i.

To overcome this, we show a generic reduction from any MinRank instance over an extension field $\mathbb{F}_{p^r}$ to an instance over the basefield $\mathbb{F}_p$. Then, exploiting the fact that MIRA as well as MiRitH use instances over the extension field $\mathbb{F}_{16}$, we show that bit knowledge on the secret key corresponds to known coefficients of the MinRank instance over the basefield $\mathbb{F}_2$. This leads to a reduced instance over the basefield which can be solved more efficiently. For MiRitH and MIRA parameters this approach leads to secret key recovery from 57–78% and 60–82% of leaked material respectively, when allowing for 2^{80} bit operations (see Table 2).

q-polynomials and application to RYDE and MIRA Note that similar to the Rank-SD problem, the rows of the low-rank matrix $\mathbf{E}$ related to a MinRank instance generate a low-dimensional subspace E. In RYDE as well as MIRA the subspace E is used to construct a specific kind of polynomial, called q-polynomial, used to prove knowledge of E, which in both cases is equivalent to knowledge of the solution.

Table 3 Asymptotic complexity exponent c and maximum erasure rate p that allows for recovery of the witness in less than 2^{80} operations

Parameter set	RYDE		MIRA	
	c	80-Bit	c	80-Bit
NIST-I	0.68	0.31	0.69	0.26
NIST-III	0.65	0.19	0.69	0.26
NIST-V	0.60	0.14	0.74	0.11

With regards to partial exposure of the polynomial, we show a generic reduction from recovering missing bits of the coefficients of any q-polynomial, with $q = 2^v$ for any integer v, to the MinRank problem. While it might appear direct as the polynomial was initially constructed from a MinRank instance (in case of MIRA), note that the instance parameters obtained in the reduction are different from the original input instance. Furthermore, a recovery of the original instance from the polynomial is non-trivial as recovering the original Rank-SD instance (in case of RYDE) would imply a reduction from MinRank to Rank-SD, while finding such a reduction is a longstanding open problem.

We then apply known algorithms to solve the MinRank instance obtained in the reduction, to eventually recover the q-polynomial from the leaked information. We show that this approach significantly improves on a brute force of the missing key material: While a brute force of the missing k bits takes about 2^k operations, a recovery via the introduced reduction runs in time $2^{c \cdot k}$ for a constant $0.6 \leq c \leq 0.74$ depending on the schemes parameters. While this attack does not stay polynomial, we find that recovery remains possible within 2^{80} operations up to 31% of erased key material. We summarize those results in Table 3.

Artifacts We provide all source code used to compute concrete estimates as well as erasure and error bounds in the GitHub repository https://github.com/gdalc/Sneaking-up-the-Ranks.

Open Question The attacks presented in this work lead to full secret key recovery from fractions of exposed key *or* witness material. In the specific setting where side channel information on both of these secrets is available, an algorithm that is able to exploit information on the witness as well as the secret key could potentially lead to higher tolerable erasure rates. We pose it as on open question to design such an algorithm or incorporate the respective information in the algorithms presented here.

Outline In Sect. 2 we give our general definition of partial key exposure attacks and recall necessary basics on the rank metric, as well as on the RYDE, MIRA, and MiRitH cryptosystems. In Sect. 3 we present the partial key exposure attack on the Rank-SD problem, its application to RYDE and the resulting improvement of the GRS attack. Additionally, we provide a translation of the attack to a different leakage setting, the *error* setting. In the following Sect. 4 we cover results on the partial key exposure attacks on MinRank. This includes the reduction from MinRank instances over extension fields to the base field as well as a study of the complexity of solving the resulting instance. Subsequently, in Sect. 5 we provide the generic reduction from recovering missing bits of a q-polynomial to an instance of the MinRank problem. Eventually, we study the complexity of solving the derived instance and derive concrete bounds for the recovery under certain runtime constraints.

2 Preliminaries

We let $\mathbb{F}_q$ denote the finite field with q elements. We let $\mathbb{F}_q^{m\times n}$ denote the vector space of $m \times n$ matrices over $\mathbb{F}_q$ and let $\mathbb{F}_q^n := \mathbb{F}_q^{n\times 1}$, therefore by default vectors are column vectors. We write $\mathbf{x}^\intercal$ for the transpose of a vector $\mathbf{x}$. We denote the Hamming weight of a vector $\mathbf{x}$, which is the number of its non-zero entries, by $W_{\mathrm{H}}(\mathbf{x})$. With $\langle \mathbf{x}_1, \ldots, \mathbf{x}_k \rangle$, we denote the vector space generated by $\mathbf{x}_1, \ldots, \mathbf{x}_k$ in $\mathbb{F}_q$. The Gaussian binomial coefficient is defined as $\begin{bmatrix} m \\ r \end{bmatrix}_q = \prod_{i=1}^{r} \frac{q^{m-i+1}-1}{q^i-1}$ and can be approximated by $q^{r(m-r)}$, as given by the following lemma.

Lemma 1 *For every prime power q there exists a constant $C_q \in (0, 1)$ such that*

$$C_q < \frac{1}{q^{r(m-r)}} \begin{bmatrix} m \\ r \end{bmatrix}_q < C_q^{-1}.$$

Proof See Appendix D. □

We let $\Pr[A]$ denote the probability of event A. We define $[n] := \{1, \ldots, n\}$ for every positive integer n. All logarithms are computed in base 2. In algorithmic descriptions, "$\leftarrow$" denotes a variable assignment. We express complexity statements using standard Landau notation $\mathcal{O}$ and Ω. By w we refer to the linear algebra constant. If not stated otherwise we consider $w = 3$.

2.1 Partial key exposure attacks

Partial key exposure attacks refer to key recovery attacks, where certain information about the secret key is known a priori. These kinds of attacks are investigated in many different settings. However, all these common settings can be summarized in the following general definition.

Definition 1 (Partial Key Exposure) Let $\mathbf{s} \in \mathbb{F}_2^n$ be a secret key and let $\mathbf{e}$ be drawn at random from $\mathbb{F}_2^n$ according to a known probability distribution D. Then a partial key exposure attack asks to recover $\mathbf{s}$ given $\tilde{\mathbf{s}} = \mathbf{s} + \mathbf{e}$ and the public parameters of the scheme.

The different settings only distinguish in the choice of the distribution D. The most prevalent setting is commonly referred to as *erasure setting* [16, 22, 24]. In this setting, the error vector $\mathbf{e}$ is distributed as follows. Let $I \subseteq [n]$ be a known fixed set. Then every coordinate of $\mathbf{e}$ is independently distributed according to distribution D_I, where for $e_i \sim D_I$ it holds $\Pr[e_i = 0 \mid i \in I] = 1/2$ and $\Pr[e_i = 0 \mid i \notin I] = 1$.

Put differently, the secret-key bits indexed by I (from *incognito*) are unknown or erased while the remaining are known. We summarize the erasure setting in the following more specific definition.

Definition 2 (Partial Key Exposure—Erasure) Let $\mathbf{s} \in \mathbb{F}_2^n$ be a secret key, and $I \subseteq [n]$ be a set of indices of erased bits. Then a partial key exposure attack *in the erasure setting* asks to recover $\mathbf{s}$ given s_i for $i \in [n] \setminus I$. We refer to $\delta := |I|/n$ as the *erasure rate*.

While our main focus lies on the erasure setting we also translate our attacks to a setting referred to as *error setting*. In this case the distribution D from Definition 1 is defined such that for $\mathbf{e} \sim D$ the number of nonzero coordinates of $\mathbf{e}$ is bounded (or bounded on average)

by a constant k. A common instantiation of D in this case is the distribution where every coordinate e_i of $\mathbf{e}$ is independently Bernoulli distributed with parameter δ. That is, the events $(e_i = 1)_{i\in[n]}$ are independent and each of them happens with probability δ. In other words, an erroneous version of the secret key $\mathbf{s}$ is known, where each bit was flipped with probability δ. This translates into the following definition.

Definition 3 (Partial Key Exposure—Error) Let $\mathbf{s} \in \mathbb{F}_2^n$ be a secret key, and let $k \leq n$ be a positive integer. Furthermore, let $\tilde{\mathbf{s}} \in \mathbb{F}_2^n$ be such that $|\{i \in [n] : \tilde{s}_i \neq s_i\}| \leq k$. Then a partial key exposure attack *in the error setting* asks to recover $\mathbf{s}$ given $\tilde{\mathbf{s}}$. We refer to $\delta := k/n$ as the *error rate*.

2.2 Rank-metric decoding

Recall that a *linear code* $\mathcal{C}$ of *dimension* k and *length* n over a finite field $\mathbb{F}$ is defined as a linear subspace of $\mathbb{F}^n$ of dimension k. We call $\mathcal{C}$ a $[n, k]_{\mathbb{F}}$*-code*, and call the elements of $\mathcal{C}$ *codewords*. Such a code can be described either via a *generator matrix* $\mathbf{G} \in \mathbb{F}^{k\times n}$ such that $\mathcal{C} = \{\mathbf{m}^\intercal\mathbf{G} : \mathbf{m} \in \mathbb{F}^k\}$, or via a *parity-check matrix* $\mathbf{H} \in \mathbb{F}^{(n-k)\times n}$ such that $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}^n : \mathbf{Hc} = \mathbf{0}\}$.

The rank metric Let $\mathcal{C}$ be an $[n, k]_{\mathbb{F}_{q^m}}$-code over the extension field $\mathbb{F}_{q^m}$. Let $b_1, \ldots, b_m$ be a basis of $\mathbb{F}_{q^m}$ as a linear space over $\mathbb{F}_q$. If $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$, then for any coordinate x_i there is a (unique) representation $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m}) \in \mathbb{F}_q^m$ in the basis elements, that is, $x_i = \sum_{j=1}^m x_{i,j} b_j$. The *rank weight* $W_{\mathrm{R}}(\mathbf{x})$ of $\mathbf{x}$ is defined as the rank of the $n \times m$ matrix over $\mathbb{F}_q$ having rows $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Finally, the *support* $\mathsf{Supp}(\mathbf{x})$ of $\mathbf{x}$ is the linear subspace of $\mathbb{F}_{q^m}$ generated by $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Note that the rank weight as well as the support of a vector is independent of the choice of the basis.

Rank syndrome decoding The Rank Syndrome Decoding (Rank-SD) problem is defined as follows.

Definition 4 (Rank Syndrome Decoding) Let (q, m, n, k, r) be positive integers such that $k/n \in (0, 1)$ is a constant and $r \leq m$. Further let $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be a parity-check matrix of an $[n, k]_{\mathbb{F}_{q^m}}$-code and $\mathbf{s} \in \mathbb{F}_{q^m}^{n-k}$. The *Rank Syndrome Decoding Problem* asks, given $\mathbf{H}$ and the *syndrome* $\mathbf{s}$, to find $\mathbf{x} \in \mathbb{F}_{q^m}^n$ of rank-weight $W_{\mathrm{R}}(\mathbf{x}) = r$ satisfying $\mathbf{Hx} = \mathbf{s}$, if such an $\mathbf{x}$ exists.

Note that in the cryptographic context, a solution is usually guaranteed to exist by construction and forms the secret key. We therefore always assume the existence of at least one solution.

Algorithms to solve the Rank-SD problem broadly classify into two categories — combinatorial and algebraic attacks. In a nutshell, algebraic attacks model the rank restriction on $\mathbf{x}$ as a quadratic system, which is then solved via techniques optimized for solving these kinds of systems. In contrast, combinatorial attacks focus on identifying elements in the support of $\mathbf{x}$, which result in linear equations involving the solution. Once enough linear equations are obtained, the solution can be recovered by solving the linear system.

With respect to RYDE, the best known attacks are of combinatorial nature. More precisely, the most efficient attack is an improvement of the GRS algorithm [19], which has been proposed in [4] and forms the starting point for our partial key exposure attacks.

2.3 The RYDE signature scheme

RYDE is a digital signature scheme following the MPCitH paradigm and its security is based on the Rank-SD problem. Hence, the secret key in RYDE is the solution $\mathbf{x}$ to a Rank-SD instance.

MPC protocol and q-polynomials As an MPCitH-based scheme, RYDE is constructed from an authentication protocol using the Fiat–Shamir transform to obtain a non-interactive signature scheme. Within this authentication protocol, the prover proves knowledge of a witness to the verifier, which guarantees that the prover knows the solution $\mathbf{x}$. Most details of the specific MPC protocol used in RYDE are not relevant for our attack. However, the crucial part is that the witness in the case of RYDE is a *q-polynomial* constructed from the solution $\mathbf{x}$. Formally, a q-polynomial is defined as follows.

Definition 5 (q-polynomial) A *q-polynomial* of q-degree r is a polynomial in $\mathbb{F}_{q^m}[X]$ of the form

$$L(X) = X^{q^r} + \sum_{i=0}^{r-1} \beta_i X^{q^i}, \ \ \beta_i \in \mathbb{F}_{q^m}.$$

Notice that, from our definition, q-polynomials are always monic. As detailed in the following proposition, q-polynomials are strictly connected to linear subspaces of $\mathbb{F}_{q^m}$.

Proposition 2 (q-polynomials and linear subspaces, [29]) *Let E be a linear subspace of $\mathbb{F}_{q^m}$ of dimension $r \leq m$. Then, there exists a unique q-polynomial $L_E(X)$ of q-degree r such that all elements of E are roots of $L_E(X)$, that is,*

$$L_E(X) = \prod_{e \in E} (X - e).$$

Such a polynomial is called *annihilator polynomial* of E. In RYDE, the secret vector $\mathbf{x}$ has as support the linear subspace

$$U = \mathsf{Supp}(\mathbf{x}) = \langle 1, x_{i_1}, \ldots, x_{i_{r-1}} \rangle, \quad \{i_1, \ldots, i_{r-1}\} \subset [n]$$

of dimension r. Then, the witness is defined as the annihilator polynomial of U which is

$$L_U(X) := \prod_{u \in U} (X - u) = \sum_{i=0}^{r} \beta_i X^{q^i},$$

where $\beta_i \in \mathbb{F}_{q^m}$, $\beta_r = 1$ and $L_U(1) = 0$.

Note that knowledge of the full polynomial, i.e., knowledge of all its coefficients $\{\beta_0, \ldots, \beta_{r-1}\}$, allows to recompute the support U, and from the support U we can derive the secret key $\mathbf{x}$, using the linear key equations $\mathbf{Hx} = \mathbf{s}$. Hence, securing these coefficients is as essential as securing the secret key.

