An Interpolation Procedure for List Decoding Reed–Solomon codes Based on Generalized Key Equations

Alexander Zeh, Christian Gentner, Daniel Augot

An Interpolation Procedure for List Decoding Reed–Solomon Codes Based on Generalized Key Equations

Alexander Zeh, Christian Gentner and Daniel Augot

***Abstract*—The key step of syndrome-based decoding of Reed–Solomon codes up to half the minimum distance is to solve the so-called Key Equation. List decoding algorithms, capable of decoding beyond half the minimum distance, are based on interpolation and factorization of multivariate polynomials. This article provides a link between syndrome-based decoding approaches based on Key Equations and the interpolation-based list decoding algorithms of Guruswami and Sudan for Reed–Solomon codes. The original interpolation conditions of Guruswami and Sudan for Reed–Solomon codes are reformulated in terms of a set of Key Equations. These equations provide a structured homogeneous linear system of equations of Block-Hankel form, that can be solved by an adaption of the Fundamental Iterative Algorithm. For an (n, k) Reed–Solomon code, a multiplicity s and a list size ℓ, our algorithm has time complexity $\mathcal{O}\left(\ell s^4 n^2\right)$.**

***Index Terms*—Guruswami–Sudan interpolation, Key Equation, list decoding, Fundamental Iterative Algorithm (FIA), Reed–Solomon codes, Hankel matrix, Block-Hankel matrix.**

I. Introduction

In 1999, Guruswami and Sudan [3]–[5] extended Sudan's original approach [6] by introducing multiplicities in the interpolation step of their polynomial-time list decoding procedure for Reed–Solomon and Algebraic Geometric codes. This modification permits decoding of (n, k) Reed–Solomon codes [7] (and Algebraic Geometric codes) of arbitrary code-rate $R = k/n$ with increased decoding radius. Guruswami and Sudan were focused on the existence of a polynomial-time algorithm. Kötter [8] and Roth–Ruckenstein [9], [10] proposed quadratic time algorithms for the key steps of the Guruswami–Sudan principle for Reed–Solomon codes, i.e., interpolation and factorization of bivariate polynomials. Various other approaches for a low-complexity realization of Guruswami–Sudan exist, e.g. the work of Alekhnovich [11], where fast computer algebra techniques are used. Trifonov's [12] contributions rely on ideal theory and divide and conquer methods. Sakata uses Gröbner–bases techniques [13], [14].

In this paper, we reformulate the *bivariate* interpolation step of Guruswami–Sudan for Reed–Solomon codes in a set of *univariate* Key Equations [1]. This extends the previous work of Roth and Ruckenstein [9], [10], where the reformulation was done for the special case of Sudan. Furthermore, we present a modification of the so-called Fundamental Iterative Algorithm (FIA), proposed by Feng and Tzeng in 1991 [15]. Adjusted to the special case of one Hankel matrix the FIA resembles the approach of Berlekamp and Massey [16], [17].

Independently of our contribution, Beelen and Høholdt reformulated the Guruswami–Sudan constraints for Algebraic Geometric codes [18], [19]. It is not clear, if the system they obtain is highly structured.

This contribution is structured as follows. The next section contains basic definitions for Reed–Solomon codes and bivariate polynomials. In Section III, we derive the Key Equation for conventional decoding of Reed–Solomon codes from the Welch–Berlekamp approach [20] and we present the adjustment of the FIA for one Hankel matrix. A modified version of Sudan's reformulated interpolation problem based on the work of Roth–Ruckenstein [9] is derived and the adjustment of the FIA for this case is illustrated in Section IV. In Section V, the interpolation step of the Guruswami–Sudan principle is reformulated. The obtained homogeneous set of linear equations has Block-Hankel structure. We adjust the FIA for this Block-Hankel structure, prove the correctness of the proposed algorithm and analyze its complexity. We conclude this contribution in Section VI.

II. Definitions and Preliminaries

Throughout this paper, $[n]$ denotes the set of integers $\{1, 2, \ldots, n\}$ and $[n]_0$ denotes the set of integers $\{0, 1, \ldots, n-1\}$. An $m \times n$ matrix $\mathbf{A} = ||A_{i,j}||$ consists of the entries $A_{i,j}$, where $i \in [m]_0$ and $j \in [n]_0$. A univariate polynomial $A(x)$ of degree less than n is denoted by $A(x) = \sum_{i=0}^{n-1} A_i x^i$. A vector of length n is represented by $\mathbf{r} = (r_1, r_2, \ldots, r_n)^T$.

