The Partial-Inverse Approach to Linearized Polynomials and Gabidulin Codes With Applications to Network Coding

Jiun-Hung Yu, *Member, IEEE*, and Hans-Andrea Loeliger, *Fellow, IEEE*

***Abstract*— This paper introduces the partial-inverse problem for linearized polynomials and develops its application to decoding Gabidulin codes and lifted Gabidulin codes in linear random network coding. The proposed approach is a natural generalization of its counterpart for ordinary polynomials, thus providing a unified perspective on Reed–Solomon codes for the Hamming metric and for the rank metric. The basic algorithm for solving the partial-inverse problem is a common parent algorithm of a Berlekamp–Massey algorithm, a Euclidean algorithm, and yet another algorithm, all of which are obtained as easy variations of the basic algorithm. Decoding Gabidulin codes can be reduced to the partial-inverse problem via a key equation with a new converse. This paper also develops new algorithms for interpolating crisscross erasures and for joint decoding of errors, erasures, and deviations in random network coding.**

***Index Terms*— Gabidulin codes, key equation, partial-inverse problem, partial-inverse algorithm, Euclidean algorithm, Berlekamp–Massey algorithm.**

I. Introduction

GABIDULIN codes [2], [3], [4] and related codes [5], [6], [7], [8], [9], [10] have recently received much attention due to interesting applications in network coding [11], [12], [13], [14], [15], [16], [17]. Gabidulin codes also have applications in cryptography and space-time coding [18], [19], [20], [21]. Such codes may be viewed as Reed–Solomon codes [22] over linearized polynomials [23], using the rank metric [2], [3], [4], [5], [6], [7], [8] instead of the Hamming metric. Moreover, linearized Reed–Solomon codes [9], [10] with sum-rank metric [24] are natural hybrids between Reed–Solomon codes and Gabidulin codes, and the sum-rank metric is a hybrid between Hamming metric and rank metric [9], [10]. Similar code constructions have also been proposed with skew polynomials [25], cf., e.g.,[9], [26], [27], and [28]; the sum-rank metric is also considered in, e.g., [29], [30], [31], [32], [33], [34], and [35].

In this paper, we are primarily interested in Gabidulin codes and lifted Gabidulin codes with the rank metric [2], [3], [4], [7]. Many of the decoding algorithms for these codes are inspired by the ones for Reed–Solomon codes. E.g., in the pioneering work [3], a key equation for Gabidulin codes is derived, and a right Euclidean algorithm for linearized polynomials is applied to the key equation; this algorithm may be seen as a generalization of Sugiyama's algorithm [36] from the Hamming metric to the rank metric. In [37], a transformed key equation is formulated, and solved by the Euclidean algorithm; the resulting algorithm is a generalization of Shiozaki–Gao's algorithm [38], [39], see also [30]. Similar equations are also developed in [7] and [8], which can be solved by the shift-register synthesis algorithm for linearized polynomials [40], [41], [42], which in turn is a generalization of the Berlekamp–Massey algorithm [43], [44]. Yet another well-known algorithm is the polynomial-reconstruction-based decoding algorithm [45], which can be seen as the rank-metric analog of Welch–Berlekamp algorithm [46].

All of these decoding algorithms can correct errors up to half the minimum rank distance, and most of them have complexity $O(n^2)$ (in terms of finite-field operations, as a function of the block length n).

For completeness, we also mention a body of work on interleaved Gabidulin codes, e.g., [30], [47], [48], [49], and [50]. However, the development of the partial-inverse approach for interleaved Gabidulin codes is not addressed in the present paper.

Throughout this prior work, both the Euclidean algorithm and the Berlekamp–Massey algorithm play key roles. However, for these algorithms, the transition from classical Reed–Solomon codes (with the Hamming metric) to Gabidulin codes (with the rank metric) is technically far from trivial since linearized polynomials form a non-commutative ring.

For classical Reed–Solomon codes (for the Hamming metric), it is well known that the Berlekamp–Massey algorithm [43], [44] and the Euclidean decoding algorithm [36], [38], [39] are related, and nontrivial translations were given in [56], [57], [58], and [59]. More recently, the partial-inverse approach [51], [52] derives (versions of) these two algorithms as specializations or easy variations of a common parent algorithm. In addition, the partial-inverse approach offers a

Manuscript received 29 July 2022; revised 9 November 2022; accepted 10 January 2023. Date of publication 13 January 2023; date of current version 19 May 2023. The work of Jiun-Hung Yu was supported in part by the Ministry of Science and Technology of Taiwan under Grant MOST111-2218-E-A49-024. An earlier version of this paper was presented in part at the 2019 IEEE International Symposium on Information Theory [DOI: 10.1109/ISIT.2019.8849588]. *(Corresponding author: Jiun-Hung Yu.)*

Jiun-Hung Yu is with the Department of Electronics and Electrical Engineering, Institute of Communications Engineering, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: yuji@nycu.edu.tw).

Hans-Andrea Loeliger is with the Department of Information Technology and Electrical Engineering, ETH Zürich, 8092 Zürich, Switzerland.

Communicated by A.-L. Horlemann-Trautmann, Associate Editor for Coding and Decoding.

Digital Object Identifier 10.1109/TIT.2023.3236720

new interface—the partial-inverse problem—between these algorithms and their applications to decoding. In this paper, we develop the generalization of the partial-inverse approach to linearized polynomials and to the decoding of Gabidulin codes. We will see that, with the proper setup, much of [51] and [52] generalizes naturally, but new arguments are nonetheless needed at many points.

The paper is structured as follows. In Section II, we establish notation and summarize the required basics of linearized polynomials. In Section III, we define the partial-inverse problem for linearized polynomials, and we propose and prove the basic partial-inverse algorithm for its solution; the proof is new and simpler than the proof in [52]. In Section IV, we specialize the partial-inverse algorithm to (a version of) the Berlekamp–Massey algorithm, to (a version of) the Euclidean algorithm, and to yet another algorithm (the quotient-saving algorithm). In Section V, we recall the evaluation transform and connect it with the rank weight. In Section VI, we recall the definition of Gabidulin codes, and propose a key equation with a new converse. It will then be clear that decoding up to half the minimum rank distance is a partial-inverse problem. We also generalize multiply-divide interpolation from [51] and [52] to Gabidulin codes. In Section VII, in preparation for Section VIII, we propose new interpolation methods for different forms of erasures, viz., erased rows, erased columns, and combinations thereof. In Section VIII, we propose new methods for joint decoding of errors, erasures, and deviations in network coding.

Some of the proofs are given in the appendices, which contain also additional material. In Appendix B, we address minimal partial inverses and further properties of the partial-inverse problem. Finally, in Appendix C (using results from Appendix B), we show that every partial-inverse problem can be transformed into an equivalent partial-inverse problem with a monomial modulus, to which the Berlekamp–Massey algorithm can be applied. The proof involves new arguments since the proof of the corresponding fact in [52, Theorem 2] does not seem to generalize to linearized polynomials.

II. Notation and Preliminaries

In this section, we establish notation and give the required basics of linearized polynomials, cf. [6], [23], [43], and [65].

A. Basics of Linearized Polynomials

Let F_q be a finite field with q elements, let L be a positive integer, and let F_{q^L} be an extension field of F_q. For fixed q and L, a linearized polynomial is a polynomial of the form

$$a(x) = \sum_{\ell=0}^{n} a_\ell x^{q^\ell} \tag{1}$$

with $a_\ell \in F_{q^L}$, $\ell = 0, 1, \ldots, n$. The sum of two linearized polynomials $a(x)$ and $b(x)$ is a linearized polynomial, but the ordinary product $a(x)b(x)$ is not, in general, a linearized polynomial. Instead, the composition of two linearized polynomials

$$a(x) \circ b(x) \triangleq a\big(b(x)\big) \tag{2}$$

is always a linearized polynomial [23]. Note that, in general, $a(x) \circ b(x) \neq b(x) \circ a(x)$.

Indeed, with ordinary addition and with the composition (2) as "multiplication", the set of linearized polynomials (for fixed q and L) forms a non-commutative ring with multiplicative identity x. Throughout the paper, we denote this ring by $F[x]_\circ$.

For nonzero $a(x)$, $b(x) \in F[x]_\circ$ with $\deg a(x) = q^n$ and $\deg b(x) = q^m$, the degree of $a(x) \circ b(x)$ is q^{n+m}. The q-degree of $a(x)$ with $\deg a(x) = q^n$ is $\deg_q a(x) = n$, and $\deg_q a(x) \circ b(x) = \deg_q a(x) + \deg_q b(x)$. The leading coefficient of $a(x)$ will be denoted by $\operatorname{lcf} a(x)$, and $\operatorname{lcf} 0 \triangleq 0$.

For nonzero polynomials $a(x), b(x) \in F[x]_\circ$, there exist unique $g(x)$ and $r(x) \in F[x]_\circ$ such that $a(x) = g(x) \circ b(x) + r(x)$ with $\deg_q r(x) < \deg_q b(x)$. We refer to the division as *right division* of $a(x)$ by $b(x)$, and we denote the quotient polynomial $g(x)$ by $a(x) \operatorname{rdiv}_\circ b(x)$, and the remainder polynomial $r(x)$ by $a(x) \operatorname{rmod}_\circ b(x)$. A nonzero polynomial $\in F[x]_\circ$ of the largest degree that *right* divides both $a(x)$ and $b(x)$ will be denoted by $\operatorname{rgcd}(a(x), b(x))$, which can be found by the right Euclidean algorithm for linearized polynomials [23].

Analogously, for any nonzero $a(x), b(x) \in F[x]_\circ$, there exist unique $g(x), r(x) \in F[x]_\circ$ such that $a(x) = b(x) \circ g(x) + r(x)$ with $\deg_q r(x) < \deg_q b(x)$. The quotient polynomials $g(x)$ will be denoted by $a(x) \operatorname{ldiv}_\circ b(x)$, and the remainder $r(x)$ will be denoted by $a(x) \operatorname{lmod}_\circ b(x)$. A nonzero polynomial $\in F[x]_\circ$ of the largest degree that *left* divides both $a(x)$ and $b(x)$ will be denoted by $\operatorname{lgcd}(a(x), b(x))$.

Following a standard convention, we define the notation $x^{[\ell]} \triangleq x^{q^\ell}$. The polynomial in (1) can then be written as $a(x) = \sum_{\ell=0}^{n} a_\ell x^{[\ell]}$, and the composition in (2) can be expressed as $a\big(b(x)\big) = \sum_{\ell \geq 0} c_\ell x^{[\ell]}$ where $c_\ell = \sum_{i=0}^{\ell} a_i b_{\ell-i}^{[i]}$.

Finally, we note the following simple fact.

Proposition 1: For fixed nonzero $b(x) \in F[x]_\circ$, the mappings $F[x]_\circ \to F[x]_\circ$

$$a(x) \mapsto a(x) \circ b(x) \tag{3}$$

and

$$a(x) \mapsto a(x) \operatorname{rmod}_\circ b(x) \tag{4}$$

are linear over F_{q^L}. □

B. Linearized Polynomials and Vector Spaces

Linearized polynomials are intimately connected to vector spaces. We will often refer to vector spaces of the form (5), where F_{q^L} is an extension field of F_q and $S = \{\beta_0, \ldots, \beta_{n-1}\}$ is a subset of F_{q^L}. Then the subspace of F_{q^L} (over F_q) spanned by S is the set

$$\operatorname{span}(\beta_0, \ldots, \beta_{n-1}) \triangleq \left\{ \sum_{j=0}^{n-1} \alpha_j \beta_j : \alpha_j \in F_q \right\} \tag{5}$$

of all linear F_q combinations of elements in S. The dimension of (5) will be denoted by $\dim \operatorname{span}(\beta_0, \ldots, \beta_{n-1})$.

Linearized polynomials get their name from the following basic fact [43].

Algorithm 1: Linearized-Polynomial Synthesis
Input: $\beta_0, \ldots, \beta_{n-1}$ from the extension field F_{q^L} of F_q.
Output: nonzero $a(x) \in F[x]_\circ$ that vanishes on the subspace spanned by $\{\beta_0, \ldots, \beta_{n-1}\}$.

```
1  a(x) = x, ℓ := 0
2  while ℓ < n begin
3     Δ := a(β_ℓ)
4     if Δ ≠ 0 begin
5        a(x) := a(x)^q − Δ^{q−1} a(x)
6     end
7     ℓ := ℓ + 1
8  end
```

Proposition 2 (Linearity): Let $\beta_0, \ldots, \beta_{n-1}$ be nonzero elements of F_{q^L} over F_q. Then for any $a(x) = \sum_{i=0}^{n} a_i x^{[i]}$ in $F[x]_\circ$, it holds that $a(\beta) = \sum_{j\geq 0} \alpha_j a(\beta_j)$ for any $\beta = \sum_{j\geq 0} \alpha_j \beta_j$ with $\alpha_j \in F_q$. □

In consequence, we have

Proposition 3: Let $\beta_0, \ldots, \beta_{n-1}$ be nonzero elements of F_{q^L} over F_q. Then $a(x) \in F[x]_\circ$ vanishes on $\beta_0, \ldots, \beta_{n-1}$ if and only if $a(x)$ vanishes on the subspace of F_{q^L} over F_q spanned by $\{\beta_0, \ldots, \beta_{n-1}\}$. □

A polynomial that vanishes on a given subspace can be computed by Algorithm 1 (see box):

Proposition 4: Algorithm 1 returns a nonzero polynomial $a(x) \in F[x]_\circ$ that vanishes on $\operatorname{span}(\beta_0, \ldots, \beta_{n-1})$ with $\deg_q a(x) = \dim \operatorname{span}(\beta_0, \ldots, \beta_{n-1})$. □

Corollary 1: Algorithm 1 returns a nonzero polynomial of the smallest degree that vanishes on $\operatorname{span}(\beta_0, \ldots, \beta_{n-1})$. □

Conversely, we have (cf. [65])

Corollary 2: Let U be any subspace of F_{q^L} over F_q. The polynomial $a(x) \triangleq \prod_{\beta\in U}(x - \beta)$ is a linearized polynomial in $F[x]_\circ$. Moreover, any nonzero polynomial $b(x) \in F[x]_\circ$ that vanishes on U satisfies $\deg_q b(x) \geq \dim U$. □

Finally, we note

Proposition 5 (Null Space Factors): Let $\beta_0, \ldots, \beta_{n-1}$ be nonzero elements of F_{q^L} and let $a(x) \in F[x]_\circ$ be a nonzero linearized polynomial of the smallest degree that vanishes on $\beta_0, \ldots, \beta_{n-1}$. Then, $b(x) \in F[x]_\circ$ vanishes on $\beta_0, \ldots, \beta_{n-1}$ if and only if $b(x) = g(x) \circ a(x)$ for some $g(x) \in F[x]_\circ$. □

III. The Partial-Inverse Problem and the Basic Algorithm

Let $F[x]_\circ$ denote the ring of linearized polynomials over the extension field F_{q^L} of F_q as in Section II.