Parameters RYDE proposes three different parameter sets for the NIST security levels I, III and V, detailed in Table 4. Recall that those NIST levels correspond to the security equivalent of AES-128, -192, and -256 respectively, and that NIST estimates the bit security of those AES instantiations to 143, 207, and 272 bits.

Table 4 Proposed RYDE parameters

Parameter set	q	n	k	m	r
RYDE-I	2	33	15	31	10
RYDE-III	2	41	18	37	13
RYDE-V	2	47	18	43	17

2.4 The MiRitH and MIRA signature schemes

MiRitH [1] and MIRA [7] are two NIST signature submissions following the MPCitH paradigm. In contrast to RYDE, both are based on the MinRank problem, which is defined as follows.

Definition 6 (MinRank) Let (q, m, n, k, r) be positive integers, with q a prime power and $r < m \leq n$. Given $k+1$ matrices $(\mathbf{M_0}, \ldots, \mathbf{M_k}) \in (\mathbb{F}_q^{m \times n})^{k+1}$, the *MinRank problem* asks to find $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_q$ (if they exist) such that

$$\mathbf{E} := \mathbf{M_0} + \sum_{i=1}^{k} \alpha_i \mathbf{M_i} \tag{1}$$

has rank r.

Note that the Rank-SD and the MinRank problems are related. Indeed, observe that each column of $\mathbf{E} = [\mathbf{e_1} \mid \ldots \mid \mathbf{e_n}]$ forms a vector in $\mathbb{F}_q^m$, or equivalently can be seen as an element in $\mathbb{F}_{q^m}$. This means the vector $\mathbf{e} = (\mathbf{e_1}, \ldots, \mathbf{e_n}) \in \mathbb{F}_{q^m}^n$ behaves similar to the solution $\mathbf{x}$ in the Rank-SD problem, as $W_{\mathrm{R}}(\mathbf{e}) = r$. In other words the entries of $\mathbf{e}$ generate an r-dimensional space

$$U = \mathsf{Supp}(\mathbf{e}) = \langle \mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_r} \rangle, \quad \{i_1, \ldots, i_r\} \subset [n],$$

similar to the low dimensional space generated by the entries of a Rank-SD solution.

MIRA's witness Some of the ideas described in the previous Sect. 2.3 also find application in the MIRA signature scheme, namely it also uses a q-polynomial constructed from the secret key as a witness in the used MPC protocol. In MIRA's case, the annihilator polynomial associated to the linear subspace U is the witness used in the MPC protocol, and is defined as

$$L_U(X) := \prod_{u \in U} (X - u) = \sum_{i=0}^{r} \beta_i X^{q^i},$$

where $\beta_i \in \mathbb{F}_{q^m}$ and $\beta_r = 1$. Note again that knowledge of this polynomial allows to reconstruct $\mathbf{E}$, from which in turn the α_i can be reconstructed by solving the linear system defined by Eq. (1).

Parameters We summarize the suggested MIRA parameters in Table 5 and the proposed MiRitH parameters in Table 6.

3 Partial key exposure on rank-SD

In this section, we describe a partial key exposure attack on the Rank-SD solution with a focus on the RYDE signature scheme.

Table 5 Proposed MIRA parameters

Parameter set	q	n	k	m	r
MIRA-I	16	16	120	16	5
MIRA-III	16	19	168	19	6
MIRA-V	16	22	271	23	6

Table 6 Proposed MiRitH parameters

Parameter set	q	n	k	m	r
MiRitH-Ia	16	15	78	15	6
MiRitH-Ib	16	16	142	16	4
MiRitH-IIIa	16	19	109	19	8
MiRitH-IIIb	16	19	167	19	6
MiRitH-Va	16	21	189	21	7
MiRitH-Vb	16	22	254	22	6

3.1 Secret key format

The secret key in RYDE is the solution $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_{q^m}^n$ to a Rank-SD instance. RYDE stores the secret key in the form of the unique representation $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m})$ of x_i for all i using the canonical basis $(1, X, \ldots, X^{m-1})$ of $\mathbb{F}_{q^m}[X]$ over $\mathbb{F}_q$. Moreover, RYDE uses $q = 2$, i.e., the $(x_{i,j})_{i,j}$ form a bit sequence stored as the secret key. Furthermore, for optimization purposes, it is guaranteed that the support of $\mathbf{x}$ includes $1 \in \mathbb{F}_{q^m}$, i.e., $\mathsf{Supp}(\mathbf{x}) = \langle 1, x_{i_1}, \ldots, x_{i_{r-1}} \rangle$ for $\{i_1, \ldots, i_{r-1}\} \subseteq [n]$.

In the following, we concentrate on the setting $n \geq m$, which is used in RYDE and many other recently suggested parameters for rank-based proposals [1, 7]. Moreover, we stick to the RYDE-choice of $q = 2$ in our explanations, while the algorithms easily extends to arbitrary q.

3.2 Attack in the erasure model

We denote the set of erased bits by $I \subseteq [n] \times [m]$, i.e., the secret key bits $x_{i,j}$ with $(i, j) \in I$ are unknown, while those $x_{i,j}$ with $(i, j) \notin I$ are known.

The known basis approach As we show in the following a basis of the hidden subspace E spanned by the entries of the solution $\mathbf{x} \in \mathbb{F}_{2^m}^n$ is sufficient to recover the whole vector $\mathbf{x}$. While this requires knowledge of an $mr/(mn) = r/n$ fraction of the elements of $\mathbf{x}$, note that the erasure rate that allows for such an attack might be far smaller than $1 - r/n$. This is because erasures appear on the *bitlevel* and, moreover, might be spread uniformly across the full secret key bits. Assuming each bit is erased independently with probability $\delta_r = |I|/(mn)$, we expect that the given partial information reveals a basis if

$$1 - (1 - \delta_r)^m \leq 1 - \frac{r}{n}.$$

This is because each component is represented via m bits, and the probability for no erasure in m bits is $(1 - \delta_r)^m$. Therefore a δ_r satisfying the above inequality leads to on expectation r complete $x_i \in \mathbf{x}$ being revealed. Solving the inequality for δ_r yields $\delta_r \leq 1 - \sqrt[m]{r/n}$. Note that δ_r converges to zero assuming $r/n < c$ for some constant c. Correspondingly, we find

that for actual parameter sets of RYDE, this corresponds to rather small erasure rates in the range of 2-3%.

We overcome this limitation in the following by giving an attack that can directly work with information on the bitlevel of $\mathbf{x}$, leading to *constant* erasure rates in the range of 59% to 64% that still allow for polynomial-time key recovery.

3.2.1 Erasure-enhanced GRS attack

Similar to the GRS attack [19], we aim at finding a subspace $F \subset \mathbb{F}_{q^m}$ that contains the hidden subspace $E = \mathsf{Supp}(\mathbf{x})$. However, we show that the extra information can be exploited to derive additional linear equations in the hidden generators of E. In turn this allows for a choice of subspaces F of larger dimension r', improving the probability of random subspaces F satisfying $E \subseteq F$.

Rank-SD equations Suppose a subspace F with $E \subseteq F$ is known. Then, given a basis $f_1, \ldots, f_{r'}$ of this subspace over $\mathbb{F}_2$, every component x_i of the secret key $\mathbf{x}$ can be written as $x_i = \sum_{j=1}^{r'} \beta_{i,j} f_j$, for some unknown coefficients $\beta_{i,j} \in \mathbb{F}_2$. The secret key relation $\mathbf{Hx} = \mathbf{s}$ leads to $n-k$ different linear equations in the secret key components, where the z-th equation has the form

$$\sum_{\ell=1}^{n} h_{z,\ell} x_\ell = \sum_{\ell=1}^{n} h_{z,\ell} \sum_{j=1}^{r'} \beta_{\ell,j} f_j = s_z,$$

where $h_{z,\ell}$ is the coefficient in the z-th row and ℓ-th column of the parity-check matrix $\mathbf{H}$. This forms a linear system of $n-k$ equations with coefficients over $\mathbb{F}_{2^m}$ and $r'n$ unknowns, namely $\beta_{i,j} \in \mathbb{F}_2$. Moreover, we can embed these equations into $(n-k)m$ equations over $\mathbb{F}_2$, using the m canonical projections $\phi_i : \mathbb{F}_{2^m} \to \mathbb{F}_2$, where $\sum_{i=1}^{m} x_i b_i \mapsto x_i$. More precisely, the z-th equation can be split into m equations of the form

$$\sum_{\ell=1}^{n} \sum_{j=1}^{r'} \beta_{\ell,j} \phi_i(h_{z,\ell} f_j) = \phi_i(s_z), \tag{2}$$

where $i = 1, \ldots, m$. This leads to $(n-k)m$ equations over $\mathbb{F}_2$, while the number of unknowns remains unchanged with $r'n$.

Linear equations from partial knowledge Recall that partial knowledge on the solution $\mathbf{x}$ corresponds to knowledge on some of the $x_{i,j} \in \mathbb{F}_2$ where for a fixed basis $b_1, \ldots, b_m$ of $\mathbb{F}_{2^m}$ every component of $\mathbf{x}$ can be written as $x_i = \sum_{j=1}^{m} x_{i,j} b_j$.

Note that for any given subspace $F = \langle f_1, \ldots, f_{r'} \rangle$ with $E \subseteq F$ we can re-write the secret key components as

$$x_i = \sum_{j=1}^{r'} \beta_{i,j} f_j, \tag{3}$$

where F is known (and correspondingly the f_j), while the coefficients $\beta_{i,j} \in \mathbb{F}_2$ are not. Since by definition we have $F \subseteq \mathbb{F}_{2^m}$, every element f_i of F can be expressed as

$$f_i = \sum_{j=1}^{m} \lambda_{i,j} b_j. \tag{4}$$

Note that the coefficients $\lambda_{i,j} \in \mathbb{F}_2$ are known, in fact, they define F and are chosen randomly to construct F. Together, the last two equations lead to the relation

$$x_i = \sum_{k=1}^{r'} \beta_{i,k} \sum_{j=1}^{m} \lambda_{k,j} b_j,$$

which implies the following relation between the coefficients:

$$x_{i,j} = \sum_{k=1}^{r'} \beta_{i,k} \lambda_{k,j}. \tag{5}$$

or in matrix notation

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_{1,1} & \dots & x_{1,m} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \dots & x_{n,m} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} = \underbrace{\begin{bmatrix} \beta_{1,1} & \dots & \beta_{1,r'} \\ \vdots & \ddots & \vdots \\ \beta_{n,1} & \dots & \beta_{n,r'} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} \lambda_{1,1} & \dots & \lambda_{1,m} \\ \vdots & \ddots & \vdots \\ \lambda_{r',1} & \dots & \lambda_{r',m} \end{bmatrix}}_{\Lambda} \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

This shows that any known bit $x_{i,j}$ of $\mathbf{x}$ corresponds to a linear equation in the unknown coefficients $\beta_{\ell,k}$. Therefore, knowledge of a partially-erased version of the secret key, where t of the $x_{i,j}$ are known, corresponds to t additional linear equations in the unknown coefficients $\beta_{i,j}$.

The full attack Note that together Eqs. (2) and (5) yield a linear system with $r'n$ unkowns, namely the coefficients $\beta_{i,j}$, and $(n-k)m + t$ equations. Therefore as long as

$$(n-k)m + t \geq r'n \quad \Leftrightarrow \quad \frac{(n-k)m + t}{n} \geq r', \tag{6}$$

the system contains more equations than unknowns and it is expected to have a unique solution, which corresponds to a vector $\mathbf{x}$ such that $\mathbf{Hx} = \mathbf{s}$ ans $\mathbf{x}$ is compatible with the given bit information. The attack now selects random subspaces F of dimension $r' = \lfloor \frac{t+m(n-k)}{n} \rfloor$ and solves the system given by Eqs. (2) and (5) until the solution has rank-weight r.

Exploiting the linearity Additionally we embed an improvement that exploits the $\mathbb{F}_{2^m}$-linearity of the code [4]. Therefore, we first inject a codeword into the code to ensure that the vector $\mathbf{x}$ is a codeword in the new code. Then the algorithm proceeds as before on the altered code. This exploits that due to the linearity of the code, all scalar multiples $\alpha\mathbf{x}$ for $\alpha \in \mathbb{F}_{2^m}^*$ also form codewords of the new code. In turn, there are many different subspaces, namely those generated by the support of $\alpha\mathbf{x}$ for different α, that give rise to the secret vector $\mathbf{x}$. Hence, the probability that a random subspace F includes any of those subspaces grows.

We give the pseudocode of the erasure-enhanced GRS attack in Algorithm 1 and summarize its complexity in Theorem 3. Note that Theorem 3 relies on the same heuristic as the attack in [4]. This heuristic regards the independent behavior of the subspaces αE being contained in F, and as been experimentally verified in [4]. We state it as follows.

Heuristic 1 (Independence heuristic) *Let $r, r', m, n \in \mathbb{N}$, $r' > r$ and $n \geq m$. Let $E \subset \mathbb{F}_{2^m}^n$ be a subspace of dimension r. Further let $p := \Pr[E \subseteq F]$ be the probability that a given subspace $F \subset \mathbb{F}_{2^m}^n$ of dimension r' includes E over the random choice of F. Then it holds that*

$$\Pr[\exists \alpha \in \mathbb{F}_{2^m}^* : \alpha E \subseteq F] = \Omega\left(1 - (1-p)^S\right),$$

where $S = |\{\alpha E : \alpha \in \mathbb{F}_{2^m}^*\}|$ is the number of subspaces of the form αE.

Algorithm 1 Erasure-enhanced GRS attack

GRS-Erasure($\mathbf{H}, \mathbf{s}, r, I, (x_{i,j})_{(i,j)\notin I}$)

Input: Parity-check matrix $\mathbf{H} \in \mathbb{F}_{2^m}^{(n-k)\times n}$ of code $\mathcal{C}$, syndrome $\mathbf{s} \in \mathbb{F}_{2^m}^{n-k}$, target rank $r \in \mathbb{N}$ of the solution to Rank-SD instance $(\mathbf{H}, \mathbf{s})$ and $(x_{i,j})_{(i,j)\notin I}$, where $|I| = mn - t$.

Output: Solution $\mathbf{x}$ with $\mathrm{rank}(\mathbf{x}) = r$ and $\mathbf{Hx} = \mathbf{s}$.