Let q be a power of a prime and let $\mathbb{F} = \mathrm{GF(q)}$ denote the finite field of order q. Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ denote nonzero distinct elements (code–locators) of $\mathbb{F}$ and let $\upsilon'_1, \upsilon'_2, \ldots, \upsilon'_n$ denote nonzero elements (column–multipliers), the associated evaluation map ev is

$$\begin{array}{rccc} \mathrm{ev}: & \mathbb{F}[x] & \to & \mathbb{F}^n \\ & f(x) & \mapsto & \left(\upsilon'_1 f(\alpha_1), \upsilon'_2 f(\alpha_2), \ldots, \upsilon'_n f(\alpha_n)\right). \end{array} \tag{1}$$

The associated Generalized Reed–Solomon code $\mathcal{GRS}(n, k)$ of length n and dimension k is [21]:

$$\mathcal{GRS}(n, k) = \{\mathbf{c} = \mathrm{ev}(f(x)) : f(x) \in \mathbb{F}_k[x]\}, \tag{2}$$

where $\mathbb{F}_k[x]$ denotes the set of all univariate polynomials with degree less than k. Generalized Reed–Solomon codes are MDS codes with minimum distance $d = n - k + 1$. The dual of

Parts of this work were published in the proceedings of 2008 IEEE International Symposium on Information Theory (ISIT 2008), Toronto, Canada [1] and 2009 IEEE Information Theory Workshop (ITW 2009), Taormina, Sicily, Italy [2]. This work was supported by the German Research Council "Deutsche Forschungsgemeinschaft" (DFG) under Grant No. Bo867/22-1.

a Generalized Reed–Solomon is also a Generalized Reed–Solomon code with the same code locators and column multipliers $\upsilon_1, \upsilon_2, \ldots, \upsilon_n$, where $\sum_{i=1}^{n} \upsilon_i' \upsilon_i \alpha_i^j = 0,\ j \in [n-1]_0$. The explicit form of the column multipliers is [22]:

$$\upsilon_i = \frac{1}{\upsilon_i'} \cdot \frac{1}{\prod_{j \neq i}(\alpha_i - \alpha_j)}, \quad i \in [n]. \tag{3}$$

We will take advantage of structured matrices and therefore we recall the definition of a Hankel matrix in the following.

Definition 1 (Hankel Matrix): An $m \times n$ Hankel matrix $\mathbf{S} = ||S_{i,j}||$ is a matrix, where $S_{i,j} = S_{i-1,j+1}$ for all $i \in [m-1]$ and $j \in [n-1]_0$ holds.

Let us recall some properties of bivariate polynomials in $\mathbb{F}[x, y]$.

Definition 2 (Weighted Degree): Let the polynomial $A(x, y) = \sum_{i,j} A_i^{(j)} x^i y^j$ be in $\mathbb{F}[x, y]$. Then, the (w_x, w_y)-weighted degree of $A(x, y)$, denoted by $\mathrm{wdeg}_{\mathrm{w_x,w_y}}$, is the maximum over all $iw_x + jw_y$ such that $A_i^{(j)} \neq 0$.

Definition 3 (Multiplicity and Hasse Derivative [23]): Let $A(x, y) = \sum_{i,j} A_i^{(j)} x^i y^j$ be a polynomial in $\mathbb{F}[x, y]$. Let $\overline{A}(x, y) = A(x + \alpha, y + \beta) = \sum_{i,j} \overline{A}_i^{(j)} x^i y^j$. A bivariate polynomial $A(x, y)$ has at least multiplicity s in the point (α, β), denoted by

$$\mathrm{mult}\left(A(x, y), (\alpha, \beta)\right) \geq s, \tag{4}$$

if the coefficients $\overline{A}_i^{(j)}$ are zero for all $i + j < s$. Furthermore, the (a, b)th Hasse derivative of the polynomial $A(x, y)$ in the point (α, β) is

$$A^{[a,b]}(\alpha, \beta) = \sum_{i \geq a, j \geq b} \binom{i}{a} \binom{j}{b} A_i^{(j)} \alpha^{i-a} \beta^{j-b} \tag{5}$$

Let $A^{[b]}(x, y) = A^{[0,b]}(x, y)$ denote the bth Hasse derivative of $A(x, y)$ with respect to the variable y.