A. The Problem

The pivotal concept of this paper is the following problem, which is the obvious generalization of the partial-inverse problem of [52] to linearized polynomials.

Partial-Inverse Problem in $\mathbf{F[x]_\circ/m(x)}$: Let $b(x)$ and $m(x)$ be nonzero linearized polynomials in $F[x]_\circ$ with $\deg_q b(x) < \deg_q m(x)$. For given $d \in \mathbb{Z}$ with $0 \leq d \leq \deg_q m(x)$, find a nonzero $\Lambda(x) \in F[x]_\circ$ of the smallest degree such that

$$\deg_q\Big(\big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x)\Big) < d. \tag{6}$$

□

Theorem 1 (Uniqueness and Degree Bound): The partial-inverse problem for linearized polynomials has a unique solution (for every $d \geq 0$), up to a scale factor in F_{q^L}. Moreover, the solution $\Lambda(x)$ satisfies

$$\deg_q \Lambda(x) \leq \deg_q m(x) - d. \tag{7}$$

□

The theorem can be proved by using the argument in [52], which turns out to apply essentially unchanged to the setting of this paper. For completeness, the proof is given in Appendix A.

Proposition 6 (Reduced Partial-Inverse Problem): Coefficients b_ℓ of $b(x)$ with $\ell < 2d - \deg_q m(x)$ and coefficients m_ℓ of $m(x)$ with $\ell \leq 2d - \deg_q m(x)$ are irrelevant. In consequence, let $s \triangleq 2d - \deg_q m(x) > 0$ and define the linearized polynomials $\tilde{b}(x)$ and $\tilde{m}(x)$ with $\tilde{b}_\ell \triangleq b_{\ell+s}$ and $\tilde{m}_\ell \triangleq m_{\ell+s}$ for $\ell \geq 0$; then the modified partial-inverse problem with $b(x)$, $m(x)$, and d replaced by $\tilde{b}(x)$, $\tilde{m}(x)$, and $\tilde{d} \triangleq d - s$, respectively, has the same solution $\Lambda(x)$ as the original partial-inverse problem. □

The proposition can be proved by (7) and the argument in [52, Propositions 6 and 7].

B. The Basic Partial-Inverse Algorithm

The partial-inverse problem is solved by the basic algorithm stated as *Algorithm 2* (in the framed box). Lines 7 and 8 of this algorithm are explained by the following lemma.

Lemma 1 (Remainder Decreasing Lemma): Let $m(x)$ be a linearized polynomial in $F[x]_\circ$ with $\deg_q m(x) \geq 1$. For further polynomials $b(x), \Lambda'(x), \Lambda''(x) \in F[x]_\circ$, let

$$r'(x) \triangleq \big(\Lambda'(x) \circ b(x)\big) \bmod_\circ m(x), \tag{8}$$

$$r''(x) \triangleq \big(\Lambda''(x) \circ b(x)\big) \bmod_\circ m(x), \tag{9}$$

$d_1 \triangleq \deg_q r'(x)$, $\kappa_1 \triangleq \operatorname{lcf} r'(x)$, $d_2 \triangleq \deg_q r''(x)$, $\kappa_2 \triangleq \operatorname{lcf} r''(x)$, and assume $d_1 \geq d_2 \geq 0$. Then

$$\Lambda(x) \triangleq \kappa_2^{[d_1-d_2]}\Lambda'(x) - \kappa_1 x^{[d_1-d_2]} \circ \Lambda''(x) \tag{10}$$

satisfies $\deg_q\Big(\big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x)\Big) < d_1$. □

Proof: From (10), we obtain

$$r(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x) \tag{11}$$

$$= \kappa_2^{[d_1-d_2]} r'(x) - \kappa_1 x^{[d_1-d_2]} \circ r''(x). \tag{12}$$

From (12), we have $\deg_q r(x) < \deg_q r'(x) = d_1$. □

In consequence, the value of d_1 is reduced in every execution of line 8. Note that lines 8 and 12 do not require the computation of the entire polynomial $\Lambda'(x) \circ b(x) \bmod_\circ m(x)$. In particular, lines 8–12 can be replaced by Algorithm 2.A (see box).

Algorithm 2: Basic Partial-Inverse Algorithm

Input: $b(x)$, $m(x)$, and d as in the problem statement.
Output: $\Lambda(x)$ as in the problem statement.

1 **if** $\deg_q b(x) < d$ **begin**
2 **return** $\Lambda(x) := x$
3 **end**
4 $\Lambda'(x) := 0,\ d_1 := \deg_q m(x),\ \kappa_1 := \operatorname{lcf} m(x)$
5 $\Lambda''(x) := x,\ d_2 := \deg_q b(x),\ \kappa_2 := \operatorname{lcf} b(x)$
6 **loop begin**
7 $\Lambda'(x) := \kappa_2^{[d_1-d_2]}\Lambda'(x) - \kappa_1 x^{[d_1-d_2]} \circ \Lambda''(x)$
8 $d_1 := \deg_q(\Lambda'(x) \circ b(x) \bmod_\circ m(x))$
9 **if** $d_1 < d$ **begin**
10 **return** $\Lambda(x) := \Lambda'(x)$
11 **end**
12 $\kappa_1 := \operatorname{lcf}\big(\Lambda'(x) \circ b(x) \bmod_\circ m(x)\big)$
13 **if** $d_1 < d_2$ **begin**
14 $(\Lambda'(x), \Lambda''(x)) := (\Lambda''(x), \Lambda'(x))$
15 $(d_1, d_2) := (d_2, d_1)$
16 $(\kappa_1, \kappa_2) := (\kappa_2, \kappa_1)$
17 **end**
18 **end**

See also the refinements (Algorithms 2.A) below.

Algorithm 2.A: Lines 8–12 of Algorithm 2 can be implemented as follows:

21 **repeat**
22 $d_1 := d_1 - 1$
23 **if** $d_1 < d$ **begin**
24 **return** $\Lambda(x) := \Lambda'(x)$
25 **end**
26 $\kappa_1 :=$ coefficient of $x^{[d_1]}$ in $\big(\Lambda'(x) \circ b(x)\big) \bmod_\circ m(x)$
27 **until** $\kappa_1 \neq 0$

Algorithm 3: Partial-Inverse Algorithm Restated

1 **if** $\deg_q b(x) < d$ **begin**
2 **return** $\Lambda(x) := x$
3 **end**
4 $\Lambda'(x) := 0,\ d_1 := \deg_q m(x),\ \kappa_1 := \operatorname{lcf} m(x)$
5 $\Lambda''(x) := x,\ d_2 := \deg_q b(x),\ \kappa_2 := \operatorname{lcf} b(x)$
Extra:
$k := 0$ (E.1)
6 **loop begin**
Assertions:
$d_1 > d_2 \geq d$ (A.1)
$\deg_q \Lambda''(x) = \deg_q m(x) - d_1$ (A.2)
$> \deg_q \Lambda'(x)$ (A.3)
Extra:
$k := k+1, \Delta_k \triangleq d_1 - d_2, \Lambda_k(x) \triangleq \Lambda''(x)$ (E.2)
7 **repeat**
8 $\Lambda'(x) := \kappa_2^{[d_1-d_2]}\Lambda'(x) - \kappa_1 x^{[d_1-d_2]} \circ \Lambda''(x)$
Assertions:
$\deg_q(\Lambda'(x) \circ b(x) \bmod_\circ m(x)) < d_1$ (A.4)
$\deg_q \Lambda'(x) = \deg_q m(x) - d_2$ (A.5)
$> \deg_q \Lambda''(x)$ (A.6)
9 $d_1 := \deg_q(\Lambda'(x) \circ b(x) \bmod_\circ m(x))$
10 **if** $d_1 < d$ **begin**
11 **return** $\Lambda(x) := \Lambda'(x)$
12 **end**
13 $\kappa_1 := \operatorname{lcf}\big(\Lambda'(x) \circ b(x) \bmod_\circ m(x)\big)$
14 **until** $d_1 < d_2$
15 $(\Lambda'(x), \Lambda''(x)) := (\Lambda''(x), \Lambda'(x))$
16 $(d_1, d_2) := (d_2, d_1)$
17 $(\kappa_1, \kappa_2) := (\kappa_2, \kappa_1)$
18 **end**

C. Preparations for the Proof

In preparation for the proof of the algorithm, we restate Algorithm 2 with added assertions as Algorithm 3 (see box), cf. [52]. Note that throughout the algorithm (except at the very beginning, before the first execution of lines 9 and 13), d_1, d_2, κ_1, and κ_2 are defined as in Lemma 1, i.e., $d_1 = \deg_q r'(x)$, $\kappa_1 = \operatorname{lcf} r'(x)$, $d_2 = \deg_q r''(x)$, and $\kappa_2 = \operatorname{lcf} r''(x)$ for $r'(x)$ and $r''(x)$ as in (8) and (9).

Assertions (A.1)–(A.3) are easily seen to be true, both from the initialization, and from (A.5) and (A.6).

As for (A.4), after the very first execution of line 8, we still have $d_1 = \deg_q m(x)$ (from line 4), which makes (A.4) obvious. For all later executions of line 8, (A.4) follows from Lemma 1.

As for (A.5) and (A.6), we note that line 8 changes the degree of $\Lambda'(x)$ as follows:

- Upon entering the **repeat** loop, line 8 increases the degree of $\Lambda'(x)$ to

$$\deg_q \Lambda''(x) + d_1 - d_2 = \deg_q m(x) - d_2 \quad (13)$$
$$> \deg_q \Lambda''(x), \quad (14)$$

  which follows from (A.1)–(A.3).
- Subsequent executions of line 8 without leaving the **repeat** loop (i.e., without executing lines 15–17) do not change the degree of $\Lambda'(x)$.

(A.1) and (A.5) together imply that $\Lambda(x)$ returned by the algorithm satisfies (7).

The following lemma refers to Δ_k and $\Lambda_k(x)$ as defined in (E.2), as well as to

$$r_k(x) \triangleq (\Lambda_k(x) \circ b(x)) \bmod_\circ m(x) \quad (15)$$

and $r_0(x) \triangleq m(x)$.

Lemma 2 (Degree Difference Lemma): Let $\Delta_1, \ldots, \Delta_K$ and $\Lambda_1(x), \ldots, \Lambda_K(x)$ be the values of Δ_k and $\Lambda_k(x)$, respectively, throughout the algorithm (i.e., K is the last value of k in (E.2)). Let $\Lambda_{K+1}(x)$ be the polynomial $\Lambda(x)$

returned by the algorithm. Then

$$\begin{aligned} \Delta_k &= \deg_q r_{k-1}(x) - \deg_q r_k(x) && (16)\\ &= \deg_q \Lambda_{k+1}(x) - \deg_q \Lambda_k(x) && (17)\\ &> 0 && (18) \end{aligned}$$

for $k = 1, \ldots, K$. □

Proof: Eq. (16) is obvious from the initialization (lines 4 and 5) and from line 9 of the algorithm. The point of the lemma is (17), which follows from the initialization and (13). Finally, (18) is immediate from (A.1). □

D. Completing the Proof of the Algorithm

It is clear at this point that the algorithm returns a polynomial $\Lambda(x)$ that satisfies (6). It remains to prove that $\Lambda(x)$ has the smallest possible degree.

Let Δ_k and $\Lambda_k(x)$ be defined as in Lemma 2. Any nonzero $\tilde{\Lambda}(x) \in F[x]_\circ$ with $\deg_q \tilde{\Lambda}(x) < \deg_q \Lambda(x)$ can be written as

$$\tilde{\Lambda}(x) = \sum_{k=1}^{K} q_k(x) \circ \Lambda_k(x) \tag{19}$$

with

$$\begin{aligned} \deg_q q_k(x) &< \deg_q \Lambda_{k+1}(x) - \deg_q \Lambda_k(x) && (20)\\ &= \Delta_k. && (21) \end{aligned}$$

Consider

$$\begin{aligned} \tilde{r}(x) &\triangleq \big(\tilde{\Lambda}(x) \circ b(x)\big) \bmod_\circ m(x) && (22)\\ &= \sum_{k=1}^{K} \big(q_k(x) \circ r_k(x)\big) \bmod_\circ m(x). && (23) \end{aligned}$$

From (21) and (16), we have

$$\deg_q q_k(x) + \deg_q r_k(x) < \deg_q r_{k-1}(x) \tag{24}$$

for $k = 1, \ldots, K$. As a first consequence, (23) becomes

$$\tilde{r}(x) = \sum_{k=1}^{K} q_k(x) \circ r_k(x). \tag{25}$$

From (25) and (24), we further obtain

$$\deg_q \tilde{r}(x) \geq \deg_q r_\ell(x), \tag{26}$$

where ℓ is the smallest index $k \in \{1, \ldots, K\}$ such that $q_k(x) \neq 0$. But $\deg_q r_k(x) \geq d$ for all $k \in \{1, \ldots, K\}$; thus $\deg_q \tilde{r}(x) \geq d$, which concludes the proof.

E. Remarks

The proof in Section III-D is new and simpler than the proof in [52]. This new proof also works for the setting of [52].

Variations of the basic partial-inverse algorithm and a discussion of their complexity will be given in Section IV. Concerning the latter, let N_{it} be the number of executions of line 26 in Algorithm 2. This quantity is bounded by

$$\begin{aligned} N_{\text{it}} &\leq \deg_q m(x) - d + \deg_q \Lambda(x), && (27)\\ &\leq 2(\deg_q m(x) - d). && (28) \end{aligned}$$

Algorithm 4: Reverse Berlekamp–Massey Algorithm

In two important special cases, the computation of line 26 in Algorithm 2.A simplifies as follows.