$r' \leftarrow \lfloor (t + m(n-k-1))/n \rfloor$
Solve $\mathbf{Hy} = \mathbf{s}$ for arbitrary $\mathbf{y}$.
Compute parity-check matrix $\mathbf{H}'$ of code $\mathcal{C}' = \mathcal{C} \cup \{\mathbf{y}\}$.
Construct linear system of $(n-k-1)m$ equations over $\mathbb{F}_2$ from $\mathbf{H}'\mathbf{x} = \mathbf{0}$ (Eq. (2)).
Construct t additional equations using the known $x_{i,j}$ (Eq. (5)).
repeat
  Choose random r'-dimensional subspace $F \subseteq \mathbb{F}_{2^m}$ by selecting the coefficients $\lambda_{i,j}$, $i \in [r']$, $j \in [m]$ (Eq. (4)).
  Solve the linear system for the $\beta_{i,j}$ and construct $\alpha\mathbf{x}$ (Eq. (3)).
until $W_{\mathrm{R}}(\alpha\mathbf{x}) = r$
return $\mathbf{x}$

Theorem 3 (Erasure-enhanced GRS attack) *Let $m, n, k, r \in \mathbb{N}$ with $n \geq m$. Let $(\mathbf{H}, \mathbf{s})$ be a Rank-SD instance over $\mathbb{F}_{2^m}$ with solution $\mathbf{x}$. Then Algorithm 1 recovers $\mathbf{x}$ under Heuristic 1 in the erasure setting with expected time complexity*

$$T = \mathcal{O}\left(\left((n-k)m + t\right)^3 \cdot \max\left(1, 2^{r\left\lceil \frac{(k+1)m-t}{n} \right\rceil - m}\right)\right),$$

where $t = mn - |I|$ is the number of known secret key bits.

Proof Note that the initial preprocessing, i.e., the injection of the codeword, the computation of the parity-check matrix $\mathbf{H}'$ and the construction of the linear system, can be done in polynomial time. Therefore, the running time is dominated by the repeat loop. The expected time complexity of the loop and, hence, the algorithm is the time complexity of each iteration of the loop divided by the success probability of one iteration.

The time of each loop iteration is dominated by the time of solving a linear system with $r'n \leq (n-k-1)m + t$ unknowns and $(n-k-1)m + t$ equations, that is, $\mathcal{O}\left(\left((n-k)m + t\right)^3\right)$.

As shown previously, the algorithm is successful whenever the chosen subspace F includes at least one of the subspaces αE, where $\alpha \in \mathbb{F}_{2^m}^*$, and E is the r-dimensional space spanned by the entries of the secret key $\mathbf{x} \in \mathbb{F}_{2^m}^n$. The probability of F including the subspace αE for any fixed α is

$$p := \Pr[\alpha E \subseteq F] = \Pr[E \subseteq F] = \begin{bmatrix} r' \\ r \end{bmatrix}_2 \Big/ \begin{bmatrix} m \\ r \end{bmatrix}_2 = \Omega(2^{-r(m-r')}),$$

which is the number of subspaces of dimension r in F divided by the number of r-dimensional subspaces in $\mathbb{F}_{2^m}$, where the last equality follows from Lemma 1.

Now, under Heuristic 1 to find the probability that F includes at least one of the subspaces αE, we can treat those probabilities as independent for different subspaces αE, which gives

$$q := \Pr[\exists \alpha \in \mathbb{F}_{2^m}^* : \alpha E \subseteq F] = 1 - (1-p)^S \geq \frac{Sp}{Sp+1} = \Omega\big(\min(1, Sp)\big),$$

where S is the number of different subspaces in $\{\alpha E\}_{\alpha \in \mathbb{F}_{2^m}^*}$. Note that in [4, Proposition III.1] it is shown that for a binary extension field, S is maximal with high probability, i.e. $S = 2^m - 1$. Overall this leads to a complexity of

$$T = \mathcal{O}\left(\big((n-k)m + t\big)^3 / q\right) = \mathcal{O}\left(\big((n-k)m + t\big)^3\right) \max\left(1, 2^{r(m-r')-m}\right),$$

as claimed, since

$$m - r' = m - \left\lfloor \frac{t + m(n-k-1)}{n} \right\rfloor = m - \left\lfloor m - \frac{(k+1)m - t}{n} \right\rfloor = \left\lceil \frac{(k+1)m - t}{n} \right\rceil$$

□

Remark 1 (Dependence on Heuristic 1) The dependence on Heuristic 1 is an artifact of embedding the improvement from [4]. If not using the improvement, we do not need to rely on any heuristics at the cost of a changed complexity exponent of $r\lceil \frac{km-t}{n} \rceil$.

3.2.2 Polynomial-time key-recovery

Note that in the specific case of no knowledge on $\mathbf{x}$, corresponding to $t = 0$ known bits, Algorithm 1 collapses to the linearity-improved GRS algorithm. In all other cases, the algorithm's complexity exponent is reduced and decreases linearly in t/n. Moreover, the following theorem shows that up to an erasure rate of roughly $1 - k/n + 1/r$ the secret $\mathbf{x}$ can be recovered in polynomial time using Algorithm 1.

Theorem 4 (Polynomial Erasure Attack) *Let $n, m, k, r \in \mathbb{N}$ with $n \geq m$. Let $(\mathbf{H}, \mathbf{s})$ be a Rank-SD instance over $\mathbb{F}_{2^m}$ with solution $\mathbf{x}$. Then $\mathbf{x}$ can be recovered in the erasure setting in polynomial time up to an erasure rate $p \leq \delta$ with*

$$1 - \frac{k+1}{n} + \frac{1}{r} \quad \geq \quad \delta \quad \geq \quad 1 - \frac{k+1}{n} + \frac{1}{r} - \frac{1}{m}.$$

Proof The time complexity of Algorithm 1 is polynomial (compare to Theorem 3) whenever

$$\begin{aligned} & 0 && \geq r\left\lceil \frac{(k+1)m - t}{n} \right\rceil - m = r\left(\frac{(k+1)m - t}{n} + \varepsilon\right) - m \\ \Leftrightarrow \quad & t && \geq (k+1)m + \varepsilon n - \frac{mn}{r} \end{aligned}$$

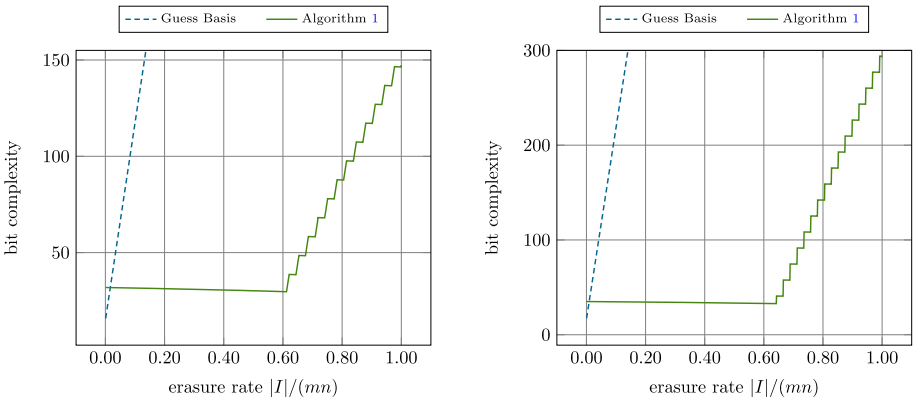
Since the solution to a Rank-SD instance over $\mathbb{F}_{2^m}$ consists of nm bits, for an erasure rate of $p \leq 1 - t/mn$ we obtain knowledge of at least t secret key bits. Hence, we find that up to an erasure rate of

$$p = 1 - \frac{t}{mn} \leq 1 - \frac{k+1}{n} - \frac{\varepsilon}{m} + \frac{1}{r} = \delta,$$

the secret $\mathbf{x}$ can be recovered in polynomial time with constant probability. The statement of the theorem follows from observing that $0 \leq \varepsilon \leq 1$. □

Table 7 Maximum erasure rates that allow for polynomial-time key recovery

	Known-basis approach (Sect. 3.2)	Erasure-enhanced GRS (Remark 1)	Erasure-enhanced improved GRS (Theorem 4)
RYDE-I	0.04	0.55	0.61
RYDE-III	0.03	0.56	0.59
RYDE-V	0.02	0.62	0.64



(a) RYDE I parameters (b) RYDE V parameters

Fig. 1 Bit complexity of the erasure-enhanced GRS attack (Algorithm 1, solid) and the guessing basis approach (dashed) applied to the RYDE-I and RYDE-V parameter sets

3.2.3 Application to RYDE parameters

In Table 7 we summarize the erasure probabilities that lead to polynomial-time key recoveries for RYDE parameters when applying (1) the known-basis approach detailed at the beginning of the section (2) the erasure-enhanced GRS attack not relying on any heuristics and (3) the erasure-enhanced linearity-improved GRS attack. We observe that the erasure-enhanced GRS attack greatly outperforms the known-basis approach, due to its ability to leverage information on the bitlevel.[1]

In Fig. 1 we compare the concrete bit complexity of the erasure-enhanced GRS attack against the known basis approach. In case no basis is revealed by the given information, the remaining bits to obtain a basis are guessed. Note that this attack, in contrast to Algorithm 1 depends on the distribution of the erasures. In the comparison we assume erasures distribute uniformly over all coordinates, which implies that the number of bits that have to be guessed is about $\delta \cdot mr$. We account for the polynomial factors of this approach with $(nr)^3$, which corresponds to solving $\mathbf{Hx} = \mathbf{s}$ given a basis of the hidden subspace. It can be observed that, in contrast to the known basis approach, Algorithm 1 yields an attack below the security level for any number of known bits $t > 0$.

[1] Note that the probability for the erasure-enhanced GRS attack (Theorem 4) in the table is computed exactly by determining the smallest t such that the max statement in the running time of Theorem 3 evaluates to one.

Table 8 Maximum erasure rates that allow for a key recovery with less than 2^{60} and 2^{80} operations

	60-Bit		80-Bit	
	Guess basis	Algorithm 1	Guess basis	Algorithm 1
RYDE-I	0.04	0.71	0.06	0.78
RYDE-III	0.03	0.67	0.04	0.70
RYDE-V	0.02	0.69	0.03	0.71

When allowing for a certain, non-polynomial, but reasonably low time complexity constraint of 60- or 80-bit, we find that key recovery remains possible up to 78% of erased key material using Algorithm 1, as shown in Table 8.

3.3 Impact on RYDE parameters

In the following, we observe that the erasure-enhanced GRS attack leads to a natural improvement of the GRS attack. This improvement leads in certain cases to a reduced complexity for recovering the Rank-SD solution *even without additional knowledge*, and impacts RYDE security levels. This improvement stems from the discontinuity of the runtime formula, namely the ceiling of the exponent (see Theorem 3). This ceiling is necessary to ensure that there are *at least* as many equations as unknowns to be able to solve the resulting linear system. Now, we find that by guessing a few bits of $\mathbf{x}$ we can control the number of equations to balance equations and unknowns without the necessity of ceiling.

More precisely, the runtime exponent of the attack when t bits of $\mathbf{x}$ are known is given in Theorem 3 as $r\left\lceil\frac{(k+1)m-t}{n}\right\rceil - m$. Furthermore even if no bits of $\mathbf{x}$ are known Algorithm 1 can be applied with parameter $t > 0$ by first guessing t bits of $\mathbf{x}$. Clearly this attack can only succeed for a correct guess on those bits. However, reapplying the attack for all possible 2^t choices for those bits of $\mathbf{x}$ eventually leads to a successful attack with a multiplicative runtime overhead of 2^t.

The following proposition summarizes the complexity of this guessing variant.

Proposition 5 (Guessing-enhanced GRS attack) *Let $m, n, k, r \in \mathbb{N}$ with $n \geq m$. Let $(\mathbf{H}, \mathbf{s})$ be a Rank-SD instance over $\mathbb{F}_{2^m}$ with solution $\mathbf{x}$. Then, $\mathbf{x}$ can be recovered with expected time complexity*

$$T = \mathcal{O}\left(\left((n-k)m+t\right)^3 \cdot \max\left(1, 2^{r\left\lceil\frac{(k+1)m-t}{n}\right\rceil - m + t}\right)\right),$$

for any $t \leq nm$.

Note that the attack complexity given in Proposition 5 is superior to the linearity-improved GRS attack, whenever there is a $t > 0$ that minimizes the runtime expression. Usually, this is the case when there is a $0 < t < r$ such that

$$\left\lceil\frac{(k+1)m-t}{n}\right\rceil = \left\lceil\frac{(k+1)m}{n}\right\rceil - 1.$$

In this case, the runtime exponent is reduced by r, which compensates for the addition of t, which leads to an overall improvement by a factor of 2^{r-t}. For the suggested RYDE parameter sets we find that the choices of $t = 1$ (RYDE-I) and $t = 6$ (RYDE-III) satisfy the above and lead to corresponding bit security reductions by $r - t = 10 - 1 = 9$ bits and

Table 9 Bit security levels of RYDE parameter sets

	Improved GRS [4]	Guessing-enhanced GRS (Theorem 5)
RYDE-I	147	**138**
RYDE-III	216	**210**
RYDE-V	**283**	**283**

Highlight in bold the best attack for each security level after evaluating our guessing approach

$r - t = 13 - 6 = 7$ bits, respectively. For the RYDE-V parameter set, the smallest t leading to such a decrease is $t = r + 1$, which therefore does not give an improvement.

We summarize the new security levels according to this attack improvement in Table 9. Note that for deriving those numbers, we follow the RYDE specification, which conservatively assumes that the linear system can be solved with exponent $w = 2$ (rather than $w = 3$). Therefore, even though the numbers lie below the NIST security threshold of 143 in case of Category-I the practical security guarantees are likely still high enough.

3.4 Attack in the error model

Recall that in the error model an erroneous secret key $\widetilde{\mathbf{x}} = \mathbf{x} + \mathbf{e} = (x_{i,j} + e_{i,j})_{i,j} = (\tilde{x}_{i,j})_{i,j}$, where $x_{i,j}, e_{i,j} \in \mathbb{F}_2$ for $(i, j) \in [n] \times [m]$ is obtained, where at most $\delta \cdot nm$ of the $e_{i,j}$ are non-zero for given $\delta \in (0, 1)$.

Connection to syndrome decoding in the Hamming metric Note that the sparseness of the error leads to a direct connection to the Hamming metric decoding problem. Hamming metric decoding asks to find a solution $\mathbf{e}$ satisfying $\mathbf{He} = \mathbf{s}'$, where $\mathbf{e}$ has small *Hamming*-weight. Note that

$$\mathbf{H}\tilde{\mathbf{x}} = \mathbf{H}(\mathbf{x} + \mathbf{e}) = \mathbf{s} + \mathbf{He}.$$

Therefore $\mathbf{e}$ is a solution to the syndrome decoding instance $(\mathbf{H}, \mathbf{s}')$, where $\mathbf{s}' := \mathbf{H}\tilde{\mathbf{x}} - \mathbf{s}$. Embedding this instance over $\mathbb{F}_2$ yields $\mathbf{H} \in \mathbb{F}_2^{m(n-k)\times mn}$ and an error $\mathbf{e} \in \mathbb{F}_2^{mn}$ of Hamming weight $\delta \cdot mn$. This leads to a first potential attack in the error setting, which is recovering the error $\mathbf{e}$ by applying decoding techniques in the Hamming metric.