We will use the inner product for bivariate polynomials to describe our algorithms.

Definition 4 (Inner Product): Let two polynomials $A(x, y) = \sum_{i,j} A_i^{(j)} x^i y^j$ and $B(x, y) = \sum_{i,j} B_i^{(j)} x^i y^j$ in $\mathbb{F}[x, y]$ be given. The inner product $\langle A(x, y), B(x, y) \rangle$ of $A(x, y)$ and $B(x, y)$ is defined by $\sum_{i,j} A_i^{(j)} B_i^{(j)}$.

III. Welch–Berlekamp as List-One Decoder and the Fundamental Iterative Algorithm

A. Syndrome-Based Decoding of Reed–Solomon Codes

Let $\mathbf{e} = (e_1, e_2, \ldots, e_n)$ denote the error word and let $\mathcal{J}$ be the set of error locations (that is $e_j \neq 0 \Leftrightarrow j \in \mathcal{J}$). Let $\tau = \lfloor (n-k)/2 \rfloor$. It is well-known that a $\mathcal{GRS}(n, k)$ code can recover uniquely any error pattern if and only if $|\mathcal{J}| \leq \tau$. The $n - k$ syndrome coefficients $S_0, S_1, \ldots, S_{n-k-1}$ depend only on the error word $\mathbf{e}$ and the associated syndrome polynomial $S(x)$ is defined by [22]:

$$S(x) = \sum_{i=0}^{n-k-1} S_i x^i \equiv \sum_{j=1}^{n} \frac{e_j \upsilon_j}{1 - \alpha_j x} \mod x^{n-k}.$$

The error-locator polynomial is $\Lambda(x) = \prod_{j \in \mathcal{J}}(1 - \alpha_j x)$ and the error-evaluator polynomial $\Omega(x)$ is $\sum_{j \in \mathcal{J}} e_j \upsilon_j \prod_{i \in \mathcal{J} \setminus \{j\}}(1 - \alpha_i x)$. They are related by the Key Equation:

$$\frac{\Omega(x)}{\Lambda(x)} \equiv S(x) \bmod x^{n-k}. \tag{6}$$

The main steps for conventional decoding up to half the minimum distance are:

1) Calculate the syndrome polynomial $S(x)$ from the received word $\mathbf{r} = \mathbf{c} + \mathbf{e}$.
2) Solve (6) for the error-locator polynomial $\Lambda(x)$ and determine its roots.
3) Compute $\Omega(x)$ and then determine the error values.

B. Derivation of the Key Equation from Welch–Berlekamp

We derive the classical Key Equation (6) from the simplest interpolation based decoding algorithm, reported as the "Welch–Berlekamp" decoding algorithm in [24]–[26]. We provide a simpler representation than in [20] and give a polynomial derivation of the Key Equation.

Consider a $\mathcal{GRS}(n, k)$ code with support set $\alpha_1, \alpha_2, \ldots, \alpha_n$, multipliers $\upsilon_1', \upsilon_2', \ldots, \upsilon_n'$ and dimension k. The Welch–Berlekamp approach is based on the following lemma [27, Ch. 5.2]:

Lemma 1 (List-One Decoder): Let $\mathbf{c} = \mathrm{ev}(f(x))$ be a codeword of a $\mathcal{GRS}(n, k)$ code and let $\mathbf{r} = \mathbf{c} + \mathbf{e} = (r_1, r_2, \ldots, r_n)$ be the received word. We search for a polynomial $Q(x, y) = Q^{(0)}(x) + Q^{(1)}(x)y$ in $\mathbb{F}[x, y]$ such that:

1) $Q(x, y) \neq 0$,
2) $Q(\alpha_i, r_i/\upsilon_i') = 0 \quad \forall i \in [n]$,
3) $\mathrm{wdeg}_{1,\text{k-1}}\, Q(x, y) < n - \tau$.