In the special case where $m(x) = x^{[\nu]}$, line 26 amounts to

31 $\quad \kappa_1 := \Lambda'_0 b^{[0]}_{d_1} + \Lambda'_1 b^{[1]}_{d_1-1} + \ldots + \Lambda'_\tau b^{[\tau]}_{d_1-\tau}$

with $\tau \triangleq \deg_q \Lambda'(x)$ and where $b_\ell \triangleq 0$ for $\ell < 0$.

In another special case where $m(x) = x^{[n]} - x^{[0]}$, line 26 becomes

51 $\quad \kappa_1 := \Lambda'_0 b^{[0]}_{d_1} + \Lambda'_1 b^{[1]}_{[d_1-1]} + \ldots + \Lambda'_\tau b^{[\tau]}_{[d_1-\tau]}$

with $b_{[\ell]} \triangleq b_{\ell \bmod n}$.

The bound (27) can be proved as in [52], and (28) follows from (7). The weaker bound (28) can also be proved directly by noting that, in every iteration, the larger of the remainder degrees is reduced by at least 1.

Additional properties of the partial-inverse problem are discussed in Appendix B.

IV. Realizations of the Partial-Inverse Algorithm

The basic partial-inverse algorithm of the previous section can be implemented or specialized in different ways, including (a version of) the Berlekamp–Massey algorithm and (a version of) the Euclidean algorithm. The generalization of the corresponding algorithms from [52] turns out to be rather obvious.

A. The (Reverse) Berlekamp–Massey Algorithm

In the special cases $m(x) = x^{[\nu]}$ and $m(x) = x^{[n]} - x^{[0]}$, line 26 of Algorithm 2.A can be computed as in Algorithm 4, which looks very much like, and is as efficient as, the generalized Berlekamp–Massey algorithm [40], [41], [42].

Following [52], we refer to this algorithm as the *reverse* Berlekamp–Massey algorithm because, in applications to decoding, it processes the syndrome in the reverse order of the Berlekamp–Massey algorithm.

As stated, the (reverse) Berlekamp–Massey algorithm applies only to the case $m(x) = x^{[\nu]}$ or $m(x) = x^{[n]} - x^{[0]}$. It is therefore noteworthy that a partial-inverse problem with general $m(x)$ can always be transformed into an equivalent partial-inverse problem with $m(x) = x^{[2\tau]}$, as shown in Appendix C.

From (28) and line 31, the complexity of this algorithm is easily seen to be $\mathcal{O}\big((\nu - d)^2\big)$ with $\nu \triangleq \deg_q m(x)$.

B. The Remainder-Saving Algorithm (= Euclidean Algorithm)

A variation or implementation of Algorithm 2 for general $m(x)$ is Algorithm 5 (see box), where we store and update the remainders $r'(x)$ and $r''(x)$ in (8) and (9). In consequence, the computation of line 26 (in Algorithm 2.A) is unnecessary. All other quantities in the algorithm remain unchanged.

Algorithm 5 may be viewed as a version of the (right) Euclidean algorithm (for linearized polynomials). The latter

Algorithm 5: Remainder Saving Partial-Inverse Algorithm (Linearized Euclidean Algorithm)

1 **if** $\deg_q b(x) < d$ **begin**
2 **return** $\Lambda(x) := x$
3 **end**
4 $\Lambda'(x) := 0,\ d_1 := \deg_q m(x),\ \kappa_1 := \operatorname{lcf} m(x)$
5 $\Lambda''(x) := x,\ d_2 := \deg_q b(x),\ \kappa_2 := \operatorname{lcf} b(x)$
6 $r'(x) := m(x),\ r''(x) := b(x)$
7 **loop begin**
8 $\Lambda'(x) := \kappa_2^{[d_1-d_2]}\Lambda'(x) - \kappa_1 x^{[d_1-d_2]} \circ \Lambda''(x)$
9 $r'(x) := \kappa_2^{[d_1-d_2]} r'(x) - \kappa_1 x^{[d_1-d_2]} \circ r''(x)$
10 $d_1 := \deg_q r'(x)$
11 **if** $d_1 < d$ **begin**
12 **return** $\Lambda(x) := \Lambda'(x)$
13 **end**
14 $\kappa_1 := \operatorname{lcf} r'(x)$
15 **if** $d_1 < d_2$ **begin**
16 $(\Lambda'(x), \Lambda''(x)) := (\Lambda''(x), \Lambda'(x))$
17 $(r'(x), r''(x)) := (r''(x), r'(x))$
18 $(d_1, d_2) := (d_2, d_1)$
19 $(\kappa_1, \kappa_2) := (\kappa_2, \kappa_1)$
20 **end**
21 **end**

Algorithm 6: Quotient Saving Partial-Inverse Algorithm

1 **if** $\deg_q b(x) < d$ **begin**
2 **return** $\Lambda(x) := x$
3 **end**
4 $\Lambda'(x) := 0,\ d_1 := \deg_q m(x),\ \kappa_1 := \operatorname{lcf} m(x)$
5 $\Lambda''(x) := x,\ d_2 := \deg_q b(x),\ \kappa_2 := \operatorname{lcf} b(x)$
6 $q'(x) := -x,\ q''(x) := 0$
7 **loop begin**
8 $\Lambda'(x) := \kappa_2^{[d_1-d_2]}\Lambda'(x) - \kappa_1 x^{[d_1-d_2]} \circ \Lambda''(x)$
9 $q'(x) := \kappa_2^{[d_1-d_2]} q'(x) - \kappa_1 x^{[d_1-d_2]} \circ q''(x)$
10 **repeat**
11 $d_1 := d_1 - 1$
12 **if** $d_1 < d$ **begin**
13 **return** $\Lambda(x) := \Lambda'(x)$
14 **end**
15 $\kappa_1 := \sum_{\ell=0}^{\tau} \Lambda'_\ell b^{[\ell]}_{d_1-\ell} - \sum_{\ell=0}^{\nu} q'_\ell m^{[\ell]}_{d_1-\ell}$
16 **until** $\kappa_1 \neq 0$
17 **if** $d_1 < d_2$ **begin**
18 $(\Lambda'(x), \Lambda''(x)) := (\Lambda''(x), \Lambda'(x))$
19 $(q'(x), q''(x)) := (q''(x), q'(x))$
20 $(d_1, d_2) := (d_2, d_1)$
21 $(\kappa_1, \kappa_2) := (\kappa_2, \kappa_1)$
22 **end**
23 **end**

is well known in the literature for both ordinary polynomials and linearized polynomials, see, e.g., [3], [30], [36], [37], [38], [39], [52], and [66]. In other words, the well-known Euclidean algorithm actually solves the partial-inverse problem.

From (28) and line 9, the complexity of this algorithm is easily seen to be $\mathcal{O}(\nu(\nu-d))$ with $\nu \triangleq \deg_q m(x)$. (The computation in line 9 may be reduced by assuming a reduced partial-inverse problem according to Proposition 6.) Asymptotically faster versions of the Euclidean algorithm have been proposed in the literature.

C. The Quotient-Saving Algorithm

By storing and updating also the quotients $q'(x)$ and $q''(x)$ defined by

$$\Lambda'(x) \circ b(x) = q'(x) \circ m(x) + r'(x) \tag{29}$$

$$\Lambda''(x) \circ b(x) = q''(x) \circ m(x) + r''(x) \tag{30}$$

with $r'(x)$ and $r''(x)$ as in (8) and (9), the coefficient of $x^{[d_1]}$ of $r'(x)$ (line 26 in Algorithm 2.A) can then be computed as

$$\kappa_1 := \sum_{\ell=0}^{\tau} \Lambda'_\ell b^{[\ell]}_{d_1-\ell} - \sum_{\ell=0}^{\nu} q'_\ell m^{[\ell]}_{d_1-\ell} \tag{31}$$

with $\tau \triangleq \deg_q \Lambda'(x)$ and $\nu \triangleq \deg_q q'(x)$, and where both $b_\ell \triangleq 0$ and $m_\ell \triangleq 0$ for $\ell < 0$. All other quantities in the algorithm remain unchanged. We then obtain Algorithm 6 (see box), which is a new algorithm of the Berlekamp–Massey type and achieves a generalization of Algorithm 4 to general $m(x)$. The complexity of Algorithm 6 is $\mathcal{O}\big((\nu-d)^2\big)$ with $\nu \triangleq \deg_q m(x)$.

V. Preliminaries for Gabidulin Codes

In this section, we briefly review the rank metric and the evaluation transform, cf. [2], [3], [4], [5], [8], and [30]. We conclude with Proposition 7, which appears to be new.

A. Rank Metric

Let F_{q^L} be an extension field of F_q. The notion of rank as a metric for codes was introduced in [2], [3], and [4]. In this paper, the rank distance between vectors in $(F_{q^L})^n$ will be defined as follows. For a vector $a = (a_0, \ldots, a_{n-1}) \in (F_{q^L})^n$, let $\operatorname{span}(a)$ be the subspace of F_{q^L} (over F_q) spanned by $\{a_0, \ldots, a_{n-1}\}$ as in (5). The *rank weight* of $a \in (F_{q^L})^n$ is then defined as $\mathrm{w_R}(a) \triangleq \dim \operatorname{span}(a)$, i.e., the dimension of $\operatorname{span}(a)$. Note that $\mathrm{w_R}(a) \triangleq 0$ if and only if $a = 0$; $\mathrm{w_R}(a-b) = \mathrm{w_R}(b-a)$ for any $a, b \in (F_{q^L})^n$, and $\mathrm{w_R}(a-b) \leq \mathrm{w_R}(a) + \mathrm{w_R}(b)$. Clearly, $\mathrm{w_R}(\cdot)$ is a metric for vectors in $(F_{q^L})^n$. The *rank distance* between any $a, b \in (F_{q^L})^n$ can thus be defined as $\mathrm{d_R}(a,b) \triangleq \mathrm{w_R}(a-b)$.

B. Evaluation Transform

We will define Gabidulin codes via the following theorem, cf. [5], [8], and [30].

Theorem 2 (Linearized Evaluation Transform): Let $\beta_0, \ldots, \beta_{n-1}$ be linearly independent elements of F_{q^L} over F_q, and let V be the set of linearized polynomials in $F[x]_\circ$ with q-degree less than n. Then, the mapping

$$\psi : V \to (F_{q^L})^n : a(x) \mapsto \big(a(\beta_0), \ldots, a(\beta_{n-1})\big) \tag{32}$$

is a linear transform over F_{q^L}, i.e., ψ is linear over F_q, injective and surjective. The inverse mapping is

$$\psi^{-1} : \big(c_0, \ldots, c_{n-1}\big) \mapsto a(x) = \sum_{\ell=0}^{n-1} c_\ell \cdot \tilde{m}_\ell(x) \tag{33}$$

with coefficients $\tilde{m}_\ell(x) \triangleq \big(m_\ell(\beta_\ell)\big)^{-1} m_\ell(x)$ where $m_\ell(x)$ is a monic linearized polynomial $\in F[x]_\circ$ of the smallest degree that vanishes on $\{\beta_0, \ldots, \beta_{n-1}\} \setminus \{\beta_\ell\}$. □

The inverse mapping (33) can be seen as a generalized Lagarange interpolation, cf., e.g., [5] and [30]. In the special case where $n = L$ and $\mathcal{B}$ is a normal basis, the transform reduces to the q-transform of [8].

C. Complementary Rank Weight Property

Let ψ be defined as in (32).

Proposition 7: Let $\beta_0, \ldots, \beta_{n-1}$ be linearly independent elements of F_{q^L} over F_q, and let $U \triangleq \operatorname{span}(\beta_0, \ldots, \beta_{n-1})$. For any $a = (a_0, \ldots, a_{n-1}) \in (F_{q^L})^n$ with rank weight $\mathrm{w_R}(a)$, the polynomial $A(x) \triangleq \psi^{-1}(a)$ with ψ as in (32) vanishes on some subspace $M \subset U$ with $\dim M = n - \mathrm{w_R}(a)$. □

Proof: Let $W \triangleq \operatorname{span}(a_0, \ldots, a_{n-1}) \subset F_{q^L}$. Note that $\dim U = n \geq \dim W = \mathrm{w_R}(a)$. We then define the mapping $\Phi : U \to W : \beta \mapsto A(\beta)$. Clearly, the mapping Φ is linear over F_q, cf. Proposition 2; in addition, it is surjective since $a_\ell = A(\beta_\ell)$ for every ℓ. Therefore, the dimension of the kernel of Φ is $\dim \ker \Phi = \dim U - \dim W$. □

VI. Decoding Gabidulin Codes

We now develop the partial-inverse approach to Gabidulin codes. The generalization of the corresponding material of [52] to linearized polynomials turns out to be straightforward. However, Proposition 10 and the converse part of Theorem 3 appear to be new results for Gabidulin codes.

A. Gabidulin Codes

Let F_{q^L} be an extension field of F_q, and let $\beta_0, \ldots, \beta_{n-1}$ be linearly independent elements of F_{q^L} over F_q. An (n, k) Gabidulin code $\mathcal{C}$ (with blocklength $n \leq L$ and dimension k) can be defined via (32) as the set

$$\{c = (c_0, \ldots, c_{n-1}) \in (F_{q^L})^n : \deg_q \psi^{-1}(c) < k\}. \tag{34}$$

It is well known that Gabidulin codes are maximum distance separable codes in the rank metric [2], [3], [4]. In terms of rank weight defined in Section V-A, the minimum rank distance of $\mathcal{C}$ in (34) can be defined as $\mathrm{d_R}(\mathcal{C}) \triangleq \min\{\mathrm{w_R}(c - c') : c, c' \in \mathcal{C}, c \neq c'\}$, which satisfies

$$\mathrm{d_R}(\mathcal{C}) = n - k + 1. \tag{35}$$

Equation (35) is proved in [3] and [30]; another (simple) proof is given in Appendix D.

B. Error-Span Polynomial and Interpolation

The problem of decoding Gabidulin codes can be described as follows. Let $y = (y_0, \ldots, y_{n-1})$ be a received word, which we wish to decompose into $y = c + e$ where $c \in \mathcal{C}$ is a codeword and $e = (e_0, \ldots, e_{n-1}) \in (F_{q^L})^n$ is the error with rank weight $\mathrm{w_R}(e)$ as small as possible.