In the following, we describe additionally a generic translation of the erasure attack from Sect. 3.2 to the error setting. In this translation we also exploit the sparseness of $\mathbf{e}$ by guessing zeros in the error and then perform the erasure-enhanced (improved) GRS attack (Algorithm 1). Therefore observe, that when t correct zero positions are guessed, an application with parameter t of Algorithm 1 recovers the solution. The pseudocode of this adaptation is given by Algorithm 2 and its complexity is summarized in the following lemma.

Lemma 6 (Error-enhanced GRS attack) *Let $m, n, k, r \in \mathbb{N}$ with $n \geq m$. Let $(\mathbf{H}, \mathbf{s})$ be a Rank-SD instance over $\mathbb{F}_{2^m}$ with solution $\mathbf{x}$. Given an erroneous version $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{e}$ of $\mathbf{x}$ with error rate δ, Algorithm 2 recovers $\mathbf{x}$ under Heuristic 1 with expected time complexity*

$$T = \mathcal{O}\left(\frac{\binom{mn}{t}}{\binom{(1-\delta)mn}{t}} \cdot \left((n-k)m + t\right)^3 \cdot \max\left(1, 2^{r\left\lceil \frac{(k+1)m-t}{n} \right\rceil - m}\right) \right),$$

for any integer $t \leq (1 - \delta) \cdot mn$.

Proof Given in Appendix A.1. □

Algorithm 2 Error-enhanced GRS attack

GRS-Error($\mathbf{H}$, $\mathbf{s}$, r, $\tilde{\mathbf{x}}$, δ)

Input: Parity-check matrix $\mathbf{H} \in \mathbb{F}_{2^m}^{(n-k)\times n}$ of code $\mathcal{C}$, syndrome $\mathbf{s} \in \mathbb{F}_{2^m}^{n-k}$, target rank $r \in \mathbb{N}$, erroneous version $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{e}$ of the solution to Rank-SD instance $(\mathbf{H}, \mathbf{s})$ where $W_{\mathrm{H}}(\mathbf{e}) \leq \delta mn$.
Output: Solution $\mathbf{x}$ with $\mathrm{rank}(\mathbf{x}) = r$ and $\mathbf{Hx} = \mathbf{s}$.

Choose optimal t
repeat
  Guess t zero bits in $\mathbf{e}$, i.e., guess $J \subseteq [n] \times [m]$ with $|J| = t$ such that $e_{i,j} = 0$ for $(i, j) \in J$
  Execute Algorithm 1 with inputs $\mathbf{H}$, $\mathbf{s}$, r and $(\tilde{x}_{i,j})_{(i,j)\in J}$ of the solution to Rank-SD instance $(\mathbf{H}, \mathbf{s})$.
until Algorithm 1 returns solution $\mathbf{x}$
return $\mathbf{x}$

3.4.1 Polynomial-time error attack

We find that the complexity of the attack remains polynomial up to an error probability of $\delta = \mathcal{O}\left(\frac{\log mn}{mn}\right)$. We formalize this in the following theorem and deliver the proof in Appendix A.2 for completeness.

Theorem 7 *(Polynomial Error Attack) Let $n, m, k, r \in \mathbb{N}$ with $n \geq m$. Let $(\mathbf{H}, \mathbf{s})$ be a Rank-SD instance over $\mathbb{F}_{2^m}$ with solution $\mathbf{x}$. Given an erroneous version $\tilde{\mathbf{x}}$ of $\mathbf{x}$ with error probability $p = \mathcal{O}(\frac{\log(mn)}{mn})$, then $\mathbf{x}$ can be recovered in polynomial time with constant probability.*

Note that the error probability of Theorem 7 matches the result obtained for codes in the Hamming metric in [16], when using the translation to the Hamming metric detailed at the beginning of the section. However, when comparing concrete complexities on actual RYDE parameter sets, the complexities of both approaches differ.

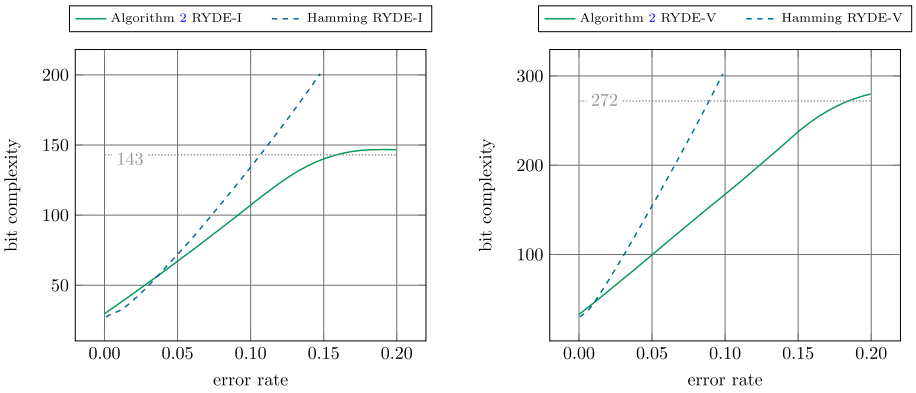
3.4.2 Concrete complexity of error attacks

In Fig. 2 we provide the complexities of both strategies as a function of the error probability for the RYDE-I (left) and RYDE-V (right) parameter sets. For estimating the hardness of the binary Hamming-metric syndrome decoding instance we use the *CryptographicEstimators* library [17], which incorporates the *Syndrome Decoding Estimator* tool [15]. We observe that for low error probabilities the translation to the Hamming case is favorable, while for larger error rates, the adaptation of the erasure-enhanced GRS attack achieves a better complexity.

In Table 10 we summarize the error rates that allow for key recovery below the respective security levels, implying that up to these error rates the leaked information can be exploited to improve over generic key recovery attacks.

4 Partial key exposure on MinRank

In this section, we describe a partial key exposure attack on the private key of a MinRank instance. We demonstrate the effectiveness of the attack by application to the MIRA and MiRitH signature schemes.



(a) RYDE I parameters (b) RYDE V parameters

Fig. 2 Bit complexity of the error-enhanced GRS attack (Algorithm 2, green line) or decoding techniques in the Hamming metric (Dumer's algorithm [13], dashed blue line) applied to the parameter sets of RYDE-I and RYDE-V

Table 10 Maximum error rates that allow for a key recovery with less than 2^{143}, 2^{207} and 2^{272} operations for RYDE-I, -III and -V respectively

	Hamming decoding	Algorithm 2
RYDE-I	0.107	0.159 ($t = 133$)
RYDE-III	0.102	0.162 ($t = 252$)
RYDE-V	0.089	0.184 ($t = 300$)

Recall that the solution to a MinRank instance is a linear combination of the input matrices that yields a low rank matrix $\mathbf{E} = \mathbf{M}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{M}_i$.

Secret key format

The secret key in MIRA and MiRitH is the solution $\alpha = (\alpha_1, \ldots, \alpha_k) \in \mathbb{F}_q^k$ of a MinRank instance. Both schemes use an extension field with $q = 2^4$. The coefficients are then represented on the bitlevel via a basis $b_1, \ldots, b_4$ of $\mathbb{F}_{16}$ as a $\mathbb{F}_2$-linear space. That is, each coefficient $\alpha_i \in \mathbb{F}_{16}$ is represented via the (unique) string of four bits $\alpha_{i,1}, \ldots, \alpha_{i,4} \in \mathbb{F}_2$ such that $\alpha_i = \sum_{j=1}^{4} \alpha_{i,j} b_j$. We remark that $b_1, \ldots, b_4$ are determined by the implementation of the scheme.

4.1 Attack in the erasure model

Note that once a coefficient α_i is known the MinRank instance can be naturally reduced to an instance with decreased value of k. Therefore the new instance is defined as $(\mathbf{M}_0 + \alpha_i \mathbf{M}_i, \mathbf{M}_1, \ldots, \mathbf{M}_{i-1}, \mathbf{M}_{i+1}, \ldots, \mathbf{M}_k)$ with solution $(\alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k)$. The resulting instance is reduced by one matrix and can therefore be solved more efficiently. However, since MiRitH as well as MIRA do not use binary coefficients but coefficients from binary extension fields, bit knowledge on the secret key does not directly translate into coefficient knowledge.

A first straightforward approach achieves the translation between bits and coefficients by guessing the remaining bits to obtain full coefficients. We briefly outline this technique in the following to obtain a baseline for later comparison.

Bruteforce Approach Given an erasure rate of δ, we expect $\delta \cdot d$ erasures per coefficient, where $q = 2^d$. Therefore guessing the remaining bits of a coefficient α_i has a complexity of $2^{\lceil \delta d \rceil}$, where d is such that $q = 2^d$. As detailed above, each correct guess of a coefficient reduces the k of the MinRank instance by one. Therefore, the complexity of recovering the secret key when ℓ guesses are made becomes

$$\min_{\ell=0,\dots,k} \left\{ 2^{\lceil \ell \delta d \rceil} \cdot \mathbb{C}(q, m, n, k-\ell, r) \right\}, \tag{7}$$

where $\mathbb{C}(q, m, n, k, r)$ is the complexity of solving a MinRank instance with parameters (q, m, n, k, r).

Note that, apart from a huge guessing overhead, the approach disregards the known bits for any coefficient not part of the guessing. In the following we improve on the bruteforce approach by giving a general reduction from any MinRank instance over an extension field $\mathbb{F}_{p^k}$ to an instance over the basefield $\mathbb{F}_p$. Moreover, we show that in case of $p = 2$ the bits of the secret key are the coefficients of the resulting MinRank instance. In case of MIRA and MiRitH this allows to directly exploit known bits to reduce the resulting instance.

4.1.1 Rewriting a MinRank instance over the base field

Let $\mathbb{E}/\mathbb{F}$ be a field extension of finite degree d. Let us describe an algorithm that transforms any MinRank instance defined over $\mathbb{E}$ into an equivalent MinRank instance defined over $\mathbb{F}$. At its core the reduction employs the fact that elements of $\mathbb{E}$ can be written as $d \times d$ matrices over $\mathbb{F}$. This representation is then exploited to construct multiple matrices which are subsequently injected into the new instance over the basefield.

Fix a basis $b_1, \dots, b_d$ of $\mathbb{E}$ as a vector space over $\mathbb{F}$. For each $a \in \mathbb{E}$, let $\mathbf{X}_a \in \mathbb{F}^{d \times d}$ be the matrix associated to the $\mathbb{F}$-linear map $\mathbb{E} \to \mathbb{E}\colon x \to ax$ with respect to the basis $b_1, \dots, b_d$. More precisely, the jth column of $\mathbf{X}_a$ consists of the coefficients of ab_j written with respect to the basis $b_1, \dots, b_d$. Thus, $\mathbf{X}_{ab} = \mathbf{X}_a \mathbf{X}_b$ for every $a, b \in \mathbb{E}$. For every matrix $\mathbf{A} \in \mathbb{E}^{s \times t}$ let Φ be the component wise application of this map. Therefore $\Phi(\mathbf{A}) \in \mathbb{F}^{ds \times dt}$ is the (block) matrix obtained from $\mathbf{A}$ by replacing each entry a of $\mathbf{A}$ with the matrix $\mathbf{X}_a$. The following theorem now states the reduction to MinRank instance over the basefield.

Theorem 8 (Reduction to Base Field MinRank Instance) *Let $\mathcal{M} = (\mathbf{M}_0, \dots, \mathbf{M}_k) \in \mathbb{E}^{m \times n}$ be a MinRank instance with parameters m, n, k, r. For $\alpha_1, \dots, \alpha_k \in \mathbb{E}$, write $\alpha_i = \sum_{j=1}^{d} \widetilde{\alpha}_{i,j} b_j$ with $\widetilde{\alpha}_{i,j} \in \mathbb{F}$.*

Let $\widetilde{\mathbf{M}}_0 := \Phi(\mathbf{M}_0)$, let $\widetilde{\mathbf{M}}_{i,j} := \Phi(b_j \mathbf{M}_i)$ for every $i \in \{1, \dots, k\}$ and $j \in \{1, \dots, d\}$, and let $\widetilde{\mathcal{M}}$ be the MinRank instance over $\mathbb{F}$ with parameters dm, dn, dk, dr consisting of the $dk + 1$ matrices $\widetilde{\mathbf{M}}_0, \widetilde{\mathbf{M}}_{i,j}$.

Then $\alpha_1, \dots, \alpha_k$ is a solution to the MinRank instance $\mathcal{M}$ if and only if $\widetilde{\alpha}_{i,j}$ is a solution to the MinRank instance $\widetilde{\mathcal{M}}$.

Before proving Theorem 8, let us introduce the following three lemmas about the behavior of the map Φ.

Lemma 9 *Let $\mathbf{A} \in \mathbb{E}^{s \times t}$ and $\mathbf{B} \in \mathbb{E}^{t \times u}$. Then $\Phi(\mathbf{AB}) = \Phi(\mathbf{A})\Phi(\mathbf{B})$.*

Proof The claim is a straightforward consequence of performing the block-wise multiplication $\Phi(\mathbf{A})\Phi(\mathbf{B})$ and employing the property $\mathbf{X}_{ab} = \mathbf{X}_a\mathbf{X}_b$ for every $a, b \in \mathbb{E}$. □

Lemma 10 *Let $\mathbf{A} \in \mathbb{E}^{s\times s}$. If $\mathbf{A}$ is invertible, then $\Phi(\mathbf{A})$ is invertible and in fact $\Phi(\mathbf{A})^{-1} = \Phi(\mathbf{A}^{-1})$.*

Proof Since $\Phi(\mathbf{I}_s) = \mathbf{I}_{ds}$, the claim follows from Lemma 9. □

Lemma 11 *Let $\mathbf{A} \in \mathbb{E}^{s\times t}$. Then* $\operatorname{rank}(\Phi(\mathbf{A})) = d\operatorname{rank}(\mathbf{A})$.