If $\mathbf{c}$ has distance less than or equal to $\lfloor (n-k)/2 \rfloor$ from the received word $\mathbf{r}$, then $f(x) = -Q^{(0)}(x)/Q^{(1)}(x)$.

Let us connect Lemma 1 to (6).

Proposition 1 (Univariate Reformulation): Let $R(x)$ be the Lagrange interpolation polynomial, such that $R(\alpha_i) = r_i/\upsilon_i',\ i \in [n]$ holds. Let $G(x) = \prod_{i=1}^{n}(x - \alpha_i)$. Then $Q(x, y) = Q^{(0)}(x) + Q^{(1)}(x)y$ satisfies Conditions *2)* and *3)* of Lemma 1 if and only if there exists a polynomial $B(x) \in \mathbb{F}[x]$ such that

$$Q(x, R(x)) = B(x) \cdot G(x), \tag{7}$$

and $\deg B(x) < n - k - \tau$.

Let $N_t = n - \tau - t(k-1),\ t = 0, 1$. Define the following reciprocal polynomials:

$$\begin{aligned} \overline{R}(x) &= x^{n-1} \cdot R(x^{-1}), \\ \overline{G}(x) &= x^{n} \cdot G(x^{-1}) = \prod_{i=1}^{n}(1 - \alpha_i x), \\ \Omega(x) &= x^{\deg B(x)} \cdot B(x^{-1}), \\ \Lambda^{(t)}(x) &= x^{N_t - 1} \cdot Q^{(t)}(x^{-1}). \end{aligned} \tag{8}$$

Inverting the order of the coefficients of (7) leads to:

$$x^{n-\tau+n-k-1}\big(Q^{(0)}(x^{-1}) + Q^{(1)}(x^{-1}) \cdot R(x^{-1})\big) = \Omega(x) \cdot \overline{G}(x).$$

With (8), we obtain:

$$x^{n-k}\Lambda^{(0)}(x) + \Lambda^{(1)}(x) \cdot \overline{R}(x) = \Omega(x) \cdot \overline{G}(x),$$

which we can consider modulo x^{n-k}. We obtain

$$\Lambda^{(1)}(x) \cdot \overline{R}(x) \equiv \Omega(x) \cdot \overline{G}(x) \bmod x^{n-k}. \tag{9}$$

Since $\overline{G}(0) \neq 0$, we can define the formal power series $\overline{R}(x)/\overline{G}(x)$:

$$\frac{\overline{R}(x)}{\overline{G}(x)} = \sum_{i=0}^{\infty} S_i x^i = S(x). \tag{10}$$

Using the column multipliers (3) for the dual code, it can be verified that $S(x)$ is the series of syndromes with

$$S_i = \sum_{j=1}^{n} v_j r_j \alpha_j^i. \tag{11}$$

Thus, dividing (9) by $\overline{G}(x)$, we obtain

$$S(x) \cdot \Lambda^{(1)}(x) \equiv \Omega(x) \mod x^{n-k}, \tag{12}$$

which corresponds to the classical Key Equation (6). The syndrome polynomial is $S(x) \bmod x^{n-k}$, and $\Lambda^{(1)}(x)$ is the error-locator polynomial $\Lambda(x)$.

In the case of τ errors, we consider only the terms of the Key Equation of degree greater than $n-k-\tau$ and we get the following homogeneous linear system of equations:

$$\begin{pmatrix} S_0 & S_1 & \dots & S_\tau \\ S_1 & S_2 & \dots & S_{\tau+1} \\ \vdots & \vdots & \iddots & \vdots \\ S_{\tau-1} & S_\tau & \dots & S_{2\tau-1} \end{pmatrix} \begin{pmatrix} \Lambda_\tau^{(1)} \\ \Lambda_{\tau-1}^{(1)} \\ \vdots \\ \Lambda_0^{(1)} \end{pmatrix} = \mathbf{0}. \tag{13}$$

The above syndrome matrix $\mathbf{S} = ||S_{i,j}||$ for all $i \in [\tau]_0$ and $j \in [\tau+1]_0$ has Hankel form (see Definition 1). Equation (12) can be solved by the well-known Berlekamp–Massey algorithm [16], [17] or with a modification of the Extended Euclidean algorithm [28]. The parallels of the Berlekamp–Massey algorithm and the Extended Euclidean algorithm have been considered in [29]–[31].