Let $C(x) \triangleq \psi^{-1}(c)$, $E(x) \triangleq \psi^{-1}(e)$, and $Y(x) \triangleq \psi^{-1}(y)$ with ψ as in (32). We then have

$$Y(x) = C(x) + E(x) \tag{36}$$

with $\deg_q C(x) < k$ and $\deg_q E(x) < n$. The task of decoding is to recover $C(x)$ from $Y(x)$. Toward this end, we wish to find an error-span polynomial:[1]

Definition 1 (Error-Span Polynomial): For any vector $e = (e_0, \ldots, e_{n-1}) \in (F_{q^L})^n$, an error-span polynomial $\Lambda_e(x)$ is a nonzero polynomial $\in F[x]_\circ$ of the smallest degree that vanishes on the subspace of F_{q^L} (over F_q) spanned by $e_0, \ldots, e_{n-1}$. □

Proposition 8: The error-span polynomial $\Lambda_e(x)$ is unique up to a scale factor $\in F_{q^L}$, and it satisfies $\deg_q \Lambda_e(x) = \mathrm{w_R}(e)$. □

The proof is immediate from Corollary 2.

Now, let $m(x) \in F[x]_\circ$ be a nonzero polynomial of the smallest degree such that $m(\beta_\ell) = 0$ for $\ell = 0, \ldots, n-1$. Note that $\deg_q m(x) = n$ by Proposition 3 and Corollary 2.

Proposition 9 (Error-Locator Equation): The error-span polynomial $\Lambda_e(x)$ satisfies $\Lambda_e(x) \circ E(x) = A(x) \circ m(x)$ for some $A(x) \in F[x]_\circ$ of $\deg_q A(x) < \deg_q \Lambda_e(x)$. Conversely, if some nonzero $\Lambda(x) \in F[x]_\circ$ satisfies

$$\Lambda(x) \circ E(x) = A(x) \circ m(x) \tag{37}$$

for some $A(x) \in F[x]_\circ$, then $\Lambda(x) = g(x) \circ \Lambda_e(x)$ for some nonzero $g(x) \in F[x]_\circ$. □

Proof: Note that $\Lambda_e\big(E(\beta_\ell)\big) = 0$ for all $\ell \in \{0, \ldots, n-1\}$. Therefore, $\Lambda_e(x) \circ E(x)$ vanishes on $\{\beta_0, \ldots, \beta_{n-1}\}$. The first claim then follows from Proposition 5. As for the converse, (37) implies that $\Lambda\big(E(\beta_\ell)\big) = 0$ for all $\ell \in \{0, \ldots, n-1\}$; thus $\Lambda(x)$ vanishes on $\{e_0, \ldots, e_{n-1}\}$. By Proposition 5, we obtain $\Lambda(x) = g(x) \circ \Lambda_e(x)$ for some $g(x) \in F[x]_\circ$. □

Proposition 10 (Multiply-Divide Interpolation): If $\Lambda(x) = g(x) \circ \Lambda_e(x)$ for some nonzero $g(x) \in F[x]_\circ$ with $\deg_q \Lambda(x) \leq n - k$, then

$$C(x) = r(x) \operatorname{ldiv}_\circ \Lambda(x) \tag{38}$$

where

$$r(x) = \big(\Lambda(x) \circ Y(x)\big) \operatorname{rmod}_\circ m(x). \tag{39}$$

□

Proof: If $\Lambda(x)$ has the stated properties, then

$$\begin{aligned} &\Lambda(x) \circ Y(x) \operatorname{rmod}_\circ m(x) \\ &\quad = \Lambda(x) \circ C(x) \operatorname{rmod}_\circ m(x) + \Lambda(x) \circ E(x) \operatorname{rmod}_\circ m(x) \\ &\quad = \Lambda(x) \circ C(x), \end{aligned} \tag{40}$$

where the last step follows from Proposition 9. □

[1]The idea of a key equation with an error span polynomial dates back to [3]. The term "error span polynomial" was introduced in [4].

In the special case where $m(x) = x^{[n]} - x^{[0]}$, computing (39) amounts to a kind of cyclic convolution as in line 51 of Algorithm 4.

C. A Key Equation

Following [52, Theorem 6], we have the following theorem, the converse part of which is new (for linearized polynomials):

Theorem 3 (A Key Equation): If $\mathrm{w_R}(e) \le \frac{n-k}{2}$, then the error-span polynomial $\Lambda_e(x)$ satisfies

$$\begin{aligned} \deg_q\big(\Lambda_e(x) \circ Y(x) \bmod_\circ m(x)\big) &< k + \deg_q \Lambda_e(x) \quad (41)\\ &\le n - (n-k)/2. \quad (42)\end{aligned}$$

Conversely, for any y and $e \in (F_{q^L})^n$ and $t \in \mathbb{R}$ with

$$\mathrm{w_R}(e) \le t \le (n-k)/2, \tag{43}$$

if some nonzero $\Lambda(x) \in F[x]_\circ$ with $\deg_q \Lambda(x) \le t$ satisfies

$$\deg_q\big(\Lambda(x) \circ Y(x) \bmod_\circ m(x)\big) < n - t, \tag{44}$$

then $\Lambda(x) = g(x) \circ \Lambda_e(x)$ for some $g(x) \in F[x]_\circ$. □

Corollary 3: If $\mathrm{w_R}(e) \le \frac{n-k}{2}$, then $\Lambda_e(x)$ is the nonzero linearized polynomial of the smallest degree (unique up to a scale factor) that satisfies

$$\deg_q\big(\Lambda(x) \circ Y(x) \bmod_\circ m(x)\big) < \frac{n+k}{2} \tag{45}$$

□

Note that $\Lambda_e(x)$ from (45) is a partial-inverse problem.

Theorem 3 can be proved by using a similar idea as in [52]. However, the proof of the converse requires Proposition 7.

Proof of Theorem 3: (41) is clear from (40) with $\Lambda(x) = \Lambda_e(x)$; (42) follows from $\deg_q \Lambda_e(x) = \mathrm{w_R}(e)$.

As for the converse, assume (43), (44), and $\deg_q \Lambda(x) \le t$. Consider

$$\begin{aligned} &\Lambda(x) \circ Y(x) \bmod_\circ m(x)\\ &\quad = \Lambda(x) \circ C(x) + \Lambda(x) \circ E(x) \bmod_\circ m(x). \quad (46)\end{aligned}$$

□

Under the stated assumptions, the q-degree of the left-hand side of (46) is smaller than $n-t$, and $\deg_q\big(\Lambda(x) \circ C(x)\big) < n-t$. Thus $\deg_q\big(\Lambda(x) \circ E(x) \bmod_\circ m(x)\big) < n-t$. Now, let $r(x) \triangleq \Lambda(x) \circ E(x) \bmod_\circ m(x)$, and write $r(x) = \Lambda(x) \circ E(x) - g(x) \circ m(x)$ for some $g(x) \in F[x]_\circ$. By Proposition 7, $E(x)$ vanishes on some subspace $M \subset \mathrm{span}\{\beta_0, \ldots, \beta_{n-1}\}$ with $\dim M = n - \mathrm{w_R}(e) \ge n - t$. It follows that $r(x)$ also vanishes on M and thus $\deg_q r(x) \ge n - t$ if $r(x) \ne 0$. But $\deg_q r(x) < n - t$. Therefore $r(x) = 0$, i.e., $\Lambda(x) \circ E(x) = g(x) \circ m(x)$. Proposition 9 concludes the proof. □

D. Decoding Algorithms

Determining $\Lambda_e(x)$ from (45) is a partial-inverse problem. We thus arrive at the following decoding procedure:

1) Compute $Y(x) = \psi^{-1}(y)$.
2) Solve the key equation (45) by any of the algorithms in Section IV.
   (If $\mathrm{w_R}(e) \le \frac{n-k}{2}$, the polynomial $\Lambda(x)$ returned by the algorithm equals $\Lambda_e(x)$, up to a scale factor.)
3) Complete decoding by Proposition 10.

For example, if $m(x) = x^{[n]} - x^{[0]}$, Algorithm 4 can be applied to (45). In this case, computing (39) amounts to the "cyclic convolution" $r_\ell = \sum_{i=0}^{\tau} \Lambda_i Y_{[\ell-i]}^{[i]}$ with $\tau \triangleq \deg_q \Lambda(x)$.

Theorem 3 is versatile; along with Proposition 10, it can also be used to prove the correctness of the Shiozaki–Gao decoder [30], [37].

We also note that in solving (45), the number of iterations N_{it} in Algorithm 2 is upper bounded by $(n-k)/2 + \mathrm{w_R}(e)$, cf. (27). By contrast, the (standard, not the reverse) Berlekamp–Massey algorithm requires $n-k$ iterations, which is typically larger.

Finally, we note that the division in (38) might have a nonzero remainder (if not all assumptions are satisfied). This condition should be checked; if it is violated, an uncorrectable error should be announced.

VII. Interpolation

In this section, we consider erasure decoding and develop new interpolation methods for different types of erasures. (The results of this section are independent of the partial-inverse approach.)

Below, an element in F_{q^L} will be viewed as a *column* vector of length L over F_q, and a vector $a = (a_0, \ldots, a_{n-1})$ in $(F_{q^L})^n$ is viewed as a $L \times n$ matrix over F_q.

1) Interpolation of Row Erasures: Consider a received word $y = c + e \in (F_{q^L})^n$, where c is a codeword $\in \mathcal{C}$ as in Section VI-A. The error $e \in (F_{q^L})^n$ is called a row(s) erasure if e corrupts only some rows of $c \in (F_{q^L})^n$ (viewed as a matrix over F_q) and the receiver knows the indices of the rows that are corrupted. In this case, $C(x) = \psi^{-1}(c)$ in (36) can be recovered as follows.

Let $\mathcal{Z}_r \subset \{1, \ldots, L\}$ be a set consisting of (known) indices of the corrupted rows, and let $\delta_i \in F_{q^L}$ (viewed as a column vector) be the transpose of $(0, \ldots, 0, 1, 0, \ldots, 0)$ where $1 \in F_q$ sits at position i. Then, we define $a_r \in (F_{q^L})^{|\mathcal{Z}_r|}$ such that the components of a_r are the vectors $\delta_i, i \in \mathcal{Z}_r$. For example, if $\mathcal{Z}_r = \{2, 5\}$, then $a_r \triangleq (\delta_2, \delta_5)$.

Definition 2 (Row-Span Polynomial): For given $a_r \in (F_{q^L})^{|\mathcal{Z}_r|}$, a row-span polynomial $\Lambda_r(x)$ is nonzero polynomial in $F[x]_\circ$ of the smallest degree that vanishes on the subspace spanned by the components of a_r. □

Lemma 3: For any rows-erasure e with erasure index set $\mathcal{Z}_r$, it holds that $\mathrm{span}(e) \subseteq \mathrm{span}(a_r)$ and $\Lambda_r(x) = g(x) \circ \Lambda_e(x)$ for some nonzero $g(x) \in F[x]_\circ$. □

Proof: For each e_i of $e = (e_0, \ldots, e_{n-1}) \in (F_{q^L})^n$, we have $e_i \in \mathrm{span}(a_r)$, and thus $\mathrm{span}(e) \subseteq \mathrm{span}(a_r)$. The second claim then follows from Proposition 5. □

The polynomial $\Lambda_r(x)$ in Lemma 3 can be computed by Algorithm 1. Note that $\deg_q \Lambda_r(x) = |\mathcal{Z}_r|$.

Proposition 11: If $|\mathcal{Z}_r| \le n - k$, then $C(x)$ can be recovered from (38) with $\Lambda(x) = \Lambda_r(x)$. □

The proof is immediate from Lemma 3 and Proposition 10.

2) Interpolation of Column Erasures: Let $\mathcal{Z}_c$ be a subset of $\{0,1,\ldots,n-1\}$. Assume that the error $e=(e_0,\ldots,e_{n-1}) \in (F_{q^L})^n$ in $y=c+e$ satisfies $e_\ell = 0 \in F_{q^L}$ for $\ell \notin \mathcal{Z}_c$ and with arbitrary (not necessarily nonzero) e_ℓ for $\ell \in \mathcal{Z}_c$. Assume that $\mathcal{Z}_c$ is known by the decoder. In this case, the polynomial $C(x)=\psi^{-1}(c)$ in (36) can be recovered by Proposition 12.

Proposition 12: Let $S \triangleq \{0,\ldots,n-1\} \setminus \mathcal{Z}_c$, and let $m_S(x)$ be a nonzero polynomial of the smallest degree that vanishes on β_ℓ for all $\ell \in S$. If $|\mathcal{Z}_c| \le n-k$, then $C(x) = Y(x) \bmod_\circ m_S(x)$. □

Proof: Since $E(x) \triangleq \psi^{-1}(e) \in F[x]_\circ$ satisfies $E(\beta_\ell)=0$ for all $\ell \in S$, we have $E(x) = g(x) \circ m_S(x)$ for some $g(x) \in F[x]_\circ$ by Proposition 5. Note that $\deg_q m_S(x) = n - |\mathcal{Z}_c|$. If $|\mathcal{Z}_c| \le n-k$, then $\deg_q C(x) < \deg_q m_S(x)$ and therefore $\big(C(x)+E(x)\big) \bmod_\circ m_S(x) = C(x)$. □

3) Interpolation of Crisscross Erasures: For given received word $y=c+e$ in $(F_{q^L})^n$, suppose that $e=r+z \in (F_{q^L})^n$ where r is a rows erasure and z is a columns erasure. For rows-erasure r, we assume that the set of erasure positions $\mathcal{Z}_r \subset \{1,\ldots,L\}$ as in Section VII-1 (or $a_r \in (F_{q^L})^{|\mathcal{Z}_r|}$ as in Lemma 3) is known by the decoder. For columns-erasure z, we assume that the set of column positions $\mathcal{Z}_c \subset \{0,\ldots,n-1\}$ is known by the decoder: for $\ell \in \mathcal{Z}_c$, the symbol y_ℓ is useless and can be ignored; for $\ell \notin \mathcal{Z}_c$, $z_\ell \triangleq 0$ and $y_\ell = c_\ell + r_\ell$.

In this case, the polynomial $C(x)=\psi^{-1}(c)$ in (36) can be recovered by the following proposition.