Proof Let $r := \operatorname{rank}(\mathbf{A})$. Then there exist invertible matrices $\mathbf{S} \in \mathbb{E}^{s\times s}$ and $\mathbf{T} \in \mathbb{E}^{t\times t}$ such that $\mathbf{A} = \mathbf{SBT}$, where $\mathbf{B} = \begin{pmatrix} \mathbf{I}_r & * \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$. Thus by Lemma 9 we get that $\Phi(\mathbf{A}) = \Phi(\mathbf{SBT}) = \Phi(\mathbf{S})\Phi(\mathbf{B})\Phi(\mathbf{S})$, where $\Phi(\mathbf{B}) = \begin{pmatrix} \mathbf{I}_{dr} & * \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ has rank dr and, by Lemma 10, $\Phi(\mathbf{S})$ and $\Phi(\mathbf{T})$ are invertible. Hence, $\operatorname{rank}(\Phi(\mathbf{A})) = dr$, as desired. □

Now, we are ready to deliver the proof of Theorem 8.

Proof of Theorem 8 We have that

$$\mathbf{M}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{M}_i = \mathbf{M}_0 + \sum_{i=1}^{k}\sum_{j=1}^{d} \widetilde{\alpha}_{i,j} b_j \mathbf{M}_i. \tag{8}$$

Suppose that $\mathbf{E} = \mathbf{M}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{M}_i$, where $\mathbf{E} \in \mathbb{E}^{m\times n}$ has rank r. Then, applying Φ to (8), taking into account that Φ is $\mathbb{F}$-linear, we get that

$$\Phi(\mathbf{E}) = \widetilde{\mathbf{M}}_0 + \sum_{i=1}^{k}\sum_{j=1}^{d} \widetilde{\alpha}_{i,j} \widetilde{\mathbf{M}}_{i,j},$$

where, by Lemma 11, the matrix $\Phi(\mathbf{E})$ has rank dr. Hence, we have that $\widetilde{\alpha}_{i,j}$ is a solution to the MinRank instance $\widetilde{\mathcal{M}}$.

Vice versa, suppose that $\mathbf{H} = \widetilde{\mathbf{M}}_0 + \sum_{i=1}^{k}\sum_{j=1}^{d} \alpha_{i,j} \widetilde{\mathbf{M}}_{i,j}$, where $\mathbf{H} \in \mathbb{F}^{dm\times dn}$ has rank dr. Then by (8) we have that

$$\mathbf{H} = \Phi\left(\mathbf{M}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{M}_i\right).$$

Letting $\mathbf{E} := \mathbf{M}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{M}_i$, from Lemma 11 we get that $\mathbf{E}$ has rank r. Thus $\alpha_1, \ldots, \alpha_k$ is a solution to the MinRank instance $\mathcal{M}$. □

For both MIRA and MiRitH, it holds that $\mathbb{E} = \mathbb{F}_{2^4}$ and $\mathbb{F} = \mathbb{F}_2$. Therefore, from now on we focus on these two fields. Algorithm 3 recaps the steps which are necessary to transform an instance for these schemes into an equivalent instance defined on the basefield.

4.1.2 Obtaining a reduced instance

As before, we denote the set of erased bits by $I \subseteq [d] \times [k]$, i.e., the secret key bits $\alpha_{i,j}$ with $(i, j) \in I$ are unknown, while those $\alpha_{i,j}$ with $(i, j) \notin I$ are known. Suppose we know t bits of the private key $\{\alpha_{i,j}\}_{i,j}$, $\alpha_{i,j} \in \mathbb{F}_2$ of a given MinRank instance over $\mathbb{F}_{2^d}$, that is $|I| = dk - t$.

Algorithm 3 Algorithm for transforming a MinRank instance defined on $\mathbb{F}_q$ with $q = 2^d$ into an equivalent instance on $\mathbb{F}_2$.

BaseField-MinRank($\{\mathbf{M_i}\}_{i=0,\dots,k}$)

Input: MinRank instance $(\mathbf{M}_0, \dots, \mathbf{M}_k, r)$ with parameters $(q = 2^d, m, n, k, r)$ and solution $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$

Output: MinRank instance $(\widetilde{\mathbf{M}}_0, \widetilde{\mathbf{M}}_{1,1} \dots, \widetilde{\mathbf{M}}_{d,k})$ with parameters $q = 2, dm, dn, dk, dr$ and solution $(\alpha_{i,j})_{j\in[d],i\in[k]}$ where $\alpha_i = \sum_{j=1}^{d} \alpha_{i,j} b_j$

1 : For each $a \in \mathbb{F}_q$, let $\mathbf{X}_a \in \mathbb{F}_2^{d\times d}$ be the matrix associated to the $\mathbb{F}_2$-linear map $\mathbb{F}_q \to \mathbb{F}_q : x \to ax$ with respect to the basis $b_1, \dots, b_d$.

2 : For each $\mathbf{M} \in \mathbb{F}_q^{m\times n}$, let $\Phi(\mathbf{M}) \in \mathbb{F}_2^{dm\times dn}$ be the (block) matrix obtained from $\mathbf{M}$ by replacing each entry a of $\mathbf{M}$ with the matrix $\mathbf{X}_a$.

3 : Set $\widetilde{\mathbf{M}}_0 \leftarrow \Phi(\mathbf{M}_0)$. Compute $\widetilde{\mathbf{M}}_{i,j} \leftarrow \Phi(b_j \mathbf{M}_i)$, with $1 \le i \le k$ and $1 \le j \le d$.

4 : **return** $(\widetilde{\mathbf{M}}_0, \widetilde{\mathbf{M}}_{1,1} \dots, \widetilde{\mathbf{M}}_{d,k})$

Theorem 8 shows how to transform the original instance into an instance over the base-field $\mathbb{F}_2$. Moreover, note that for MIRA and MiRitH, using binary extension fields, the secret key bits $\alpha_{i,j}$ directly correspond to the coefficients of the the MinRank instance $\widetilde{\mathcal{M}}$ output by Algorithm 3, when choosing the same basis $b_1, \dots, b_d$ as in the implementation. Therefore, every known bit can be leveraged to decrease the dimension of the matrix code related to the instance $\widetilde{\mathcal{M}}$ by one. Overall, this leads to a MinRank instance with parameters $(2, dm, dn, |I|, dr)$.

4.1.3 Complexity of solving the reduced instance

Generally there are, similar to Rank-SD, two different approaches to solve MinRank instances, algebraic and combinatorial techniques. For the parameters encountered in the instances derived from Algorithm 3 for proposed MiRitH and MIRA parameters, we find combinatorial techniques to perform best using the publicly available MinRank estimator provided within in CryptographicEstimators library [17]. In particular, we observe that the Kernel-Search (KS) algorithm [21] achieves the best time complexity for nearly all erasure rates, with single bit differences where this is not the case. Given these negligible complexity differences and the fact that the KS algorithm allows for a closed-form complexity expression, we focus on the KS algorithm to solve the reduced instance obtained from Algorithm 3 in the following. The following lemma summarizes the complexity of the Kernel-Search algorithm to solve a MinRank instance.

Lemma 12 (Kernel-Search, [21]) *Let q, m, n, k, r be positive integers, where q is a prime power. Then a solution to the MinRank problem with parameters (q, m, n, k, r) can be found in time $\mathcal{O}(q^{r\lceil \frac{k}{m} \rceil} k^w)$, where w is the constant of linear algebra.*

We now apply the Kernel-Search algorithm to the solve reduced instance obtained from Algorithm 3 and correspondingly recover the secret key. This procedure is summarized in Algorithm 4, and its complexity is summarized in the following theorem.

Algorithm 4 Erased Key—Kernel Search Attack

KS-erasure($\{\mathbf{M_i}\}_{i=0,\dots,k}, r, I, (\alpha_{i,j})_{(i,j)\notin I}$)

Input: MinRank instance $(\mathbf{M}_0, \dots, \mathbf{M}_k)$ with parameters $(q = 2^d, m, n, k, r)$ and $(\alpha_{i,j})_{(i,j)\notin I}$, where $|I| = dk - t$.

Output: Solution $\alpha = (\alpha_1, \dots, \alpha_k) \in \mathbb{F}_q^k$ such that $\mathbf{E} = \mathbf{M}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{M}_i$ and $\text{rank}(\mathbf{E}) = r$.

1 : Use Algorithm 3 to obtain an equivalent MinRank instance over $\mathbb{F}_2$ with parameters $(2, dm, dn, dk, dr)$.

2 : Incorporate the knowledge of the bits $(\alpha_{i,j})_{(i,j)\notin I}$ to obtain a new instance with parameters $(2, dm, dn, |I|, dr)$.

3 : Solve the latter instance using Kernel Search to obtain solution $(\alpha_{i,j})_{j\in[d],i\in[k]}$.

4 : **return** $(\alpha_i)_{i\in[k]}$ with $\alpha_i = \sum_{j=1}^{d} \alpha_{i,j} b_j$

Theorem 13 (Erasure-enhanced Kernel Search Attack) *Let d, n, m, k, r be positive integers and $q = 2^d$. Let $\{\mathbf{M}_i\}_{i=0,\dots,k}$ be a MinRank instance over $\mathbb{F}_q$ with solution α. Then Algorithm 4 recovers α in the erasure setting with expected time complexity*

$$T = \mathcal{O}\left(2^{dr\left\lceil \frac{|I|}{dm} \right\rceil} \cdot |I|^w\right),$$

where I is the index set of missing coefficient bits.

Proof Note that the transformation of the starting instance into a base field instance using Algorithm 3 can be performed in polynomial time. Therefore, the computational time of Algorithm 4 is equal to the complexity of the Kernel Search algorithm applied to an instance of parameters $(2, dm, dn, |I|, dr)$, which is $\mathcal{O}\left(2^{dr\left\lceil \frac{|I|}{dm} \right\rceil} \cdot |I|^w\right)$ (see Lemma 12). □

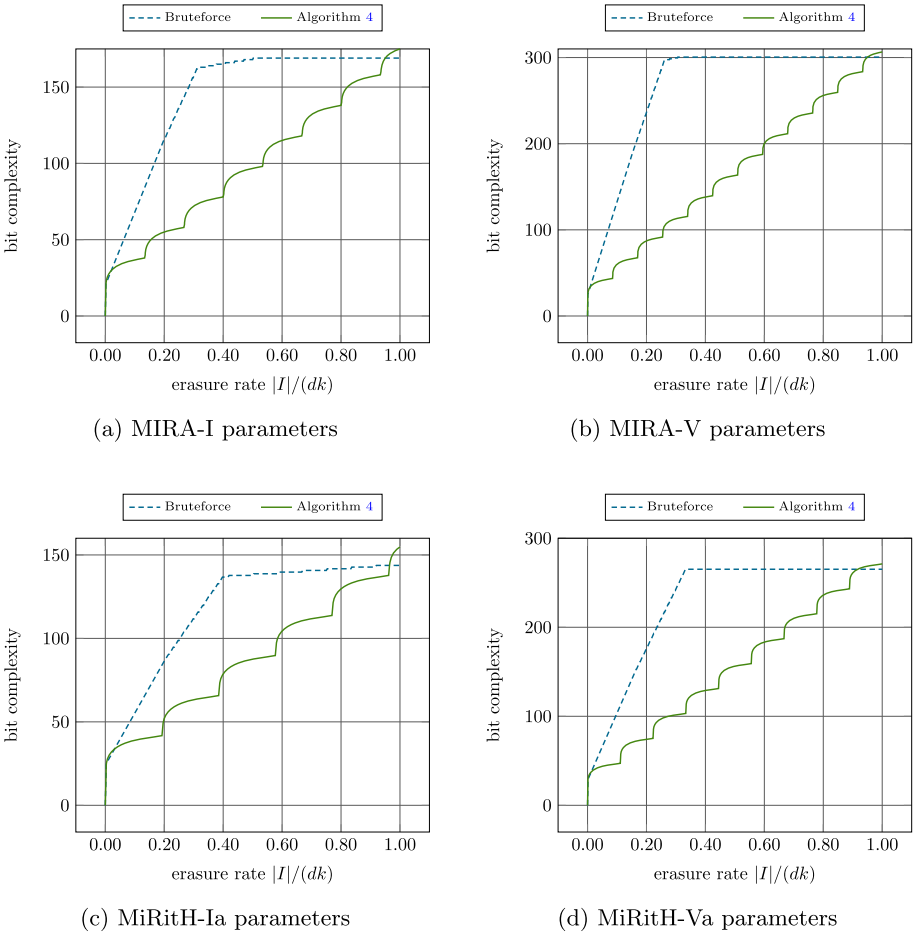
Note that due to the ceiling of $\frac{|I|}{dm}$ in the exponent, there is, in contrast to Sect. 3, no erasure regime for which the complexity statement of Theorem 13 yields a polynomial running time. However, we find that with respect to concrete complexities there are large regimes for which a recovery is possible under reasonable runtime constraints.

4.1.4 Application to MIRA and MiRitH parameters

We now compare the concrete complexities of the reduction approach (Algorithm 4) and the brute force approach (see beginning of Sect. 4.1.1). For estimating the complexity of solving the MinRank instance, we use the CryptographicEstimators library [17], which on top of Theorem 16 allows for some hybrid-guessing techniques, which can lead to some gains in practice.

Figure 3 shows the concrete bit complexity of solving the instance via the reduction approach as a function of the erasure rate in comparison to the bruteforce approach for NIST category I and V parameters. Note that for sake of clarity we restrict to the MiRitH "a" parameter sets. However, note that, the "b" parameters lead very similar results. We observe that in contrast to the bruteforce approach, which yields improvements only up to a certain upper bound, the application of the reduction leads to a steady improvement for any erasure rate $\delta = |I|/(dk) < 1$.

In Table 11 we provide bounds on the erasure rate, up to which the secret key can be recovered in less than 2^{60} and 2^{80} bit operations using the brute force approach and the



(a) MIRA-I parameters

(b) MIRA-V parameters

(c) MiRitH-Ia parameters

(d) MiRitH-Va parameters

Fig. 3 Bit complexity of the erasure-enhanced Kernel Search attack (Algorithm 4) and the bruteforce approach (dashed) applied to the MIRA-I, MIRA-V, MiRitH-Ia and MiRitH-Va parameter sets

reduction approach. Notably, for many parameter sets, the tolerable erasure rate increases significantly, by a factor of approximately three.

Attack in the Error Model In Appendix B we give for completeness a translation of the erasure-enhanced Kernel-Search algorithm to the error setting via the general translation also used in Sect. 3.4.