We consider in the following the FIA [15], that can be used to find the first $\mu+1$ linearly dependent columns and connection coefficients $T_1, T_2, \dots, T_\mu$ for an arbitrary matrix. The FIA allows a significant reduction of complexity when adjusted to a Hankel matrix as in (13).

C. The FIA for One Hankel Matrix

Given an arbitrary $m \times n$ matrix $\mathbf{A} = ||A_{i,j}||$, the FIA outputs the minimal number of $\mu+1$ linearly dependent columns together with the polynomial $T(x) = \sum_{j=0}^{\mu} T_j x^j$, with $T_\mu \neq 0$, such that $\sum_{j=0}^{\mu} T_j A_{i,j} = 0, i \in [m]_0$ holds. The FIA scans the μth column of the matrix $\mathbf{A}$ row-wise in the order $A_{0,\mu}, A_{1,\mu}, \dots$ and uses previously stored polynomials to update the current polynomial $T(x)$. Let μ be the index of the current column under inspection, and let $T(x) = \sum_{j=0}^{\mu} T_j x^j$ be the current candidate polynomial that satisfies:

$$\sum_{j=0}^{\mu} T_j A_{i,j} = 0, \quad i \in [\kappa]_0,$$

for some value of the row index κ. In other words, the coefficients of the polynomial $T(x)$ give us the vanishing linear combination of the matrix consisting of the first κ rows and the first $\mu+1$ columns of the matrix $\mathbf{A}$. Suppose that the discrepancy:

$$\Delta = \sum_{j=0}^{\mu} T_j A_{\kappa,j} \neq 0. \tag{14}$$

for next row κ is nonzero. If there exists a previously stored polynomial $T^{(\kappa)}(x)$ and a nonzero discrepancy $\Delta^{(\kappa)}$, corresponding to row κ, then the current polynomial $T(x)$ is updated in the following way:

$$T(x) \leftarrow T(x) - \frac{\Delta}{\Delta^{(\kappa)}} T^{(\kappa)}(x). \tag{15}$$

The proof of the above update rule is straightforward [15].

In the case $\Delta \neq 0$ and there is no discrepancy $\Delta^{(\kappa)}$ stored, the actual discrepancy Δ is stored as $\Delta^{(\kappa)}$. The corresponding auxiliary polynomial is stored as $T^{(\kappa)}(x)$. Then, the FIA examines a new column $\mu+1$.

Definition 5 (True Discrepancy): Let the FIA examine the κth row of the μth column of matrix $\mathbf{A}$. Furthermore, let the calculated discrepancy (14) be nonzero and no other nonzero discrepancy be stored for row κ. Then, the FIA examines a new column $\mu+1$. We call this case a **true** discrepancy.

Theorem 1 (Correctness and Complexity of the FIA [15]): For an $m \times n$ matrix with $n > m$, the Fundamental Iterative Algorithm stops, when the row pointer has reached the last row of column μ. Then, the last polynomial $T^{(\mu)}(x)$ corresponds to a valid combination of the first $\mu+1$ columns. The complexity of the algorithm is $O(m^3)$.

For a Hankel matrix $\mathbf{S}$ (as in Definition 1), the FIA can be adjusted. Assume the case of a true discrepancy, when the FIA examines the κth row of the μth column of the structured matrix $\mathbf{S}$. The current polynomial is $T(x)$. Then, the FIA starts examining the $(\mu+1)$th column at row $\kappa - 1$ with $T(x) \leftarrow x \cdot T(x)$ and not at row zero. This reduces the cubic time complexity into a quadratic time complexity [15].

To illustrate the complexity reduction of the FIA when adjusted to a Hankel matrix (compared to the original, unadjusted FIA), we traced the examined rows for each column in Figure 1. Figure 1(a) shows the values of κ of the FIA without any adaption. The row pointer κ of the adapted FIA is traced in Figure 1(b).

The points on the lines in both figures indicate the case, where a true discrepancy has been encountered.