Proposition 13: Let $\Lambda_r(x)$ be a nonzero polynomial of the smallest degree that vanishes on $\mathrm{span}(a_r)$, and let $m_S(x)$ be a nonzero polynomial of the smallest degree that vanishes on β_ℓ for all $\ell \in \{0,\ldots,n-1\} \setminus \mathcal{Z}_c$. If $|\mathcal{Z}_r|+|\mathcal{Z}_c| \le n-k$, then

$$C(x) = P(x) \operatorname{ldiv}_\circ \Lambda_r(x) \tag{47}$$

where $P(x) \triangleq \big(\Lambda_r(x) \circ Y(x)\big) \bmod_\circ m_S(x)$. □

Proof: We write $Y(x)=C(x)+R(x)+Z(x)$ with $R(x)=\psi^{-1}(r)$ and $Z(x)=\psi^{-1}(z)$, and obtain

$$\Lambda_r(x) \circ Y(x) = \Lambda_r(x) \circ \big(C(x)+R(x)+Z(x)\big). \tag{48}$$

Note that $\Lambda_r(x) \circ R(x) = A(x) \circ m(x)$ holds for some nonzero $A(x) \in F[x]_\circ$ because $\Lambda_r(R(\beta_\ell))=0$ for $\ell=0,\ldots,n-1$. On the other hand, we have $Z(x) = g(x) \circ m_S(x)$ for some $g(x)$ because $Z(\beta_\ell)=z_\ell=0$ for $\ell \notin \mathcal{Z}_c$. Thus, $m_S(x)$ right divides both $Z(x)$ and $\Lambda_r(x) \circ R(x)$. Note that $\deg_q m_S(x) = n-|\mathcal{Z}_c|$. If $|\mathcal{Z}_r|+|\mathcal{Z}_c| \le n-k$, then $k+|\mathcal{Z}_r| \le \deg_q m_S(x)$. Clearly, $P(x) = \Lambda_r(x) \circ C(x)$. □

If $\mathcal{Z}_c$ is empty, Proposition 13 reduces to Proposition 11; if $r=0$, $\Lambda_r(x) \triangleq x$ and Proposition 13 reduces to Proposition 12. Note that both Propositions 11 and 12 are generalizations of [52, Propositions 8 and 9], see also [53], [54], [55], [67], and [68].

VIII. Joint Decoding of Errors, Deviations, and Erasures

In this section, we address the problem of simultaneously correcting errors, deviations, and erasures, which appears in random network coding [7]. Joint decoding of errors and crisscrosses erasures is included as a special case.

A. Channel Model: Errors, Deviations, and Erasures

Let $\beta_0,\ldots,\beta_{n-1}$ be linearly independent elements of F_{q^L} over F_q, and let $m(x) \in F[x]_\circ$ be a nonzero polynomial of the smallest degree that vanishes on $\beta_0,\ldots,\beta_{n-1}$. Moreover, let $\mathcal{C}$ be an (n,k) Gabidulin code as in Section VI-A.

As in Section VII, a codeword $c \in \mathcal{C}$ may be viewed as a $L \times n$ matrix over F_q. By mapping each $c \in \mathcal{C}$ to a matrix X of the form $X=[I, c^{\mathsf{T}}]$ where I is a $n \times n$ identity matrix, a new code is obtained [7], which is commonly referred to as a lifted Gabidulin code.

Suppose that X is injected into a network applying network coding and the network is corrupted by errors, deviations, and erasures. Then by processing the received "packets" as in [7], we obtain

$$y = c + e + r + z, \tag{49}$$

where $e \in (F_{q^L})^n$ is an error, $r \in (F_{q^L})^n$ is a *deviation*, and $z \in (F_{q^L})^n$ is a special kind of erasure. The deviation r in (49) is a vector such that

$$r = a_r \cdot B, \tag{50}$$

where $a_r \in (F_{q^L})^{\mathrm{w_R}(r)}$ is some vector with rank weight $\mathrm{w_R}(r)$, and B is a $\mathrm{w_R}(r) \times n$ matrix over F_q; the vector a_r is known by the receiver, but B is unknown. The erasure z in (49) is a vector such that

$$z = a_z \cdot P, \tag{51}$$

where $a_z \in (F_{q^L})^{\mathrm{w_R}(z)}$ is some vector with rank weight $\mathrm{w_R}(z)$, and P is a $\mathrm{w_R}(z) \times n$ matrix over F_q; the vector a_z is unknown by the receiver, but P is known.

The decoding problem is to recover c from y with the side information a_r and P, which is referred to as the joint errors, deviations, and erasures decoding of Gabidulin codes. In the special case where (i) r in (50) is a rows erasure with known a_r as in Section VII-1 and (ii) P in (51) has only $\mathrm{w_R}(z)$ nonzero columns, then the problem reduces to joint errors and crisscrosses-erasures decoding as in [41], [60], and [61].

Below, we develop a new (simple) algorithm for this decoding problem, which is guaranteed to work correctly (i.e., to retrieve c) if

$$2\mathrm{w_R}(e) + \mathrm{w_R}(r) + \mathrm{w_R}(z) \le n-k. \tag{52}$$

B. Outline of the Proposed Decoding Algorithm

By turning P in (51) into a columns erasure, the decoding problem (as described above) can be solved by combining the decoding algorithm of Section VI-D with interpolation as in Section VII.

In outline, the proposed algorithm goes as follows:

1) Determine the set S according to (55) and compute $m_S(x)$ as described above Lemma 5.
2) Compute Λ_r as described above (59).
3) Compute $Y'(x)$ from (59).
   At this point, we have a transformed decoding problem with $m(x)$, $Y(x)$, $C(x)$, n, and k as in (53) replaced by $m_S(x)$, $Y'(x)$, $C'(x)$, n', and k' as in (61)–(63).

4) If $\deg_q Y'(x) < k'$, then set $C'(x) = Y'(x)$; otherwise, solve this transformed decoding problem by the algorithm of Section VI-D.
5) Compute $C(x) = C'(x)\ \mathrm{ldiv}_\circ\, \Lambda_r(x)$.

C. Details and Proof of the Proposed Decoding Algorithm

Let $Y(x) \triangleq \psi^{-1}(y)$, $C(x) \triangleq \psi^{-1}(c)$, $E(x) \triangleq \psi^{-1}(e)$, $R(x) \triangleq \psi^{-1}(r)$, and $Z(x) \triangleq \psi^{-1}(z)$ with ψ as in (32). Clearly,

$$Y(x) = C(x) + E(x) + R(x) + Z(x). \tag{53}$$

The task is to recover $C(x)$ from $Y(x)$.

If $z \neq 0$, then from the given P in (51), let T be an invertible $n \times n$ matrix over F_q such that

$$PT = P' \tag{54}$$

where P' is a $\mathrm{w_R}(z) \times n$ matrix over F_q with only $\mathrm{w_R}(z)$ nonzero columns. Note that T is not unique. If $z = 0$, then T is an identity matrix.

Then, we let $\mathcal{Z}_c \subset \{0, 1, \ldots, n-1\}$ denote the indices of nonzero columns of P' with $|\mathcal{Z}_c| = \mathrm{w_R}(z)$, and let

$$S \triangleq \{0, 1, \ldots, n-1\} \setminus \mathcal{Z}_c. \tag{55}$$

Moreover, let

$$(\beta'_0, \ldots, \beta'_{n-1}) \triangleq (\beta_0, \ldots, \beta_{n-1}) \cdot T \tag{56}$$

with T in (54).

Lemma 4: $Z(x) \triangleq \psi^{-1}(z)$ satisfies $Z(\beta'_\ell) = 0$ for all $\ell \in S$. □

Proof: For $z = (z_0, \ldots, z_{n-1}) \in (F_{q^L})^n$ in (51), let

$$\begin{aligned}(z'_0, \ldots, z'_{n-1}) &\triangleq (z_0, \ldots, z_{n-1}) \cdot T &(57)\\ &= a_z \cdot P' &(58)\end{aligned}$$

where the last step follows from (54). It is clear that $z'_\ell = 0$ for all $\ell \in S$. Note that $z'_\ell = Z(\beta'_\ell)$ for all $\ell \in \{0, \ldots, n-1\}$, which follows from (56), (57), $z_\ell = Z(\beta_\ell)$, and Proposition 2. The lemma then follows. □

For given S and $\{\beta'_0, \ldots, \beta'_{n-1}\}$ defined in (55) and (56), let $m_S(x) \in F[x]_\circ$ be a nonzero polynomial of the smallest degree that satisfies $m_S(\beta'_\ell) = 0$ for all $\ell \in S$. Note that $\deg_q m_S(x) = |S| = n - \mathrm{w_R}(z)$.

Lemma 5: $Z(x) = g(x) \circ m_S(x)$ for some $g(x) \in F[x]_\circ$. □

Proof: It follows from Lemma 4 and Proposition 5. □

Recall that $m(x) \in F[x]_\circ$ is a nonzero polynomial of the smallest degree that satisfies $m(\beta_\ell) = 0$ for $\ell = 0, \ldots, n-1$. If $z = 0$, then $Z(x) = 0$, $\beta'_\ell \triangleq \beta_\ell$, and $m_S(x) \triangleq m(x)$.

For given a_r in (50), let $\Lambda_r(x) \in F[x]_\circ$ be a nonzero polynomial of the smallest degree that vanishes on $\mathrm{span}(a_r)$. Note that $\deg_q \Lambda_r(x) = \mathrm{w_R}(r)$. If $r = 0$, $\Lambda_r(x) = x$.

Let

$$\begin{aligned}Y'(x) &\triangleq \big(\Lambda_r(x) \circ Y(x)\big)\ \mathrm{rmod}_\circ\, m_S(x) &(59)\\ E'(x) &\triangleq \big(\Lambda_r(x) \circ E(x)\big)\ \mathrm{rmod}_\circ\, m_S(x). &(60)\end{aligned}$$

Lemma 6: If (52) is satisfied, $Y'(x) = \Lambda_r(x) \circ C(x) + E'(x)$. □

Proof: Write $\Lambda_r(x) \circ Y(x) = \Lambda_r(x) \circ \big(C(x) + E(x) + R(x) + Z(x)\big)$. Note that $\Lambda_r(x) \circ R(x)$ satisfies $\Lambda_r(R(\beta_\ell)) = 0$ for all $\ell \in \{0, \ldots, n-1\}$ since $\Lambda_r(x)$ vanishes on $\mathrm{span}(r) \subseteq \mathrm{span}(a_r)$; we thus have $\Lambda_r(x) \circ R(x) = A(x) \circ m(x)$ for some nonzero $A(x)$ (by Proposition 5). Note that $m_S(x)$ right divides $m(x)$ since $\mathrm{span}(\{\beta'_\ell\}) \subset \mathrm{span}(\beta_0, \ldots, \beta_{n-1})$ from (56). It follows that $m_S(x)$ right divides $\Lambda_r(x) \circ R(x)$. Note also that $m_S(x)$ right divides $\Lambda_r(x) \circ Z(x)$ by Lemma 5. Finally, (52) implies that $k + \mathrm{w_R}(r) \leq n - \mathrm{w_R}(z)$; therefore, $\deg_q(\Lambda_r(x) \circ C(x)) < \deg_q m_S(x)$. The lemma follows. □

Lemma 7: For given S and $\{\beta'_0, \ldots, \beta'_{n-1}\}$ in (55) and (56), let $e' \in (F_{q^L})^{|S|}$ be a vector such that every component of e' corresponds to a $E'(\beta'_\ell)$ for every $\ell \in S$. Then, it holds that $\mathrm{w_R}(e') \leq \mathrm{w_R}(e)$. □

Proof: Let $\tilde{e} \in (F_{q^L})^{|S|}$ be a vector such that every component of $\tilde{e}$ corresponds to a $E(\beta'_\ell)$ for every $\ell \in S$. Since $\mathrm{span}(\{\beta'_\ell\}) \subseteq \mathrm{span}(\beta_0, \ldots, \beta_{n-1})$, clearly $\mathrm{span}(\tilde{e}) \subseteq \mathrm{span}(e)$ and thus $\mathrm{w_R}(\tilde{e}) \leq \mathrm{w_R}(e)$. Now, write $\Lambda_r(x) \circ E(x) = g(x) \circ m_S(x) + E'(x)$. Clearly $E'(\beta'_\ell) = \Lambda_r(E(\beta'_\ell))$ for all $\ell \in S$, which implies $\dim \mathrm{span}(e') \leq \dim \mathrm{span}(\tilde{e})$ (since $\Lambda_r(x) \in F[x]_\circ$ is a linear map over F_q). Therefore, $\mathrm{w_R}(e') \leq \mathrm{w_R}(\tilde{e})$. □

Now, let

$$\begin{aligned}C'(x) &\triangleq \Lambda_r(x) \circ C(x), &(61)\\ k' &\triangleq k + \mathrm{w_R}(r), &(62)\\ n' &\triangleq n - \mathrm{w_R}(z). &(63)\end{aligned}$$

By Lemma 6, (59) amounts to

$$Y'(x) = C'(x) + E'(x) \tag{64}$$

with $\deg_q C'(x) < k'$ and $\deg E'(x) < n' = \deg_q m_S(x)$. Note that condition (52) becomes $2\mathrm{w_R}(e) \leq n' - k'$; By Lemma 7. $E'(x)$ in (60) satisfies

$$2\mathrm{w_R}(e') \leq n' - k'. \tag{65}$$

If (52) is satisfied and if $e = 0$, then $E(x) = 0$ and thus $E'(x) = 0$. In this case, $Y'(x)$ in (59) equals $C'(x)$ in (61), i.e., $Y'(x) = C'(x)$ with $\deg Y'(x) < k'$.

If $\deg Y'(x) \geq k'$, recovering $C'(x)$ from $Y'(x)$ in (59) is a standard error-decoding problem as in Section VI, i.e., (64) plays the role of (36).

Theorem 4: Let $\Lambda_{e'}(x) \in F[x]_\circ$ be a nonzero polynomial of the smallest degree that vanishes on $E'(\beta'_\ell)$ for every $\ell \in S$. Propositions 9 and 10, Theorems 3, and Corollary 3 hold with $\Lambda_e(x)$, $E(x)$, $m(x)$, $Y(x)$, $C(x)$, n, and k replaced by $\Lambda_{e'}(x)$, $E'(x)$, $m_S(x)$, $Y'(x)$, $C'(x)$, n', and k', respectively. □

Proof: It follows from Lemmas 6 and 7, (64), and (65). □

D. Remarks

The problem of joint errors and crisscrosses-erasures decoding was first considered in [41], [60], and [61] and later generalized in [62] and [63], whose most general form was considered in [30], which coincides with the problem in [7], i.e., (49)–(51).