5 Partial exposure of the annihilator polynomial

In this section, we consider a partial key exposure attack on the witness used in the MPC protocol of RYDE and MIRA signature schemes. The witness in both schemes is an annihilator polynomial of a linear subspace directly connected to the solution. In the case of RYDE this subspace is generated by the entries of the solution vector, while in MIRA the subspace is generated by the columns of the low-rank matrix that is a linear combinations of the input

Table 11 Maximum erasure rates that allow for a key recovery with less than 2^{60} and 2^{80} operations

	60-Bit		80-Bit	
	Bruteforce	Algorithm 4	Bruteforce	Algorithm 4
MIRA-I	0.08	0.27	0.13	0.40
MIRA-III	0.05	0.14	0.08	0.24
MIRA-V	0.03	0.10	0.05	0.18
MiRitH-Ia	0.11	0.26	0.18	0.43
MiRitH-IIIa	0.06	0.18	0.11	0.27
MiRitH-Va	0.04	0.11	0.07	0.22

matrices. In both cases, the recovery of this linear subspace, or equivalently the recovery of the full annihilator polynomial, allows one to recover the secret key in polynomial time. Therefore in RYDE as well as MIRA once the subspace is known the public key relation is used to obtain a linear system whose solution is the secret key.

More precisely, in MIRA for a basis $b_1, \ldots, b_r \in \mathbb{F}_q^m$ of the column-space of $\mathbf{E}$, we have $\mathbf{E} = \mathbf{BC}$, where $\mathbf{B} \in \mathbb{F}_q^{m \times r}$ is the matrix having columns $b_1, \ldots, b_r$ and $\mathbf{C} \in \mathbb{F}_q^{r \times n}$. Hence, the relation $\mathbf{E} = \mathbf{M}_0 + \sum_{i=1}^{k} \alpha_i \mathbf{M}_i$ can be rewritten as a linear system over $\mathbb{F}_q$ of mn equations in the $k + mr$ unknowns given by $\alpha_1, \ldots, \alpha_k$ and the entries of $\mathbf{C}$. This linear system can be solved in polynomial time, thus recovering the secret key $\alpha_1, \ldots, \alpha_k$.

In the case of RYDE the recovery works similar to the recovery of the solution in the context of the GRS attack once a suitable subspace is known by solving the linear system $\mathbf{Hx} = \mathbf{s}$ provided by the public key.

Witness-format

The annihilator polynomial in both schemes is stored via its coefficient representation, i.e. the polynomial

$$L_U(X) := \prod_{u \in U} (X - u) = \sum_{i=0}^{r} \beta_i X^{q^i},$$

is represented via its r coefficients β_i, $i = 0, \ldots, r-1$, since $\beta_r = 1$. Both schemes use an r-dimensional subspace $U \subset \mathbb{F}_{q^m}$ with $q = 2^v$, for a positive integer v. The coefficients are then represented on the bitlevel via a basis $b_1, \ldots, b_{mv}$ of $\mathbb{F}_{q^m}$ as a $\mathbb{F}_2$-linear space. That is, each coefficient $\beta_i \in \mathbb{F}_{q^m}$ is represented via the (unique) string of mv bits $\beta_{i,1}, \ldots, \beta_{i,mv} \in \mathbb{F}_2$ such that $\beta_i = \sum_{j=1}^{mv} \beta_{i,j} b_j$. We remark that $b_1, \ldots, b_{mv}$ are determined by the implementation of the scheme.

5.1 Attack in the erasure model

We let $I \subseteq \{0, \ldots, r-1\} \times [mv]$ define the index set of unknown bits of the annihilator polynomial. That is, each bit $\beta_{i,j}$ with $(i, j) \in I$ is unknown, while each bit $\beta_{i,j}$ with $(i, j) \notin I$ is known.

In the following, we show that recovering the unknown coefficients is equivalent to solving a MinRank instance that can be computed in polynomial time from the known coefficients.

Theorem 14 (Annihilator Polynomial Coefficient Exposure) *Retrieving the unknown $\beta_{i,j}$'s with $(i,j) \in I$ is equivalent to solving an instance of the MinRank problem with parameters $q' = 2$, $m' = n' = mv$, $k' = |I|$, $r' = (m-r)v$, which can be computed in polynomial time from the basis $b_1, \ldots, b_{mv}$ and the given coefficient knowledge $(\beta_{k,h})_{(k,h)\notin I}$.*

Proof By construction, we have that U is the set of zeros of L_U. Thus the elements of U are exactly those of the form $\sum_{j=1}^{mv} x_j b_j$, where $x_1, \ldots, x_{mv} \in \mathbb{F}_2$ satisfy

$$L_U\left(\sum_{j=1}^{mv} x_j b_j\right) = \sum_{j=1}^{mv} L_U(b_j)\, x_j = 0. \tag{9}$$

Note that in Eq. (9) we used the fact that L_U is a $\mathbb{F}_q$-linear map and, a fortiori, an $\mathbb{F}_2$-linear map, since $q = 2^v$. For every $j \in [mv]$ we have that

$$\begin{aligned}
L_U(b_j) &= \sum_{k=0}^{r} \beta_k (b_j)^{q^k} \\
&= \sum_{k=0}^{r} \sum_{h=1}^{mv} \beta_{k,h} b_h (b_j)^{q^k} \\
&= \sum_{(k,h)\notin I} \beta_{k,h} b_h (b_j)^{q^k} + \sum_{(k,h)\in I} \beta_{k,h} b_h (b_j)^{q^k} \\
&= \sum_{i=1}^{mv} c_{0,i,j} b_i + \sum_{(k,h)\in I} \beta_{k,h} \sum_{i=1}^{mv} c_{k,h,i,j} b_i \\
&= \sum_{i=1}^{mv} \left(c_{0,i,j} + \sum_{(k,h)\in I} \beta_{k,h} c_{k,h,i,j} \right) b_i,
\end{aligned} \tag{10}$$

where the $c_{0,i,j}$'s and $c_{k,h,i,j}$'s belong to $\mathbb{F}_2$ and are uniquely determined by the identities

$$\sum_{(k,h)\notin I} \beta_{k,h} b_h (b_j)^{q^k} = \sum_{i=1}^{mv} c_{0,i,j} b_i \quad \text{and} \quad b_h (b_j)^{q^k} = \sum_{i=1}^{mv} c_{k,h,i,j} b_i. \tag{11}$$

Note that, since the basis $b_1, \ldots, b_{mv}$ and each $\beta_{k,h}$ with $(k,h) \notin I$ is known, the left-hand sides of both equations in (11) are known elements in $\mathbb{F}_{q^m}$. Hence, all $c_{0,i,j}$'s and $c_{k,h,i,j}$'s can be computed in polynomial time.

Therefore, employing Eq. (10), we get that Eq. (9) is equivalent to

$$\sum_{j=1}^{mv} \sum_{i=1}^{mv} \left(c_{0,i,j} + \sum_{(k,h)\in I} \beta_{k,h} c_{k,h,i,j} \right) b_i x_j = 0,$$

that is,

$$\sum_{i=1}^{mv} \left(\sum_{j=1}^{mv} \left(c_{0,i,j} + \sum_{(k,h)\in I} \beta_{k,h} c_{k,h,i,j} \right) x_j \right) b_i = 0. \tag{12}$$

Since $b_1, \ldots, b_{mv}$ form a basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_2$, and hence the b_i's are $\mathbb{F}_2$-linearly independent, we get that Eq. (12) is equivalent to

$$\sum_{j=1}^{mv} \left(c_{0,i,j} + \sum_{(k,h)\in I} \beta_{k,h} c_{k,h,i,j} \right) x_j = 0, \quad \text{for all} \quad i \in [mv]. \tag{13}$$

Let $\mathbf{C}_0 = (c_{0,i,j})_{i,j}$ and $\mathbf{C}_{k,h} = (c_{k,h,i,j})_{i,j}$, with $(k,h) \in I$, be matrices in $\mathbb{F}_2^{mv\times mv}$. Then, the linear system of Eq. (12) can be written in matrix form as

$$\left(\mathbf{C}_0 + \sum_{(k,h)\in I} \beta_{k,h} \mathbf{C}_{k,h} \right) \mathbf{x} = \mathbf{0}, \tag{14}$$

where $\mathbf{x} = (x_j)_j$ is a vector in $\mathbb{F}_q^{mv\times 1}$.

Thus U is the set of solutions $\mathbf{x}$ to the linear system defined by Eq. (14). Since U has dimension r over $\mathbb{F}_q$, the linear system has a total of $|U| = q^r = 2^{rv}$ solutions, which implies that the $\mathbb{F}_2$-rank of the linear system is

$$\operatorname{rank}\left(\mathbf{C}_0 + \sum_{(k,h)\in I} \beta_{k,h} \mathbf{C}_{k,h} \right) = mv - rv.$$

This proves that the known matrices $\mathbf{C}_0, (\mathbf{C}_{k,h})_{k,h}$ form an instance of the MinRank problem with target rank $v(m-r)$, whose solution is formed by the missing bits of the annihilator polynomial's coefficients $(\beta_{k,h})_{(k,h)\in I}$. □

Theorem 14 implies that algorithms solving the MinRank problem can be employed to recover the unknown coefficients of the annihilator polynomial. Technically, the theorem even applies if $I = \{0, \ldots, r-1\} \times [n]$, i.e., in the case where no additional knowledge on the coefficients is provided. However, note that in such a case any annihilator polynomial of an arbitrary r-dimensional subspace forms a solution to the constructed instance. In other words, the resulting MinRank instance has many solutions. More generally, the set of solutions to the constructed MinRank instance is exactly the set of annihilator polynomials compatible with the given bit information. Assuming the annihilator polynomials distribute uniformly, we expect about

$$\frac{\begin{bmatrix} m \\ r \end{bmatrix}_{2^v}}{2^{rmv-|I|}} = \Theta\left(2^{|I|-vr^2}\right)$$

solutions to the constructed instance. Here $rmv - |I|$ is the number of known bits of the coefficients and $\begin{bmatrix} m \\ r \end{bmatrix}_{2^v}$ counts the number of subspaces of $\mathbb{F}_{(2^v)^m}$ of dimension r, or equivalently the number of annihilator polynomials. Note that this number is up to a constant approximated by $2^{rv(m-r)}$ as given by Lemma 1. This implies that as long as $|I| \leq vr^2$ we expect a unique annihilator polynomial to be compatible with the bit information and in turn a unique solution to the MinRank instance.

Proposition 15 (Unique Solution) *The annihilator polynomial is uniquely determined by the given bit information and in turn the MinRank instance constructed in Theorem 14 has a unique solution, as long as* $|I| \leq vr^2$.

Algorithm 5 Algorithm for retrieving the unknown coefficients of L_U.

Annihilator-Recovery$(m, v, r, (b_i)_{i\in[mv]}, I, (\beta_{k,h})_{(k,h)\notin I})$

Input: parameters m, v, r, a basis $(b_i)_{i\in[mv]}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_2$, a set I of unknown coefficients, and set $(\beta_{k,h})_{(k,h)\notin I}$ of known coefficients.

Output: the coefficients $(\beta_{k,h})_{(k,h)}$ such that

$L_U(X) := \prod_{u\in U}(X-u) = \sum_{i=0}^{r}\beta_i X^{q^i}$ with $\beta_i = \sum_{j=1}^{mv}\beta_{i,j}b_j$.

1 : Compute $\mathbf{C}_0 = (c_{0,i,j}) \in \mathbb{F}_2^{mv\times mv}$ such that $\sum_{(k,h)\notin I}\beta_{k,h}b_h(b_j)^{q^k} = \sum_{i=1}^{mv} c_{0,i,j}b_i$ for $j \in [mv]$

2 : Compute $\mathbf{C}_{k,h} = (c_{k,h,i,j})_{i,j} \in \mathbb{F}_2^{mv\times mv}$ for $(k,h) \in I$ such that $b_h(b_j)^{q^k} = \sum_{i=1}^{mv} c_{k,h,i,j}b_i$ for $j \in [mv]$

3 : Solve the MinRank instance $\operatorname{rank}\left(\mathbf{C}_0 + \sum_{(k,h)\in I}\beta_{k,h}\mathbf{C}_{k,h}\right) = mv - rv$ using the kernel search attack.

4 : **return** $(\beta_{k,h})_{(k,h)\in\{0,\dots,r\}\times[mv]}$

5.1.1 Complexity of solving the MinRank instance

For the parameters encountered in the instances derived from Theorem 14 for proposed RYDE and MIRA parameters we use the CryptographicEstimators library to estimate the concrete complexity. For all parameters, we find the Kernel-Search algorithm (see Lemma 12) to obtain the best complexity.

In the following we therefore derive the complexity of the Kernel-Search algorithm applied to the instance constructed in Theorem 14. This procedure to recover the missing coefficients of the annihilator polynomial is summarized in Algorithm 5 and its complexity is given by the following theorem.

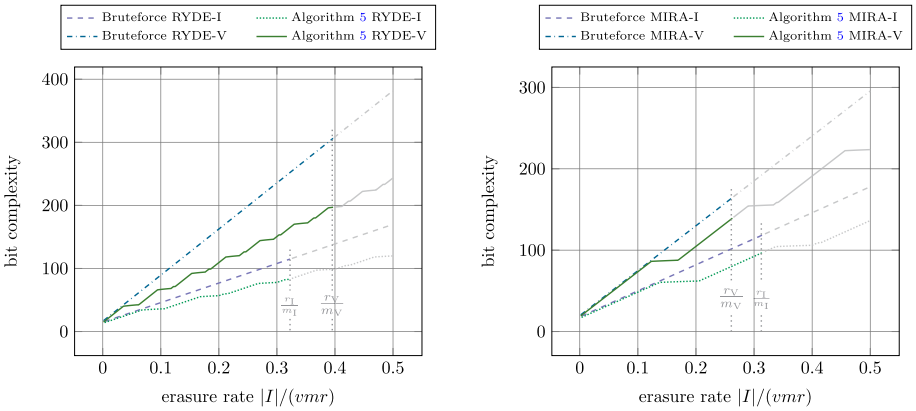
Theorem 16 *Let v, n, m be positive integers and $q = 2^v$. Let $\delta \in (0, \frac{r}{m})$ be the erasure rate. Then Algorithm 5 recovers the coefficients of the annihilator polynomial in the erasure setting in time complexity $\mathcal{O}(2^{v(m-r)\left\lceil\frac{|I|}{mv}\right\rceil}\cdot|I|^3)$, where I is the index set of missing coefficient bits.*

Proof According to Theorem 14 recovering the coefficients is equivalent to solving a Minrank instance with parameters $q' = 2, m' = n' = mv, k' = |I|, r' = (m-r)v$. Now Lemma 12 states the complexity of solving this instance as

$$\mathcal{O}(q'^{r'\lceil\frac{k'}{m'}\rceil}k'^3) = \mathcal{O}(2^{v(m-r)\left\lceil\frac{|I|}{mv}\right\rceil}\cdot|I|^3).$$

Note that $\delta \le \frac{r}{m}$ ensures a unique solution to the MinRank instance according to Proposition 15 and, hence, guarantees that the returned solution reveals the coefficients of the searched annihilator polynomial. □

Note that the asymptotic complexity of solving the MinRank instance given by Theorem 16 is roughly $2^{(1-\frac{r}{m})|I|}$. On the other hand a naive bruteforce of the missing bits of the coefficients has an asymptotic complexity of $2^{|I|}$. Hence, the reduction to the MinRank problem improves the exponent significantly by a factor of $0.6 \le (1-r/m) \le 0.74$, for the suggested parameter sets of RYDE and MIRA.