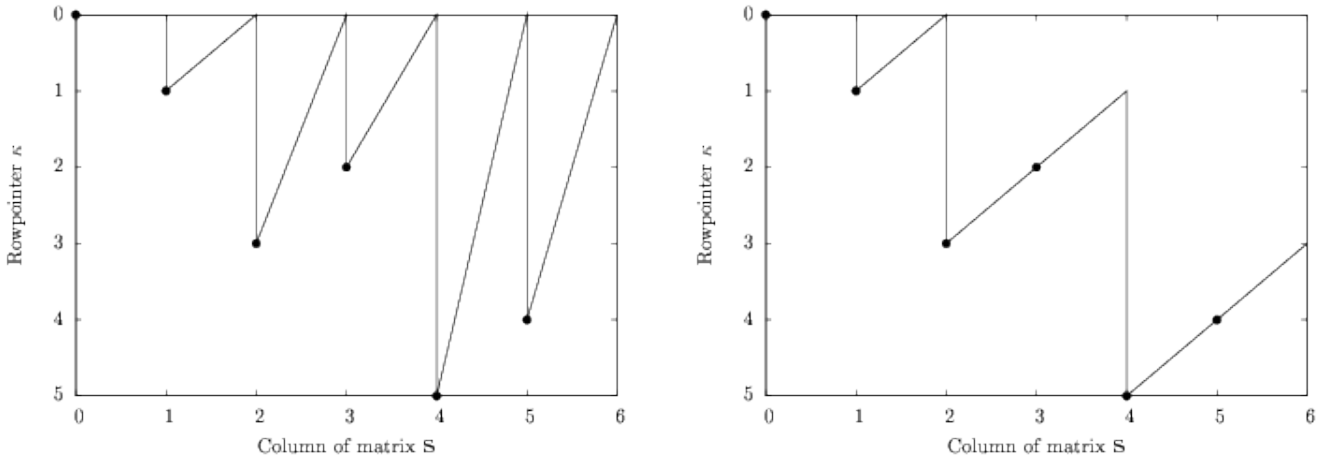
IV. Sudan Interpolation Step with a Horizontal Band of Hankel Matrices

A. Univariate Reformulation of the Sudan Interpolation Step

In this section, we recall parts of the work of Roth and Ruckenstein [9], [10] for the interpolation step of the Sudan [6] principle. The aimed decoding radius is denoted by τ, the corresponding list size is ℓ.

Problem 1 (Sudan Interpolation Step [6]): Let the aimed decoding radius τ and the received word $(r_1, r_2, \dots, r_n)$ be given. The Sudan interpolation step determines a polynomial $Q(x,y) = \sum_{t=0}^{\ell} Q^{(t)}(x) y^t \in \mathbb{F}[x,y]$, such that

1) $Q(x,y) \neq 0$,



(a) Classic FIA (unadjusted)

(b) Adjusted FIA

Fig. 1. Illustration of the row pointer κ of the classic FIA (Sub-figure 1(a)) and of the adjusted FIA (Sub-figure 1(b)) when both algorithms are applied to the same 6×7 Hankel syndrome matrix of a $\mathcal{GRS}(16, 4)$ code. The dots indicate a true discrepancy. In this case, both algorithms enter a new column, but with different initial values of their row pointers.

2) $Q(\alpha_i, r_i/\upsilon_i') = 0, \quad \forall i \in [n]$,
3) $\mathrm{wdeg}_{1,\text{k-1}}\, Q(x, y) < n - \tau$.

We present here a slightly modified version of [9], to get an appropriate basis for the extension to the interpolation step in the Guruswami–Sudan case.

We have $\deg Q^{(t)}(x) < N_t \stackrel{\text{def}}{=} n - \tau - t(k-1), \ t \in [\ell+1]_0$. Let $R(x)$ be the Lagrange interpolation polynomial, s.t. $R(\alpha_i) = r_i/\upsilon_i'$, $i \in [n]$ and $G(x) = \prod_{i=1}^{n}(x - \alpha_i)$. The reciprocal polynomial of $Q^{(t)}(x)$ is denoted by $\Lambda^{(t)}(x) = x^{N_t-1}Q^{(t)}(x^{-1})$.