The decoding algorithms proposed in [7] and [30] are based on the Berlekamp–Massey algorithm (for linearized polynomials) and the Euclidean algorithm, respectively. The algorithm proposed in this section can be implemented with any of the Algorithms in Section IV and works for general $m(x)$.

The proposed decoding method (implemented with Algorithms 4–6) has complexity $O(n^2)$ over F_{q^L}, which is practical unless $n-k$ is large. If $n-k$ is very large[2], sub-quadratic decoding can be obtained by using, e.g., the asymptotically faster operations in [64] for computing $\Lambda_r(x)$, $Y(x)$, and $m_S(x)$ in (59), cf. Table I of [64], together with an asymptotically faster Euclidean algorithm (cf. Section IV-B).

IX. Conclusion

We have generalized the partial-inverse approach from ordinary polynomials to linearized polynomials, and developed its application to Gabidulin codes and to lifted Gabidulin codes in random network coding. As with ordinary polynomials, the basic partial-inverse algorithm is a common parent algorithm of the (reverse) Berlekamp–Massey algorithm, the Euclidean algorithm, and the quotient-saving algorithm, and the partial-inverse problem is a natural common interface between these algorithms and their applications to decoding. The generalization to linearized polynomials was mostly straightforward, but new arguments were required for several key points. The basic partial-inverse algorithm itself was proved by a new and simpler proof, which also works for ordinary polynomials.

Decoding Gabidulin codes has been reduced to the partial-inverse problem via a key equation with a new converse. We have also developed new algorithms for interpolating crisscross erasures and for joint decoding of errors, erasures, and deviations in random network coding.

Appendix A
Proof of Theorem 1

The following proof of Theorem 1 adapts the proof of Propositions 1–3 of [52] to linearized polynomials.

Proof of Theorem 1: We first prove uniqueness. Assume that $\Lambda^{(1)}(x)$ and $\Lambda^{(2)}(x)$ are two solutions of the problem, which implies $\deg_q \Lambda^{(1)}(x) = \deg_q \Lambda^{(2)}(x) \geq 0$. Define

$$r^{(1)}(x) \triangleq \big(\Lambda^{(1)}(x) \circ b(x)\big) \bmod_\circ m(x) \tag{66}$$

$$r^{(2)}(x) \triangleq \big(\Lambda^{(2)}(x) \circ b(x)\big) \bmod_\circ m(x) \tag{67}$$

and consider

$$\Lambda(x) \triangleq \Big(\operatorname{lcf} \Lambda^{(2)}(x)\Big)\Lambda^{(1)}(x) - \Big(\operatorname{lcf} \Lambda^{(1)}(x)\Big)\Lambda^{(2)}(x). \tag{68}$$

Then

$$r(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x) \tag{69}$$

$$= \Big(\operatorname{lcf} \Lambda^{(2)}(x)\Big) r^{(1)}(x) - \Big(\operatorname{lcf} \Lambda^{(1)}(x)\Big) r^{(2)}(x). \tag{70}$$

[2]Asymptotically faster implementation may require n to be large. E.g., by Remark 9 of [64], using the composition in Algorithm 1 of [64] (with Strassen's multiplication) has complexity $O(n^{1.91})$, which can be faster than the naive implementation $O(n^2)$ when $n \geq 7225$.

Clearly, (70) implies that $\Lambda(x)$ also satisfies (6). But (68) implies $\deg_q \Lambda(x) < \deg_q \Lambda^{(1)}(x)$, which is a contradiction unless $\Lambda(x) = 0$. Thus $\Lambda(x) = 0$, which means that $\Lambda^{(1)}(x)$ and $\Lambda^{(2)}(x)$ are equal up to a scale factor.

It remains to prove existence and the degree bound. The case $d = \deg_q m(x)$ is obvious, for which $\Lambda(x) = x$ will do. Otherwise, let $n \triangleq \deg_q m(x)$ and $\nu \triangleq \deg_q m(x) - d > 0$. For fixed $b(x)$ and $m(x) \neq 0 \in F[x]_\circ$, consider the mapping

$$F_{q^L}^{\nu+1} \to F_{q^L}^{\nu} \tag{71}$$

given by

$$\begin{aligned}(\Lambda_0, \ldots, \Lambda_\nu) &\mapsto \Lambda(x) \triangleq \Lambda_0 x^{[0]} + \ldots, +\Lambda_\nu x^{[\nu]} \\ &\mapsto r(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x) \\ &\mapsto (r_0, \ldots, r_{n-1}) \\ &\mapsto (r_d, \ldots, r_{n-1}).\end{aligned} \tag{72}$$

Clearly, this mapping is linear over F_{q^L} by Proposition 1, and it has a nontrivial kernel by (71). But any nonzero element in the kernel corresponds (by (72)) to a nonzero $\Lambda(x)$ that satisfies (6) and (7). □

Appendix B
More About the Partial-Inverse Problem

This section generalizes the results in [52, Section III] to linearized polynomials.

A. Minimal Partial Inverses

Definition (Minimal Partial Inverse): For fixed nonzero $b(x)$ and $m(x) \in F[x]_\circ$ with $\deg_q b(x) < \deg_q m(x)$, a nonzero polynomial $\Lambda(x) \in F[x]_\circ$ is a *minimal partial inverse of* $b(x)$ *rmod*$_\circ$ $m(x)$ *(with respect to composition* $\circ$*)* if every nonzero $\Lambda^{(1)}(x) \in F[x]_\circ$ with

$$\begin{aligned}\deg_q\Big(\big(\Lambda^{(1)}(x) \circ b(x)\big) \bmod_\circ m(x)\Big) \\ \leq \deg_q\Big(\big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x)\Big)\end{aligned} \tag{73}$$

satisfies $\deg_q \Lambda^{(1)}(x) \geq \deg_q \Lambda(x)$. □

In the following, we often refer to the partial-inverse problem as defined in Section III. Let $\Lambda(x) = \Lambda_{\text{null}}(x)$ be the solution of the partial-inverse problem for $d = 0$.

Proposition 14 (Minimal Partial Inverses Solve Partial-Inverse Problems): The solution $\Lambda(x)$ of the partial-inverse problem is a minimal partial inverse of $b(x)$ $\bmod_\circ$ $m(x)$.

Conversely, $\Lambda(x) \triangleq \Lambda_{\text{null}}(x)$ solves the partial-inverse problem with $d = 0$; every other minimal partial inverse $\Lambda(x)$ of $b(x)$ $\bmod_\circ$ $m(x)$ solves the partial-inverse problem with

$$d = \deg_q\Big(\big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x)\Big) + 1. \tag{74}$$

□

Proposition 14 corresponds to [52, Proposition 4] and can be proved in the same way.

Proposition 15 (Minimal Partial Inverses of the Same Degree Are Unique): For fixed nonzero $b(x)$ and $m(x) \in F[x]_\circ$ with $\deg_q b(x) < \deg_q m(x)$, let $\Lambda^{(1)}(x)$ and $\Lambda^{(2)}(x)$ be two minimal partial inverses of $b(x)$ with

$\deg_q \Lambda^{(1)}(x) = \deg_q \Lambda^{(2)}(x)$. Then $\Lambda^{(1)}(x) = \alpha\Lambda^{(2)}(x)$ for some nonzero $\alpha \in F_{q^L}$. □

Proposition 15 corresponds to [52, Proposition 5] and can be proved in the same way.

In consequence, we have

Proposition 16 (Chain of Minimal Partial Inverses): Let $\Lambda^{(1)}(x), \ldots, \Lambda^{(K)}(x)$ denote *all* the minimal partial inverses of $b(x)$, and let $r^{(i)}(x) \triangleq \big(\Lambda^{(i)}(x) \circ b(x)\big) \operatorname{rmod}_\circ m(x)$. Then, $\deg_q \Lambda^{(K)}(x) > \ldots > \deg_q \Lambda^{(1)}(x)$ if and only if $\deg_q r^{(K)}(x) < \ldots < \deg_q r^{(1)}(x)$. □

B. Degree Change Property

Let $\Lambda^{(1)}(x) \triangleq x$, and let $\Lambda^{(1)}(x), \ldots, \Lambda^{(K)}(x)$ denote all the minimal partial inverses of $b(x)$ ($\operatorname{rmod}_\circ m(x)$) as in Proposition 16. Moreover, let

$$g^{(i)}(x) \triangleq \big(\Lambda^{(i)}(x) \circ b(x)\big) \operatorname{rdiv}_\circ m(x) \tag{75}$$

$$r^{(i)}(x) \triangleq \big(\Lambda^{(i)}(x) \circ b(x)\big) \operatorname{rmod}_\circ m(x) \tag{76}$$

for $i = 1, \ldots, K$. Note that $\Lambda^{(1)}(x) \triangleq x$ and $r^{(1)}(x) = b(x)$; also, $\deg_q r^{(K)}(x) < \ldots < \deg_q r^{(1)}(x)$.

Now, we define $r^{(0)}(x) \triangleq m(x)$ and let

$$\Delta_i \triangleq \deg_q r^{(i-1)}(x) - \deg_q r^{(i)}(x) \tag{77}$$

for $i = 1, \ldots, K$. Clearly, $\Delta_1 = \deg_q m(x) - \deg_q b(x)$, and $\Delta_i > 0$ for all $i \in \{1, \ldots, K\}$.

Lemma 8: If some nonzero $\Lambda(x)$ satisfies

$$\deg_q\Big(\big(\Lambda(x) \circ b(x)\big) \operatorname{rmod}_\circ m(x)\Big) < \deg_q r^{(i-1)}(x) \tag{78}$$

for some $i \in \{1, \ldots, K\}$, then $\deg_q \Lambda(x) \geq \deg_q \Lambda^{(i)}(x)$. If in addition $\Lambda(x)$ satisfies

$$\deg_q \Lambda(x) < \deg_q \Lambda^{(i)}(x) + \Delta_i, \tag{79}$$

then the following (80)–(82) hold for some $a(x) \in F[x]_\circ$

$$\Lambda(x) = a(x) \circ \Lambda^{(i)}(x) \tag{80}$$

$$r(x) = a(x) \circ r^{(i)}(x) \tag{81}$$

$$g(x) = a(x) \circ g^{(i)}(x) \tag{82}$$

where $g(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \operatorname{rdiv}_\circ m(x)$, and where $r(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \operatorname{rmod}_\circ m(x)$. □

Proof: It is obvious that $\deg_q \Lambda(x) \geq \deg_q \Lambda^{(i)}(x)$ since $\Lambda^{(i)}(x)$ is a nonzero polynomial of the smallest degree that satisfies (6) for $d = \deg_q r^{(i-1)}(x)$.

Now, we write $\Lambda(x) \circ b(x) = g(x) \circ m(x) + r(x)$, and write $\Lambda(x) = a(x) \circ \Lambda^{(i)}(x) + e(x)$ for some (unique) $a(x)$ and $e(x)$ with $\deg_q e(x) < \deg_q \Lambda^{(i)}(x)$. We then have

$$\big(a(x) \circ \Lambda^{(i)}(x) + e(x)\big) \circ b(x) = g(x) \circ m(x) + r(x) \tag{83}$$

and therefore

$$a(x) \circ \Lambda^{(i)}(x) \circ b(x) + e(x) \circ b(x) = g(x) \circ m(x) + r(x). \tag{84}$$

Then from $\Lambda^{(i)}(x) \circ b(x) = g^{(i)}(x) \circ m(x) + r^{(i)}(x)$, we obtain

$$\begin{aligned} e(x) \circ b(x) &= \big(g(x) - a(x) \circ g^{(i)}(x)\big) \circ m(x) \\ &\quad + r(x) - a(x) \circ r^{(i)}(x). \end{aligned} \tag{85}$$

The stated assumption (79) implies that $\deg_q a(x) < \Delta_i$, and therefore $\deg_q\big(a(x) \circ r^{(i)}(x)\big) < \deg_q r^{(i-1)}(x)$ by (77). Note also that $\deg_q r(x) < \deg_q r^{(i-1)}(x)$ by (78).

It follows that $\deg_q\big(r(x) - a(x) \circ r^{(i)}(x)\big) < \deg_q r^{(i-1)}(x)$ and therefore

$$\deg_q\big(e(x) \circ b(x) \operatorname{rmod}_\circ m(x)\big) < \deg_q r^{(i-1)}(x). \tag{86}$$

But $\Lambda^{(i)}(x)$ is a nonzero polynomial of the smallest degree that satisfies (6) for $d = \deg_q r^{(i-1)}(x)$, and therefore (86) is not tenable unless $e(x) = 0$. Therefore $e(x) = 0$, and we obtain (80)–(82). □

Proposition 17: Let $\Lambda^{(0)}(x) \triangleq 0$. For $i = 1, \ldots, K-1$, let $p^{(i)}(x) \triangleq r^{(i-1)}(x) \operatorname{rdiv}_\circ r^{(i)}(x)$. Then, it holds that $\Lambda^{(i+1)}(x) = \Lambda^{(i-1)}(x) - p^{(i)}(x) \circ \Lambda^{(i)}(x)$. □

Proof: Lemma 8 implies that $\deg_q \Lambda^{(i+1)}(x) \geq \deg_q \Lambda^{(i)}(x) + \Delta_i$. Indeed, if $\deg_q \Lambda^{(i+1)}(x) < \deg_q \Lambda^{(i)}(x) + \Delta_i$, then by Lemma 8, $r^{(i+1)}(x) = a(x) \circ r^{(i)}(x)$, which contradicts the fact $\deg_q r^{(i+1)}(x) < \deg_q r^{(i)}(x)$.

Now, let $\Lambda(x) \triangleq \Lambda^{(i-1)}(x) - p^{(i)}(x) \circ \Lambda^{(i)}(x)$ Clearly, $\deg_q \Lambda(x) = \deg_q \Lambda^{(i)}(x) + \Delta_i$ since $\deg_q p^{(i)}(x) = \Delta_i$. Note that $\Lambda(x)$ satisfies $\deg_q\big(\big(\Lambda(x) \circ b(x)\big) \operatorname{rmod}_\circ m(x)\big) < \deg_q r^{(i)}(x)$. It turns out that $\Lambda^{(i+1)}(x) = \Lambda(x)$. □

Proposition 18 (Degree Change Property): For every $i \in \{1, \ldots, K\}$, it holds that

$$\deg_q \Lambda^{(i)}(x) = \deg_q m(x) - \deg_q r^{(i-1)}(x). \tag{87}$$

□

The proof can be based on Lemma 2 or on Lemma 8.