(a) RYDE-I and RYDE-V (b) MIRA-I and MIRA-V

Fig. 4 Comparison of bruteforce of coefficients (dashed and dash-dotted lines) and recovery of the polynomial using Algorithm 5 (solid and dotted lines). Vertical dotted lines mark bounds to multiple solution regimes

Similar to Sect. 4.1.1, the ceiling of $\frac{k'}{m'}$ in the exponent prevents the running time from becoming polynomial. However, again we find that with respect to concrete complexities there are large regimes for which a recovery of the polynomial remains possible under reasonable runtime constraints.

5.1.2 Concrete complexity of recovering $L_U(x)$

In the following we compare the concrete complexity of recovering the missing bits of the coefficients via Algorithm 5 to a bruteforce of the missing bits. Here we account for the polynomial factors of the bruteforce approach with m^3, which is determining if the current guess of the bits leads to a linearized polynomial that has exactly q^r roots, i.e., is an annihilator polynomial of an r-dimensional subspace.

For estimating the complexity of solving the MinRank instance, we again rely on the CryptographicEstimators library [17] which incorporates state-of-the-art polynomial factor improvements.

In Fig. 4 we illustrate the concrete complexity for the recovery of $L_U(x)$ as a function of erasure rate $\delta = |I|/(rmv)$ for RYDE (left) and MIRA (right). Note that for MIRA we have $v = 4$, since $q = 16 = 2^4$, while for RYDE it holds $v = 1$. Here r_i and m_i refer to the r and m parameter of the i-th parameter set. Recall that Proposition 15 guarantees the annihilator to be uniquely determined by the given bit information up to an erasure rate of $vr^2/(rmv) = r/m$. Past that point multiple solutions to the constructed MinRank instance exist and the running time for both, the bruteforce attack as well as Algorithm 5 would increase. We only provide the complexity plot past that point to indicate the asymptotic scaling.

We give the maximum erasure rates r_i/m_i, which lead to uniquely determined annihilator polynomials for the different parameter sets in Table 12. We also state the concrete bit complexity for recovering the polynomial in those cases, observing large gains of up to 108 bits in the case of RYDE and 41 bits in the case of MIRA. Note that all attacks for the maximum erasure rates stay far below the respective security levels. Table 13 states the maximum erasure rates which allow for a recovery of the missing coefficients using less than 2^{60} or 2^{80} bit operations. We observe that especially in the case of RYDE the attacks are

Table 12 Maximum erasure rates that uniquely determine the polynomial and bit complexity to recover $L_U(x)$ in those cases

	Max erasure rate $\left(\frac{r}{m}\right)$	Bruteforce	Algorithm 5
RYDE-I	0.32	115	83
RYDE-III	0.35	185	131
RYDE-V	0.40	305	197
MIRA-I	0.31	118	96
MIRA-III	0.32	163	122
MIRA-V	0.26	164	138

Table 13 Maximum erasure rates that allow for recovery with less than 2^{60} and 2^{80} operations

	60-Bit		80-Bit	
	Bruteforce	Algorithm 5	Bruteforce	Algorithm 5
RYDE-I	0.15	0.21	0.21	0.31
RYDE-III	0.09	0.12	0.13	0.19
RYDE-V	0.06	0.09	0.09	0.14
MIRA-I	0.13	0.14	0.19	0.26
MIRA-III	0.09	0.09	0.13	0.19
MIRA-V	0.07	0.08	0.11	0.11

specifically effective. In the case of RYDE-I a recovery of the polynomial remains feasible with a bit complexity of 80 for erasure rates smaller or equal to 0.31, which is just slightly less than the maximum rate of $r/m \approx 0.32$ in that case. On MIRA parameters Algorithm 5 also obtains considerable improvements over the bruteforce approach. However, due to the smaller choices of m, n and r resulting from the larger choice of q the attack requires longer to converge to its asymptotic scaling. Generally, this shows that the attack remains most effective for $q = 2$, i.e., for the choice of $v = 1$.

Translation to the error setting For completeness we give a translation of Theorem 16 to the error setting in Appendix C following the generic translation from the erasure to the error setting also used in Sect. 3.4.

A Rank-SD error attack

In this section we provide the proofs of Lemma 6 and Theorem 7 for completeness.

A.1 Attack complexity

Lemma 6 (Error-enhanced GRS attack) *Let $m, n, k, r \in \mathbb{N}$ with $n \geq m$. Let $(\mathbf{H}, \mathbf{s})$ be a Rank-SD instance over $\mathbb{F}_{2^m}$ with solution $\mathbf{x}$. Given an erroneous version $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{e}$ of $\mathbf{x}$ with error rate δ, Algorithm 2 recovers $\mathbf{x}$ under Heuristic 1 with expected time complexity*

$$T = \mathcal{O}\left(\frac{\binom{mn}{t}}{\binom{(1-\delta)mn}{t}} \cdot \left((n-k)m+t\right)^3 \cdot \max\left(1, 2^{r\left\lceil \frac{(k+1)m-t}{n}\right\rceil - m}\right)\right),$$

for any integer $t \leq (1-\delta)\cdot mn$.

Proof The complexity of Algorithm 2 is equal to the complexity of one execution of the loop divided by the success probability of a single iteration. One iteration is dominated by the execution of Algorithm 1 in line 4. From Theorem 3 we know that the complexity of Algorithm 1 is

$$T = \mathcal{O}\left(\left((n-k)m+t\right)^3 \cdot \max\left(1, 2^{r\left\lceil \frac{(k+1)m-t}{n}\right\rceil - m}\right)\right).$$

This execution of Algorithm 1 returns the solution $\mathbf{x}$ whenever the guess of the t coordinates made in line 3 of Algorithm 2 is correct. Since there are $\delta \cdot mn$ errors, i.e., non-zero entries, in $\mathbf{e}$, we have

$$s = \Pr\left[\text{guess } t \text{ zeros correctly}\right] = \frac{\binom{(1-\delta)mn}{t}}{\binom{mn}{t}}.$$

Here the denominator counts the possibilities to make the t zero guesses among all mn coordinates, while the numerator restricts the guesses to be among the non-zero coordinates. Finally, this leads to an expected number of s^{-1} iterations of the loop until we expect at least one correct guess, which allows to conclude that the expected time complexity is

$$T = \mathcal{O}\left(\frac{\binom{mn}{t}}{\binom{(1-\delta)mn}{t}} \cdot \left((n-k)m+t\right)^3 \cdot \max\left(1, 2^{r\left\lceil \frac{(k+1)m-t}{n}\right\rceil - m}\right)\right),$$

for any $t \leq (1-\delta)mn$. □

A.2 Polynomial error attack

In the following we deliver the proof for Theorem 7. That is, we show that Algorithm 2 runs in polynomial time, whenever $\delta = \mathcal{O}(\frac{\log mn}{mn})$. Note that this is essentially the same result obtained for codes in the Hamming metric in [16]. This is explained by the fact that the time, once we ensure that Algorithm 1 runs in polynomial time, is dominated by guessing a constant fraction $t = c \cdot mn$ of zeros contained in $\mathbf{e}$. Asymptotically speaking this is the same as applying information set decoding in the Hamming, rather than rank metric. Before we prove the theorem, we recall the following Lemma, which follows from [16, Theorem 2.1].

Lemma 17 *Let* n, k, δ *be positive integers such that* $n - k = c \cdot n$, *for a positive constant* c. *Then if* $\delta = \mathcal{O}(\log n)$ *we have that*

$$\frac{\binom{n}{\delta}}{\binom{n-k}{\delta}} = n^{\mathcal{O}(1)}.$$

We restate the theorem for convenience.

Theorem 7 *(Polynomial Error Attack) Let* $n, m, k, r \in \mathbb{N}$ *with* $n \geq m$. *Let* $(\mathbf{H}, \mathbf{s})$ *be a Rank-SD instance over* $\mathbb{F}_{2^m}$ *with solution* $\mathbf{x}$. *Given an erroneous version* $\tilde{\mathbf{x}}$ *of* $\mathbf{x}$ *with error probability* $p = \mathcal{O}(\frac{\log(mn)}{mn})$, *then* $\mathbf{x}$ *can be recovered in polynomial time with constant probability.*

Proof Note that once we choose a number t of zeros to guess in Algorithm 2 that leads to a polynomial runtime of Algorithm 1 in line 4, the running time of Algorithm 2 (compare to Theorem 6) becomes (up to polynomial factors)

$$\frac{\binom{mn}{t}}{\binom{(1-\delta)mn}{t}} = \frac{\binom{mn}{\delta \cdot mn}}{\binom{mn-t}{\delta \cdot mn}} \leq \frac{\binom{mn}{t}}{\binom{mn-t}{\log mn}}.$$

Notice that according to Lemma 17 this term is polynomial as long as $mn - t$ or equivalently t is a constant fraction of mn.

From the proof of Theorem 4, we have that Algorithm 1 runs in polynomial time whenever

$$t \geq (k+1)m + \varepsilon n - \frac{mn}{r},$$

where $0 \leq \varepsilon \leq 1$. Therefore, we choose t equal to this bound which gives

$$\frac{t}{mn} = \frac{k+1}{n} + \frac{\varepsilon}{m} - \frac{1}{r} = c \pm o(1),$$

where c is a constant, since we only consider codes with constant rate k/n (compare to Definition 4). □

B Error-enhanced kernel-search attack

In the following we apply the generic translation from the erasure to the error setting, already applied in Algorithm 2, to obtain an attack in case an erroneous version of the secret key is provided. Recall that such an erroneous version is defined as $\boldsymbol{\alpha} = \boldsymbol{\alpha} + \mathbf{e}$, where $\boldsymbol{\alpha}$ is the bit representation of the secret key and $\mathbf{e} \in \mathbb{F}_2^{dk}$ an error following a sparse distribution.

The translation then works by guessing t zero positions in $\mathbf{e}$, revealing t bits of the secret key. By treating all other coordinates as erased, this allows the application of Algorithm 4 to recover the full secret key. The application of Algorithm 4 is successful whenever the guess for the t zero positions is correct, which happens with probability

$$\frac{\binom{dk}{t}}{\binom{(1-\delta)dk}{t}}.$$

Accordingly, we obtain the following complexity for recovering the secret key from an erroneous version.

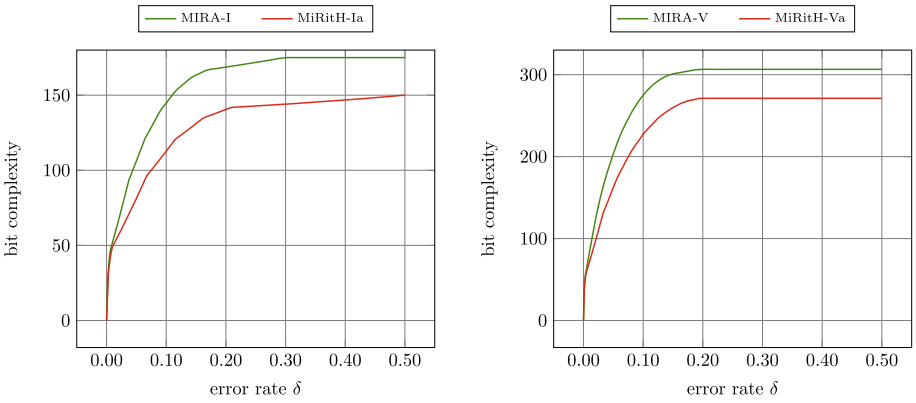
Lemma 18 (Error-enhanced Kernel Search attack) *Let $m, n, k, r \in \mathbb{N}$. Let $(\mathbf{M}_0, \mathbf{M}_1, \ldots, \mathbf{M}_k)$ be a MinRank instance over $\mathbb{F}_{2^d}$ with solution $\boldsymbol{\alpha}$. Given an erroneous version $\tilde{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \mathbf{e}$ with error rate δ, the secret key α can be recovered in expected time complexity*

$$T = \mathcal{O}\left(\frac{\binom{dk}{t}}{\binom{(1-\delta)dk}{t}} \cdot 2^{dr\left\lceil \frac{dk-t}{dm} \right\rceil} \cdot (dk-t)^w \right),$$

for any integer $t \leq (1-\delta)dk$.

B.1 Concrete complexity of error attacks

In Fig. 5 we provide the concrete complexities of recovering the secret key in the error setting for MIRA and MiRitH for NIST category I and V parameter sets.



(a) Level I parameters. (b) Level V parameters.

Fig. 5 Bit complexity from Lemma 18 for the Level I and V parameters of MIRA and MiRitH

Table 14 Maximum error rates that allow for a key recovery with less than 2^{143}, 2^{207} and 2^{272} operations for MIRA and MiRitH for levels I, III and V respectively

	Error rate
MIRA-I	0.096
MIRA-III	0.113
MIRA-V	0.097
MiRitH-Ia	0.256
MiRitH-IIIa	0.275
MiRitH-Va	0.196

In Table 14 we summarize the error rates that allow for key recovery below the respective security levels, implying that up to these error rates the leaked information can be exploited to improve over generic key recovery attacks.