Similar to Proposition 1, Roth–Ruckenstein [9] proved the following. There is an interpolation polynomial $Q(x, y)$ satisfying Conditions 2 and 3 if and only if there exists a univariate polynomial $B(x)$ with degree smaller than $\ell(n-k) - \tau$, s.t. $Q(x, R(x)) = B(x) \cdot G(x)$.

Let the reciprocal polynomials be defined as in (8). From [9, Equation (19)] we have:

$$\sum_{t=0}^{\ell} \Lambda^{(t)}(x) \cdot x^{(\ell-t)(n-k)} \cdot \overline{R}(x)^t \equiv \overline{B}(x) \cdot \overline{G}(x) \bmod x^{n-\tau+\ell(n-k)}, \tag{16}$$

where $\deg \overline{B}(x) < \ell(n-k)-\tau$. We introduce the power series

$$T^{(t)}(x) \stackrel{\text{def}}{=} \frac{\overline{R}(x)^t}{\overline{G}(x)} = \sum_{i=0}^{\infty} T_i^{(t)} x^i. \tag{17}$$

Inserting (17) into (16) leads to:

$$\sum_{t=0}^{\ell} \Lambda^{(t)}(x) \cdot x^{(\ell-t)(n-k)} \cdot T^{(t)}(x) \equiv \overline{B}(x) \bmod x^{n-\tau+\ell(n-k)}. \tag{18}$$

Based on (18) we can now define syndromes for Problem 1.

Definition 6 (Syndromes for Sudan): The $\ell+1$ generalized syndrome polynomials $S^{(t)}(x) \stackrel{\text{def}}{=} \sum_{i=0}^{n+N_t-1} S_i^{(t)} x^i$ are given by:

$$S_i^{(t)} = T_{i+(t-1)(n-1)}^{(t)}, \quad i \in [n + N_t]_0, \quad t \in [\ell+1]_0. \tag{19}$$

The first order **Extended Key Equation** is:

$$\sum_{t=0}^{\ell} \Lambda^{(t)}(x) \cdot S^{(t)}(x) \equiv \overline{B}(x) \bmod x^{n-\tau+\ell(n-k)}, \tag{20}$$

with $\deg \overline{B}(x) < \ell(n-k) - \tau$.
An explicit form of $S_i^{(t)}$ is:

$$S_i^{(t)} = \sum_{j=1}^{n} \upsilon_j \frac{r_j^t}{\upsilon_i'^{t-1}} \alpha_j^i, \quad i \in [n + N_t]_0, \quad t \in [\ell+1]_0. \tag{21}$$

Note 1: In [9], a further degree reduction is proposed. Then (18), is modulo x^{n-k} and the polynomial $\Lambda^{(0)}(x)$ disappears. We do not present this improvement here, because we cannot properly reproduce this behavior in the Guruswami–Sudan case (see Note 2).

The degree of the LHS of (16) is smaller than $n-\tau+\ell(n-k)$. If we consider the terms of degree higher than $\ell(n-k)-\tau$, we obtain n homogeneous linear equations. Reverting back to the originals univariate polynomials $Q^{(t)}(x)$, we get the following system:

$$\sum_{t=0}^{\ell} \sum_{i=0}^{N_t-1} Q_i^{(t)} \cdot S_{i+j}^{(t)} = 0, \quad j \in [n]_0. \tag{22}$$

With $\mathbf{Q}^{(t)} = (Q_0^{(t)}, Q_1^{(t)}, \ldots, Q_{N_t-1}^{(t)})^T$, we obtain the following matrix form:

$$\left(\mathbf{S}^{(0)}\, \mathbf{S}^{(1)}\, \cdots\, \mathbf{S}^{(\ell)}\right) \cdot \begin{pmatrix} \mathbf{Q}^{(0)} \\ \mathbf{Q}^{(1)} \\ \vdots \\ \mathbf{Q}^{(\ell)} \end{pmatrix} = \mathbf{0}, \tag{23}$$

where each sub-matrix $\mathbf{S}^{(t)} = ||S_{i,j}^{(t)}||$, $i \in [n]_0, j \in [N_t]_0, t \in [\ell+1]_0$ is a Hankel matrix. The $\ell+1$ syndrome polynomials $S^{(t)}(x) = \sum_{i=0}^{N_t-1} S_i^{(t)} x^i$ of Definition 6 are associated with this horizontal band of $\ell+1$ Hankel matrices by $S_{i,j}^{(t)} = S_{i+j}^{(t)}$.