Proof of Proposition 18: First, we note that (87) holds for $i = 1$ since $\deg_q \Lambda^{(1)}(x) = 0$ and $r^{(0)}(x) \triangleq m(x)$. Now, we assume that (87) holds for $i = j$, $1 \leq j \leq K-1$; we will prove that (87) holds for $i = j+1$. From Proposition 17, we have

$$\begin{aligned} \deg_q \Lambda^{(j+1)}(x) &= \deg_q \Lambda^{(j)}(x) + \Delta_j \\ &= \deg_q m(x) - \deg_q r^{(j)}(x), \end{aligned} \tag{88}$$

where the last step follows from (87) with $i = j$. □

C. Degree Bound With a Converse

The following Theorem 5 is an amalgam of Lemma 8 and Proposition 18. Theorem 5 will be used to prove (the converse part of) Theorem 6, which in turn will be used to prove Theorem 7.

Theorem 5: For fixed d with $0 \leq d \leq \deg_q m(x)$, let $\Lambda'(x)$ be the solution of the partial-inverse problem, and let

$$r'(x) \triangleq \big(\Lambda'(x) \circ b(x)\big) \operatorname{rmod}_\circ m(x), \tag{89}$$

$$g'(x) \triangleq \big(\Lambda'(x) \circ b(x)\big) \operatorname{rdiv}_\circ m(x). \tag{90}$$

If some nonzero $\Lambda(x)$ and $r(x) \triangleq (\Lambda(x) \circ b(x)) \bmod_\circ m(x)$ satisfy

$$\deg_q r(x) < d \tag{91}$$

and

$$\deg_q \Lambda(x) \le \deg_q m(x) - d, \tag{92}$$

then the following (93)–(95) hold

$$\begin{aligned} \Lambda(x) &= a(x) \circ \Lambda'(x) && (93)\\ r(x) &= a(x) \circ r'(x) && (94)\\ g(x) &= a(x) \circ g'(x) && (95) \end{aligned}$$

where $a(x)$ is any nonzero polynomial such that (91) and (92) hold, and where $g(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \operatorname{rdiv}_\circ m(x)$. □

Proof: We will prove the theorem via Lemma 8 and Proposition 18. Let $\Lambda^{(1)}(x) \triangleq x$ and let $\Lambda^{(1)}(x), \Lambda^{(2)}(x), \ldots, \Lambda^{(K)}(x)$ denote all the minimal partial inverses of $b(x) \bmod_\circ m(x)$. Moreover, let $g^{(i)}(x)$, $r^{(i)}(x)$, and Δ_i denote all the corresponding quantities as defined in (75)–(77).

Then by Proposition 14, $\Lambda'(x)$ equals $\Lambda^{(i)}(x)$ for some $i \in \{1, \ldots, K\}$ (up to a scale factor), and $\deg_q r'(x) = \deg_q r^{(i)}(x) < d$.

Since $d \le \deg_q r^{(i-1)}(x)$, (91) implies that $\deg_q r(x) < \deg_q r^{(i-1)}(x)$, which agrees with (78). We next note from (92) and $\deg_q r^{(i)}(x) < d$ that

$$\begin{aligned} \deg_q \Lambda(x) &< \deg_q m(x) - \deg_q r^{(i)}(x) && (96)\\ &= \deg_q \Lambda^{(i)}(x) + \Delta_i && (97) \end{aligned}$$

where the last step follows from (77) and from $\deg_q m(x) = \deg_q \Lambda^{(i)}(x) + \deg_q r^{(i-1)}(x)$ by Proposition 18. The theorem then follows from Lemma 8. □

Theorem 6 (Degree Bound with a Converse): For fixed nonzero $b(x)$ and $m(x) \in F[x]_\circ$ with $\deg_q b(x) < \deg_q m(x)$, a nonzero $\Lambda(x) \in F[x]_\circ$ is a minimal partial inverse of $b(x)$ if and only if both

$$\deg_q \Lambda(x) + \deg_q \Big(\big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x) \Big) < \deg_q m(x) \tag{98}$$

and

$$\operatorname{lgcd}\big(\Lambda(x), g(x)\big) = x, \tag{99}$$

where $g(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \operatorname{rdiv}_\circ m(x)$. □

In the special case where $g(x) = 0$, (99) requires $\Lambda(x) = \alpha x$ for some nonzero $\alpha \in F_{q^L}$. Theorem 6 is a generalization of [52, Theorem 1] to linearized polynomials, which will be needed for the proof of Theorem 7. (The converse part of the proof in [52, Theorem 1] does not seem to generalize to linearized polynomials.)

Proof of Theorem: 6: Let

$$\begin{aligned} r(x) &\triangleq \big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x) && (100)\\ &= \Lambda(x) \circ b(x) - g(x) \circ m(x). && (101) \end{aligned}$$

For the direct part, assume that $\Lambda(x)$ is a minimal partial inverse of $b(x)$. Then (98) is immediate from Proposition 14 and (7) of Theorem 1. As for (99), assume that $\operatorname{lgcd}\big(\Lambda(x), g(x)\big) = c(x)$ with $\deg_q c(x) > 0$, i.e.,

$$\Lambda(x) = c(x) \circ \Lambda'(x) \tag{102}$$

for some $\Lambda'(x) \in F[x]_\circ$, and $g(x) = c(x) \circ g'(x)$ for some $g'(x) \in F[x]_\circ$. From (101), we then have

$$r(x) = c(x) \circ \big(\Lambda'(x) \circ b(x) - g'(x) \circ m(x)\big). \tag{103}$$

It follows that $c(x)$ left divides $r(x)$, i.e.,

$$r(x) = c(x) \circ r'(x) \tag{104}$$

for some $r'(x) \in F[x]_\circ$, and thus

$$\begin{aligned} r'(x) &= \Lambda'(x) \circ b(x) - g'(x) \circ m(x) && (105)\\ &= \big(\Lambda'(x) \circ b(x)\big) \bmod_\circ m(x). && (106) \end{aligned}$$

But $\deg_q \Lambda'(x) < \deg_q \Lambda(x)$ and $\deg_q r'(x) < \deg_q r(x)$, which is impossible because $\Lambda(x)$ is a minimal partial inverse.

For the converse part, we assume that some nonzero $\Lambda(x)$ satisfies (98) and (99). Let $r(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \bmod_\circ m(x)$ and $d \triangleq \deg_q r(x) + 1$. Then by (98), $\deg_q \Lambda(x) + d \le \deg_q m(x)$. It then follows from Theorem 5 that

$$\Lambda(x) = a(x) \circ \Lambda'(x) \text{ and } g(x) = a(x) \circ g'(x) \tag{107}$$

where $\Lambda'(x)$ is the solution of the partial-inverse problem with $d = \deg_q r(x) + 1$. But $\Lambda(x)$ and $g(x)$ satisfy (99), which implies $a(x) = \gamma x$ for some nonzero $\gamma \in F_{q^L}$. We therefore have $\Lambda(x) = \gamma \Lambda'(x)$, which is a minimal partial inverse, cf. Proposition 14. □

Appendix C
Monomialized Partial-Inverse Problem

Consider a partial-inverse problem with general $m(x)$ (as stated in Section III) and $d < \deg_q m(x)$. Let $n \triangleq \deg_q m(x)$ and $\tau \triangleq n - d > 0$. Further, let

$$w(x) \triangleq x^{[n+2\tau-1]} \operatorname{ldiv}_\circ m(x) \tag{108}$$

and

$$\tilde{b}(x) \triangleq \big(b(x) \circ w(x)\big) \operatorname{rdiv}_\circ x^{[n-1]}. \tag{109}$$

Theorem 7 (Monomialized Partial-Inverse Problem): The partial-inverse problem with general $m(x)$ and $d < \deg_q m(x)$ can be transformed into another partial-inverse problem where (6) is replaced by

$$\deg_q \Big(\big(\Lambda(x) \circ \tilde{b}(x)\big) \bmod_\circ x^{[2\tau]} \Big) < \tau \tag{110}$$

with $\tilde{b}(x)$ defined in (109). The modified problem (110) has the same solution $\Lambda(x)$ as the original problem and we have $\big(\Lambda(x) \circ b(x)\big) \operatorname{rdiv}_\circ m(x) = \big(\Lambda(x) \circ \tilde{b}(x)\big) \operatorname{rdiv}_\circ x^{[2\tau]}$. □

Note that $w(x)$ in (108) can be precomputed and $\deg_q w(x) = 2\tau - 1$. Note also that $\deg_q \tilde{b}(x) < 2\tau$.

Proof of Theorem 7: Let $\Lambda(x)$ be the solution of the original partial-inverse problem (which is unique up to a nonzero

scale factor), and let $r(x) \triangleq \big(\Lambda(x) \circ b(x)\big) \bmod_{\circ} m(x)$, where $\deg_q r(x) < d$. We then write

$$\Lambda(x) \circ b(x) = g(x) \circ m(x) + r(x) \tag{111}$$

for some (unique) $g(x)$ with

$$\deg_q g(x) < \deg_q \Lambda(x) \leq \tau, \tag{112}$$

where the second inequality follows from Theorem 1. Note that $\operatorname{lgcd}\big(\Lambda(x), g(x)\big) = x$ by Theorem 6 and Proposition 14.

Multiplying both sides of (111) by $w(x)$ defined in (108), we obtain

$$\Lambda(x) \circ b(x) \circ w(x) = (g(x) \circ m(x) + r(x)) \circ w(x). \tag{113}$$

Note that $x^{[n+2\tau-1]} = m(x) \circ w(x) + e(x)$ for some $e(x) \in F[x]_{\circ}$ with $\deg_q e(x) < n$, and thus

$$g(x) \circ m(x) \circ w(x) = g(x) \circ \big(x^{[n+2\tau-1]} - e(x)\big), \tag{114}$$

and $\deg_q\big(g(x) \circ e(x)\big) < \tau + n - 1$ from (112). Note also that $r(x) \circ w(x)$ in (113) satisfies $\deg_q\big(r(x) \circ w(x)\big) < d + 2\tau - 1 = n + \tau - 1$. Equation (113) can therefore be written as

$$\Lambda(x) \circ b(x) \circ w(x) = g(x) \circ x^{[n+2\tau-1]} + \tilde{r}(x) \tag{115}$$

with $\tilde{r}(x) \triangleq -g(x) \circ e(x) + r(x) \circ w(x)$ and $\deg_q \tilde{r}(x) < n + \tau - 1$.

We next note that $b(x) \circ w(x) = \tilde{b}(x) \circ x^{[n-1]} + e_b(x)$ for some $e_b(x)$ with $\deg_q e(x) < n - 1$ from (109), and therefore

$$\Lambda(x) \circ b(x) \circ w(x) = \Lambda(x) \circ (\tilde{b}(x) \circ x^{[n-1]} + e_b(x)). \tag{116}$$

With $\tilde{e}(x) \triangleq \Lambda(x) \circ e_b(x)$, we then have

$$\Lambda(x) \circ b(x) \circ w(x) = \Lambda(x) \circ \tilde{b}(x) \circ x^{[n-1]} + \tilde{e}(x) \tag{117}$$

and $\deg_q \tilde{e}(x) < \tau + n - 1$. From (115) and (117), we obtain $\Lambda(x) \circ \tilde{b}(x) \circ x^{[n-1]} + \tilde{e}(x) = g(x) \circ x^{[n+2\tau-1]} + \tilde{r}(x)$, and therefore

$$\Lambda(x) \circ \tilde{b}(x) = g(x) \circ x^{[2\tau]} + r_b(x) \tag{118}$$

where $r_b(x) \triangleq \big(\tilde{r}(x) - \tilde{e}(x)\big) \operatorname{rdiv}_{\circ} x^{[n-1]}$. Note that $\deg_q r_b(x) < \tau$. Clearly, (110) holds, and by Theorem 6, $\Lambda(x)$ is a minimal partial inverse of $\tilde{b}(x)$ (with respect to $x^{[2\tau]}$).

We still have to show that $\Lambda(x)$ is the solution of the partial-inverse problem (110). In the following, we prove this fact by contradiction. Assume that $\Lambda^{(1)}(x)$ is the solution of the partial-inverse problem (110); we therefore have

$$\deg_q \Lambda^{(1)}(x) \leq \deg_q \Lambda(x) \tag{119}$$

and $r^{(1)}(x) \triangleq \big(\Lambda^{(1)}(x) \circ \tilde{b}(x)\big) \bmod_{\circ} x^{[2\tau]}$ satisfies $\deg_q r^{(1)}(x) < \tau$. By Proposition 14, $\Lambda^{(1)}(x)$ is a minimal partial inverse of $\tilde{b}(x)$ (with respect to $x^{[2\tau]}$). But $\Lambda(x)$ is also minimal partial inverse of $\tilde{b}(x)$. If $\deg_q \Lambda(x) > \deg_q \Lambda^{(1)}(x)$, then by Proposition 18 (with $m(x) = x^{[2\tau]}$ and $r^{(i-1)}(x) = r^{(1)}(x)$) we have $\deg_q \Lambda(x) \geq 2\tau - \deg_q r^{(1)}(x) > \tau$, which contradicts (112). Therefore, (119) holds only for $\deg_q \Lambda^{(1)}(x) = \deg_q \Lambda(x)$. It then follows from Proposition 15 that $\Lambda(x) = \alpha \Lambda^{(1)}(x)$ for some $\alpha \in F_{q^L}$.

Finally, from (111) and (118), we clearly have $\big(\Lambda(x) \circ b(x)\big) \operatorname{rdiv}_{\circ} m(x) = \big(\Lambda(x) \circ \tilde{b}(x)\big) \operatorname{rdiv}_{\circ} x^{[2\tau]}$. □

A. Remarks

This section generalizes most of the results in [52, Section III] to linearized polynomials and contains also some results without a counterpart in [52]. In particular, Lemma 8, Proposition 17, and Theorem 5 are new. Proposition 18, Theorem 6, and Theorem 7 generalize their counterparts in [52], but their proofs require different arguments than in [52].

Appendix D
Minimum Rank Distance

Let $\mathcal{C}$ be a code as in (34). The following fact was proved in [3].