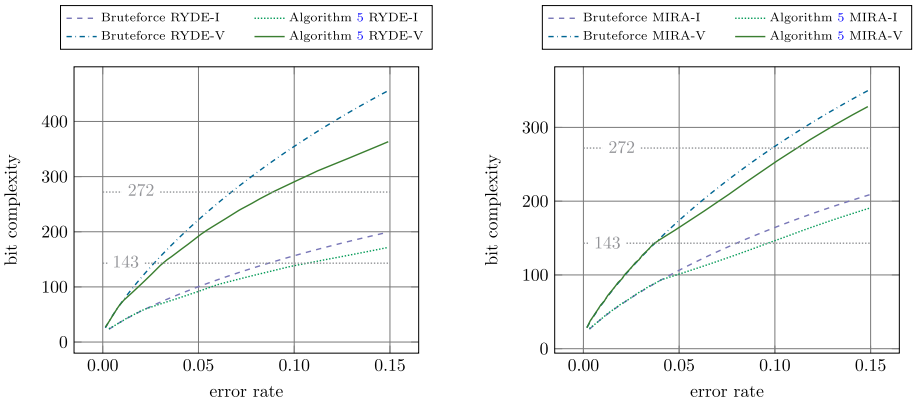
C Annihilator polynomial recovery in the error setting

Similar to Sect. 3.4 it is possible to translate the attacks from the erasure setting in the previous section to the error setting. Therefore first, given the erroneous key material $(\tilde{\beta}_{i,j})_{i,j}$, we guess an index set J_1 with $|J_1| = t_1$, such that $\tilde{\beta}_{i,j} = \beta_{i,j}$ is error-free for $(i,j) \in J_1$, as well as an index set J_2 with $|J_2| = t_2$, such that $\tilde{\beta}_{i,j} = \beta_{i,j} + 1$ is erroneous for $(i,j) \in J_2$. Subsequently, we apply Algorithm 5 with $I = \{0, r-1\} \times [mv] \setminus J_1 \cup J_2$ and the corresponding $\tilde{\beta}_{i,j}$, $(i,j) \in J_1$ and $\tilde{\beta}_{i,j} + 1$, $(i,j) \in J_2$.

We summarize the complexity of this approach in the following lemma.

Lemma 19 (Recovery of $L_U(x)$ in the Error Setting) *Let v, n, m be positive integers and $q = 2^v$. Let $\delta \in (0,1)$ be the error rate. Then the annihilator polynomial can be recovered from an erroneous version $(\tilde{\beta}_{i,j} = \beta_{i,j} + e_{i,j})_{i,j}$ in time complexity*

$$\mathcal{O}\left(\frac{\binom{rmv}{t_1}\binom{rmv-t_1}{t_2}}{\binom{(1-\delta)rmv}{t_1}\binom{\delta rmv}{t_2}} \cdot 2^{v(m-r)\left\lceil \frac{rmv-t_1-t_2}{mv} \right\rceil} \cdot (rmv - t_1 - t_2)^3 \right),$$



(a) RYDE-I and RYDE-V (b) MIRA-I and MIRA-V

Fig. 6 Comparison of bruteforce of coefficients (dashed and dash-dotted lines) and recovery of the polynomial using guessing strategy in combination with Algorithm 5 (solid and dotted lines)

Table 15 Maximum error rates that allow for a key recovery with less than 2^{143}, 2^{207} and 2^{272} operations for RYDE / MIRA-I, -III and -V respectively

Parameter set	RYDE		MIRA	
	Bruteforce	Lemma 19	Bruteforce	Lemma 19
NIST-I	0.087	0.106	0.081	0.097
NIST-III	0.081	0.104	0.086	0.101
NIST-V	0.067	0.089	0.098	0.111

for any $t_1 \leq (1-\delta) \cdot rmv$, $t_2 \leq \delta \cdot rmv$ *and* $t_1 + t_2 \geq rv(m-r)$.

Proof Note that the running time of Algorithm 5 on input a set I with $|I| = rmv - t_1 - t_2$ is given by Theorem 16 as

$$\mathcal{O}\left(2^{v(m-r)\left\lceil \frac{rmv-t_1-t_2}{mv} \right\rceil} \cdot (rmv - t_1 - t_2)^3\right),$$

Further note that $t_1 + t_2 \geq rv(m-r)$, ensures $|I|/rmv \leq r/m$ as required by Theorem 16. The probability of making a correct guess for the sets J_1 and J_2 with $|J_i| = t_i$ is

$$s = \Pr\left[\forall(i,j) \in J_1 \colon \tilde{\beta}_{i,j} = \beta_{i,j} \wedge \forall(i,j) \in J_2 \colon \tilde{\beta}_{i,j} = \beta_{i,j} + 1\right] = \frac{\binom{(1-\delta)rmv}{t_1}\binom{\delta rmv}{t_2}}{\binom{rmv}{t_1}\binom{rmv-t_1}{t_2}}.$$

This implies s^{-1} guesses on expectation, which results in the claimed time complexity. □

In Fig. 6 we compare the concrete complexity of this translation to the bruteforce of the error positions. While the adaptation generally yields an improvement over the naive bruteforce, especially for smaller error rates, the complexities remain close. Overall, the error attacks remain favorable to a direct key recovery up to error rates of about 10% in the case of RYDE and up to 11% in the case of MIRA (see Table 15).

D Gaussian coefficients approximation

Lemma 20 *For every prime power q there exists a constant $C_q \in (0,1)$ such that*

$$C_q < \frac{1}{q^{r(m-r)}} \begin{bmatrix} m \\ r \end{bmatrix}_q < C_q^{-1}.$$

Proof We have that

$$\begin{aligned} \begin{bmatrix} m \\ r \end{bmatrix}_q &:= \prod_{i=1}^{r} \frac{q^{m-i+1}-1}{q^i-1} = \prod_{i=1}^{r} \left(q^{m-2i+1} \cdot \frac{1-q^{-m+i-1}}{1-q^{-i}} \right) \\ &= q^{mr-r(r+1)+r} P = q^{r(m-r)} P, \end{aligned}$$

where

$$P := \prod_{i=1}^{r} \frac{1-q^{-m+i-1}}{1-q^{-i}} \geq \prod_{i=1}^{r} (1-q^{-m+i-1}) > \prod_{j=1}^{\infty} (1-q^{-j}) =: C_q \in (0,1)$$

and the infinite product converges since $\sum_{j=1}^{\infty} q^{-j} < +\infty$. Furthermore, it holds

$$P \leq \prod_{i=1}^{r} \frac{1}{1-q^{-i}} < \prod_{i=1}^{\infty} \frac{1}{1-q^{-i}} =: C_q^{-1}.$$

The claim follows. □

Acknowledgements G. D'Alconzo, A. Gangemi and C. Sanna are members of GNSAGA of INdAM. They are also members of CrypTO, the group of Cryptography and Number Theory of the Politecnico di Torino C. Sanna was partially supported by the Italian Ministry of University and Research in the framework of the Call for Proposals for scrolling of final rankings of the PRIN 2022 call - Protocol no. 2022RFAZCJ. This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union—NextGenerationEU. A. Esser is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID MA 2536/12.

Author Contributions All authors contributed equally to this work.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement. This work was partially supported by Project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union—NextGenerationEU. A. Esser is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—Project-ID MA 2536/12.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethical approval Not applicable.

Materials availability Not applicable.

Code availability Not applicable.

References

1. Adj G., Barbero S., Bellini E., Esser A., Rivera-Zamarripa L., Sanna C., Verbel J., Zweydinger F.: MiRitH specification (2023). https://pqc-mirith.org/assets/downloads/mirith_specifications_v1.0.0.pdf.
2. Adj G., Aragon N., Barbero S., Bardet M., Bellini E., Bidoux L., Chi-Domínguez J.-J., Dyseryn V., Esser A., Feneuil T., Gaborit P., Neveu R., Rivain M., Rivera-Zamarripa L., Sanna C., Tillich J.-P., Verbel J., Zweydinger F.: MIRATH website (2024). https://pqc-mirath.org.
3. Alagic G., et al.: Status report on the first round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process. Technical report, NIST (October 2024). https://doi.org/10.6028/NIST.IR.8528.
4. Aragon N., Gaborit P., Hauteville A., Tillich J.-P.: A new algorithm for solving the rank syndrome decoding problem. In: 2018 IEEE International Symposium on Information Theory (ISIT), pp. 2421–2425. IEEE (2018).
5. Aragon N., Barreto P., Bettaieb S., Bidoux L., Blazy O., Deneuville J.-C., Gaborit P., Ghosh S., Gueron S., Güneysu T., et al.: BIKE: bit flipping key encapsulation. HAL Open Sci. (2022).
6. Aragon N., Bardet M., Bidoux L., Chi-Domínguez J.-J., Dyseryn V., Feneuil T., Gaborit P., Joux A., Rivain M., Tillich J.-P., Vinçotte A.: RYDE specification (2023). https://pqc-ryde.org/assets/downloads/ryde_spec.pdf.
7. Aragon N., Bardet M., Bidoux L., Chi-Domínguez J.-J., Dyseryn V., Feneuil T., Gaborit P., Neveu R., Rivain M., Tillich J.-P.: MIRA specification (2023). https://pqc-mira.org/assets/downloads/mira_spec.pdf.
8. Baum C., Braun L., Saint Guilhem C.D., Klooß M., Orsini E., Roy L., Scholl P.: Publicly verifiable zero-knowledge and post-quantum signatures from vole-in-the-head. In: Annual International Cryptology Conference, pp. 581–615. Springer, New York (2023).
9. Chen C., Danba O., Hoffstein J., Hülsing A., Rijneveld J., Schanck J.M., Schwabe P., Whyte W., Zhang Z.: NTRU Algorithm Specifications and Supporting Documentation. Brown University and Onboard Security Company, Wilmington (2019).
10. Coppersmith D.: Small solutions to polynomial equations, and low exponent RSA vulnerabilities. J. Cryptol. **10**(4), 233–260 (1997). https://doi.org/10.1007/s001459900030.
11. Dachman-Soled D., Gong H., Kulkarni M., Shahverdi A.: (In) security of ring-LWE under partial key exposure. J. Math. Cryptol. **15**(1), 72–86 (2020).
12. Ding J., Chen M.-S., Petzoldt A., Schmidt D., Yang B.-Y., Kannwischer M., Patarin J.: Rainbow-algorithm specification and documentation. Specification document of NIST PQC 2nd round submission package (2019).
13. Dumer I.: On minimum distance decoding of linear codes. In: Proceedings of the Fifth Joint Soviet–Swedish International Workshop on Information Theory, pp. 50–52. Moscow (1991).
14. Ernst M., Jochemsz E., May A., Weger B.: Partial key exposure attacks on RSA up to full size exponents. In: Cramer R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 371–386. Springer, New York (2005). https://doi.org/10.1007/11426639_22.
15. Esser A., Bellini E.: Syndrome decoding estimator. In: Hanaoka G., Shikata J., Watanabe Y. (eds.) PKC 2022, Part I. LNCS, vol. 13177, pp. 112–141. Springer, New York (2022). https://doi.org/10.1007/978-3-030-97121-2_5.
16. Esser A., May A., Verbel J.A., Wen W.: Partial key exposure attacks on BIKE, rainbow and NTRU. In: Dodis Y., Shrimpton T. (eds.) CRYPTO 2022, Part III. LNCS, vol. 13509, pp. 346–375. Springer, New York (2022). https://doi.org/10.1007/978-3-031-15982-4_12.
17. Esser A., Verbel J., Zweydinger F., Bellini E.: SoK: CryptographicEstimators—a software library for cryptographic hardness estimation. In: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security, pp. 560–574 (2024).

18. Fiat A., Shamir A.: How to prove yourself: practical solutions to identification and signature problems. In: Conference on the Theory and Application of Cryptographic Techniques, pp. 186–194. Springer, New York (1986).
19. Gaborit P., Ruatta O., Schrek J.: On the complexity of the rank syndrome decoding problem. IEEE Trans. Inf. Theory **62**(2), 1006–1019 (2015).
20. Gennaro R.: An improved pseudo-random generator based on the discrete logarithm problem. J. Cryptol. **18**(2), 91–110 (2005). https://doi.org/10.1007/s00145-004-0215-y.
21. Goubin L., Courtois N.T.: Cryptanalysis of the TTM cryptosystem. In: Advances in Cryptology–ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings, vol. 6, pp. 44–57. Springer, New York (2000).
22. Heninger N., Shacham H.: Reconstructing RSA private keys from random key bits. In: Halevi S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 1–17. Springer, New York (2009). https://doi.org/10.1007/978-3-642-03356-8_1.
23. Ishai Y., Kushilevitz E., Ostrovsky R., Sahai A.: Zero-knowledge from secure multiparty computation. In: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, pp. 21–30 (2007).
24. Kirshanova E., May A.: Decoding McEliece with a hint—secret Goppa key parts reveal everything. In: International Conference on Security and Cryptography for Networks, pp. 3–20. Springer, New York (2022).
25. May A., Nowakowski J., Sarkar S.: Partial key exposure attack on short secret exponent CRT-RSA. In: Tibouchi M., Wang H. (eds.) ASIACRYPT 2021, Part I. LNCS, vol. 13090, pp. 99–129. Springer, New York (2021). https://doi.org/10.1007/978-3-030-92062-3_4.
26. May A., Nowakowski J., Sarkar S.: Approximate divisor multiples—factoring with only a third of the secret CRT-exponents. In: Dunkelman O., Dziembowski S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 147–167. Springer, New York (2022). https://doi.org/10.1007/978-3-031-07082-2_6.
27. McEliece R.J.: A public-key cryptosystem based on algebraic coding theory. DSN Prog. Rep. **42–44**, 114–116 (1978).
28. Niederreiter H.: Knapsack-type cryptosystems and algebraic coding theory. Probab. Control Inf. Theory **15**(2), 157–166 (1986).
29. Ore O.: On a special class of polynomials. Trans. Am. Math. Soc. **35**(3), 559–584 (1933).
30. Patel S., Sundaram G.S.: An efficient discrete log pseudo random generator. In: Krawczyk H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 304–317. Springer, New York (1998). https://doi.org/10.1007/BFb0055737.
31. Paterson K.G., Villanueva-Polanco R.: Cold boot attacks on NTRU. In: Patra A., Smart N.P. (eds.) INDOCRYPT 2017. LNCS, vol. 10698, pp. 107–125. Springer, New York (2017).
32. Takayasu A., Kunihiro N.: Partial key exposure attacks on RSA: achieving the Boneh-Durfee bound. In: Joux A., Youssef A.M. (eds.) SAC 2014. LNCS, vol. 8781, pp. 345–362. Springer, New York (2014). https://doi.org/10.1007/978-3-319-13051-4_21.
33. Villanueva-Polanco R.: Cold boot attacks on Bliss. In: Schwabe P., Thériault N. (eds.) LATINCRYPT 2019. LNCS, vol. 11774, pp. 40–61. Springer, New York (2019). https://doi.org/10.1007/978-3-030-30530-7_3.
34. Villanueva-Polanco R.: Cold boot attacks on post-quantum schemes. PhD thesis, Royal Holloway, University of London (2019).
35. Villanueva-Polanco R.: Cold boot attacks on LUOV. Appl. Sci. **10**(12), 4106 (2020).