Proposition 19 (Minimum Rank Distance of Gabidulin Codes): The minimum rank distance of $\mathcal{C}$

$$\mathrm{d_R}(\mathcal{C}) \triangleq \min\{\mathrm{w_R}(c - c') : c, c' \in \mathcal{C}, c \neq c'\} \tag{120}$$

is

$$\begin{aligned} \mathrm{d_R}(\mathcal{C}) &= \min\{\mathrm{w_R}(c) : c \in \mathcal{C}, c \neq 0\} &(121)\\ &= n - k + 1. &(122)\end{aligned}$$

□

An alternative proof was given in [30]. Yet another proof goes as follows.

Proof of Proposition 19: Eq. (121) is clear since $\mathcal{C}$ is a linear code. It remains to prove (122). By Proposition 7, for any nonzero $c \in \mathcal{C}$, $\psi^{-1}(c)$ vanishes on a subspace of dimension $n - \mathrm{w_R}(c)$. It then follows (by Corollary 2) that $\deg_q \psi^{-1}(c) \geq n - \mathrm{w_R}(c)$. But $\deg_q \psi^{-1}(c) < k$, and thus $\mathrm{w_R}(c) > n - k$.

On the other hand, we have

$$\mathrm{d_R}(\mathcal{C}) \leq n - k + 1 \tag{123}$$

from the Singleton bound for linearized polynomials [2], [3]; alternatively, (123) follows from the existence of $a(x) \in F[x]_{\circ}$ with $\deg_q a(x) = k - 1$ that has exactly $k - 1$ zeros in $\{\beta_0, \ldots, \beta_{n-1}\}$ (cf. Alg. 1), i.e., $\mathrm{w_R}(a) \leq n - (k - 1)$. □

Acknowledgment

The comments by the anonymous reviewers have been very helpful for improving the presentation of this article.

References

[1] J.-H. Yu and H.-A. Loeliger, "Decoding Gabidulin codes via partial inverses of linearized polynomials," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 2059–2063.

[2] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. Combinat. Theory, A*, vol. 25, pp. 226–241, Nov. 1978.

[3] È. M. Gabidulin, "Theory of codes with maximum rank distance," *Problems Inf. Transmiss.*, vol. 21, no. 1, pp. 1–12, 1985.

[4] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 328–336, Mar. 1991.

[5] D. Silva and F. R. Kschischang, "Rank-metric codes for priority encoding transmission with network coding," in *Proc. 10th Can. Workshop Inf. Theory (CWIT)*, Jun. 2007, pp. 81–84.

[6] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[7] D. Silva, F. R. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.

[8] D. Silva and F. R. Kschischang, "Fast encoding and decoding of Gabidulin codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Seoul, South Korea, Jun. 2009, pp. 2858–2862.

[9] U. Martínez-Peñas, "Skew and linearized Reed–Solomon codes and maximum sum rank distance codes over any division ring," *J. Algebr.*, vol. 504, pp. 587–612, Jun. 2018.

[10] U. Martinez-Penas and F. R. Kschischang, "Reliable and secure multishot network coding using linearized Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 8, pp. 4785–4803, Aug. 2019.

[11] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[12] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[13] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[14] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE Int. Symp. Inf. Theory*, Yokohama, Japan, Jun./Jul. 2003, p. 442.

[15] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Annu. Allerton Conf. Commun. Control Comput.*, Monticello, IL, USA Oct. 2003, pp. 40–49.

[16] T. Ho et al., "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[17] D. Silva and F. R. Kschischang, "Universal secure network coding via rank-metric codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1124–1135, Feb. 2011.

[18] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, "Ideals over a non-commutative ring and their application in cryptology," in *Eurocrypt 1991: Advances in Cryptology*. Brighton, U.K.: Springer, Apr. 1991, pp. 482–489.

[19] K. Gibson, "The security of the Gabidulin public key cryptosystem," in *Eurocrypt 1996: Advances in Cryptology*. Saragossa, Spain: Springer, May 1996, pp. 212–223.

[20] P. Lusina, È. M. Gabidulin, and M. Bossert, "Maximum rank distance codes as space-time codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2757–2760, Oct. 2003.

[21] H.-F. Lu and P. V. Kumar, "A unified construction of space-time codes with optimal rate-diversity tradeoff," *IEEE Trans. Inf. Theory*, vol. 51, no. 5, pp. 1709–1730, May 2005.

[22] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *J. Soc. Ind. Appl. Math.*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[23] O. Ore, "On a special class of polynomials," *Trans. Amer. Math. Soc.*, vol. 35, no. 3, pp. 559–584, Jul. 1933.

[24] R. W. Nobrega and B. F. Uchoa-Filho, "Multishot codes for network coding using rank-metric codes," in *Proc. 3rd IEEE Int. Workshop Wireless Netw. Coding*, Jun. 2010, pp. 1–6.

[25] O. Ore, "Theory of non-commutative polynomials," *Ann. Math.*, vol. 34, no. 3, pp. 480–508, 1933.

[26] D. Boucher and F. Ulmer, "Linear codes using skew polynomials with automorphisms and derivations," *Des., Codes Cryptogr.*, vol. 70, no. 3, pp. 405–431, 2014.

[27] S. Liu, F. Manganiello, and F. R. Kschischang, "Construction and decoding of generalized skew-evaluation codes," in *Proc. IEEE 14th Can. Workshop Inf. Theory (CWIT)*, St. John's, NL, Canada, Jul. 2015, pp. 9–13.

[28] D. Boucher, "An algorithm for decoding skew Reed–Solomon codes with respect to the skew metric," in *Proc. Workshop Coding Cryptogr.*, 2019, pp. 1991–2005.

[29] A. Wachter, V. Sidorenko, M. Bossert, and V. Zyablov, "Partial unit memory codes based on Gabidulin codes," in *Proc. IEEE Int. Symp. Inf. Theory Proc.*, Saint Petersburg, Russia, Jul. 2011, pp. 2487–2491.

[30] A. Wachter-Zeh, "Decoding of block and convolutional codes in rank metric," Ph.D. dissertation, Inst. Commun. Eng., Univ. Ulm, Ulm, Germany, 2013.

[31] A. Wachter-Zeh, M. Stinner, and V. Sidorenko, "Convolutional codes in rank metric with application to random network coding," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3199–3213, Jun. 2015.

[32] R. Mahmood, A. Badr, and A. Khisti, "Convolutional codes with maximum column sum rank for network streaming," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3039–3052, Jun. 2016.

[33] R. Mahmood, A. Badr, and A. Khisti, "Streaming codes for multiplicative-matrix channels with burst rank loss," *IEEE Trans. Inf. Theory*, vol. 64, no. 7, pp. 5296–5311, Jul. 2018.

[34] D. Napp, R. Pinto, and V. Sidorenko, "Concatenation of convolutional codes and rank metric codes for multi-shot network coding," *Des., Codes Cryptogr.*, vol. 86, no. 2, pp. 303–318, Feb. 2018.

[35] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde, "Fast decoding of codes in the rank, subspace, and sum-rank metric," *IEEE Trans. Inf. Theory*, vol. 67, no. 8, pp. 5026–5050, Aug. 2021.

[36] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equation for decoding Goppa codes," *Inf. Control*, vol. 27, pp. 87–99, Jan. 1975.

[37] A. Wachter-Zeh, V. Afanassiev, and V. Sidorenko, "Fast decoding of Gabidulin codes," *Des., Codes Cryptogr.*, vol. 66, pp. 57–73, Jan. 2013.

[38] A. Shiozaki, "Decoding of redundant residue polynomial codes using Euclid's algorithm," *IEEE Trans. Inf. Theory*, vol. IT-34, no. 5, pp. 1351–1354, Sep. 1988.

[39] S. Gao, "A new algorithm for decoding Reed–Solomon codes," in *Communications, Information and Network Security*, vol. 712, V. Bhargava, H. V. Poor, V. Tarokh, and S. Yoon, Eds. Norwell, MA, USA: Kluwer, 2003, pp. 55–68.

[40] A. V. Paramonov and O. V. Tretjakov, "An analogue of Berlekamp–Massey algorithm for decoding codes in rank metric," presented at the Moscow Inst. Phys. Technol. (MIPT), Moscow, Russia, 1991.

[41] G. Richter and S. Plass, "Error and erasure decoding of rank-codes with a modified Berlekamp–Massey algorithm," in *Proc. ITG Conf. Source Channel Coding*, Erlangen, Germany, Jan. 2004, pp. 249–256.

[42] V. Sidorenko, G. Richter, and M. Bossert, "Linearized shift-register synthesis," *IEEE Trans. Inf. Theory*, vol. 57, no. 9, pp. 6025–6032, Sep. 2011.

[43] E. R. Berlekamp, *Algebraic Coding Theory*. New York, NY, USA: McGraw-Hill, 1968.

[44] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.

[45] P. Loidreau, "A Welch–Berlekamp like algorithm for decoding Gabidulin codes," in *Proc. Int. Workshop Coding Cryptogr.*, O. Ytrehus, Ed. Berlin, Germany: Springer, 2006, pp. 36–45.

[46] E. R. Berlekamp and L. Welch, "Error correction of algebraic block codes," U.S. Patent 4 633 470 A, Dec. 12, 1986.

[47] P. Loidreau and R. Overbeck, "Decoding rank errors beyond the error-correcting capability," in *Proc. 10th Int. Workshop Algebr. Combinat. Coding Theory (ACCT)*, 2006, pp. 186–190.

[48] V. Sidorenko, L. Jiang, and M. Bossert, "Skew-feedback shift-register synthesis and decoding interleaved Gabidulin codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 621–632, Feb. 2011.

[49] A. Wachter-Zeh and A. Zeh, "List and unique error-erasure decoding of interleaved Gabidulin codes with interpolation techniques," *Des., Codes Cryptogr.*, vol. 73, no. 2, pp. 547–570, 2014.

[50] S. Puchinger, J. R. Nielsen, W. Li, and V. Sidorenko, "Row reduction applied to decoding of rank-metric and subspace codes," *Des., Codes Cryptogr.*, vol. 82, nos. 1–2, pp. 389–409, 2017.

[51] J.-H. Yu and H.-A. Loeliger, "Reverse Berlekamp–Massey decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Istanbul, Turkey, Jul. 2013, pp. 1212–1216.

[52] J.-H. Yu and H.-A. Loeliger, "Partial inverses mod $m(x)$ and reverse Berlekamp–Massey decoding," *IEEE Trans. Inf. Theory*, vol. 62, no. 12, pp. 6737–6756, Dec. 2016.

[53] J.-H. Yu and H.-A. Loeliger, "An algorithm for simultaneous partial inverses," in *Proc. 52nd Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Sep. 2014, pp. 928–935.

[54] J.-H. Yu and H.-A. Loeliger, "Decoding of interleaved Reed–Solomon codes via simultaneous partial inverses," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 2396–2400.

[55] J.-H. Yu and H.-A. Loeliger, "Simultaneous partial inverses and decoding interleaved Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 12, pp. 7511–7528, Dec. 2018.

[56] J. L. Dornstetter, "On the equivalence between Berlekamp's and Euclid's algorithms (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-33, no. 3, pp. 428–431, May 1987.

[57] A. E. Heydtmann and J. M. Jensen, "On the equivalence of the Berlekamp–Massey and the Euclidean algorithms for decoding," *IEEE Trans. Inf. Theory*, vol. 46, no. 7, pp. 2614–2624, Nov. 2000.

[58] M. Bras-Amorós and M. E. O'Sullivan, "From the Euclidean algorithm for solving a key equation for dual Reed–Solomon codes to the Berlekamp–Massey algorithm," in *Proc. AAECC* (Lecture Notes in Computer Science), vol. 5527, M. Bras-Amorós and T. Høholdt, Eds. New York, NY, USA: Springer, Jun. 2009, pp. 32–42.

[59] T. D. Mateer, "On the equivalence of the Berlekamp–Massey and the Euclidean algorithms for algebraic decoding," in *Proc. 12th Can. Workshop Inf. Theory*, Kelowna, BC, Canada, May 2011, pp. 139–142.
[60] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, "Rank errors and rank erasures correction," in *Proc. 4th Int. Colloq. Coding Theory*, Dilijan, Armenia, Oct. 1991, pp. 11–19.
[61] G. Richter and S. Plass, "Fast decoding of rank-codes with rank errors and column erasures," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Chicago, IL, USA, 2004, p. 398.
[62] E. M. Gabidulin and N. I. Pilipchuk, "Error and erasure correcting algorithms for rank codes," *Des., Codes Cryptogr.*, vol. 49, no. 1, pp. 105–122, Dec. 2008.
[63] E. B. Gabidulin, M. Bossert, and N. I. Pilipchuk, "Correcting generalized matrix erasures with applications to random network coding," in *Proc. Int. ITG Conf. Source Channel Coding (SCC)*, Siegen, Germany, Jan. 2010, pp. 1–7.
[64] S. Puchinger and A. Wachter-Zeh, "Fast operations on linearized polynomials and their applications in coding theory," *J. Symb. Comput.*, vol. 89, pp. 194–215, Nov./Dec. 2018.
[65] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: Elsevier, 1977.
[66] J. Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed. Cambridge, U.K.: Cambridge Univ. Press, 2013.
[67] J.-H. Yu and H.-A. Loeliger, "On irreducible polynomial remainder codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Saint Petersburg, Russia, Jul. 2011, pp. 1190–1194.
[68] J.-H. Yu and H.-A. Loeliger, "On polynomial remainder codes," 2012, *arXiv:1201.1812*.

Jiun-Hung Yu (Member, IEEE) received the M.S. degree in communication engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2003, and the Ph.D. degree in electrical engineering from ETH Zürich, in 2014. From 2003 to 2008, he was with Realtek Semiconductor Cooperation, Hsinchu. From 2008 to 2017, he was with the Signal and Information Processing Laboratory, ETH Zürich. Since 2017, he has been with National Yang Ming Chiao Tung University. He is currently an Assistant Professor. His research interests include communication theory, error-correcting codes, and statistical signal processing.

Hans-Andrea Loeliger (Fellow, IEEE) received the Diploma degree in electrical engineering from ETH Zürich, Switzerland, and the Ph.D. degree from ETH Zürich in 1992. From 1992 to 1995, he was with Linköping University, Linköping, Sweden. From 1995 to 2000, he was a Technical Consultant and the Co-Owner of consulting company. Since 2000, he has been a Professor with the Department of Information Technology and Electrical Engineering, ETH Zürich. His research interests have been in the broad areas of signal processing, machine learning, information theory, communications, error correcting codes, electronic circuits, quantum systems, and neural computation.