In the following, we describe how the FIA can be adapted to solve the homogeneous system of equations (23).

B. Adjustment of the FIA for the Reformulated Sudan Interpolation Problem

The FIA can directly be applied to the matrix $||\mathbf{S}^{(0)}\,\mathbf{S}^{(1)}\,\cdots\,\mathbf{S}^{(\ell)}||$ of (23), but if we want to take advantage of the Hankel structure we have to scan the columns of $||\mathbf{S}^{(0)}\,\mathbf{S}^{(1)}\,\cdots\,\mathbf{S}^{(\ell)}||$ in a manner given by the weighted degree requirement of the interpolation problem.

Let $\prec_H$ denote the ordering for the pairs $\{(\nu,\mu)|\nu \in [\ell+1]_0 \text{ and } \mu \in \mathbb{N}\}$, where $(\nu,\mu) \prec_H (\overline{\nu},\overline{\mu})$ is given by:

$$(\nu,\mu) \prec_H (\overline{\nu},\overline{\mu}) \iff \begin{cases} \nu+\mu(k-1) < \overline{\nu}+\overline{\mu}(k-1) \\ \text{or} \\ \nu+\mu(k-1) = \overline{\nu}+\overline{\mu}(k-1) \text{ and } \mu < \overline{\mu}. \end{cases} \tag{24}$$

The pair that immediately follows (ν,μ) with respect to the order defined by $\prec_H$ is denoted by $\mathrm{succ}(\prec_H,(\nu,\mu))$. The columns of the matrix $\mathbf{S} = ||\mathbf{S}^{(0)}\,\mathbf{S}^{(1)}\,\cdots\,\mathbf{S}^{(\ell)}||$ are reordered according to $\prec_H$. The pair (ν,μ) indexes the μth column of νth sub-matrix $\mathbf{S}^{(\nu)}$. More explicitly, we obtain the following matrix $\mathbf{S}'$, where the columns of $\mathbf{S}$ are reordered (see Equation (25)).

The corresponding homogeneous system of equations can now be written in terms of the inner product for bivariate polynomials (see Definition 4).

Problem 2 (Reformulated Sudan Interpolation Problem): Let the $\ell+1$ syndrome polynomials $S^{(0)}(x)$,$S^{(1)}(x)$,..., $S^{(\ell)}(x)$ be given by Definition 6 and let $S(x,y) \stackrel{\text{def}}{=} \sum_{t=0}^{\ell} S^{(t)}(x)y^t$ be the corresponding bivariate syndrome polynomial. We search a nonzero bivariate polynomial $T(x,y)$ such that:

$$\langle x^{\kappa}T(x,y), S(x,y)\rangle = 0, \quad \kappa \in [n]_0. \tag{26}$$

Hence, the bivariate polynomial $T(x,y)$ is a valid interpolation polynomial for Problem 1. Note that each polynomial $S^{(t)}(x)$, as defined in (16), has degree smaller than N_t+n-1. To index the columns of the rearranged matrix $\mathbf{S}'$, let

$$C_{\nu,\mu} = \big|\{(t,i)\,|\,(t,i) \prec_H (\nu,\mu)\}\big|. \tag{27}$$

Algorithm 1 is the modified FIA for solving Problem 2. In contrast to the original Roth–Ruckenstein adaption we consider all n homogeneous linear equations (instead of τ), according to Note 1. The column pointer is given by (ν,μ), for indexing the μth column of the νth sub-matrix $\mathbf{S}^{(\nu)}$. Algorithm 1 virtually scans the rearranged matrix $\mathbf{S}'$ column after column (see Line 23 of Algorithm 1). The true discrepancy value for row κ is stored in array D as $D[\kappa]$, and the corresponding intermediate bivariate polynomial is stored in array A as $A[\kappa]$. The discrepancy calculation and the update rule (see (14) and (15) for the basic FIA) is adapted to the bivariate case (see Line 16 of Algorithm 1). For each sub-matrix $\mathbf{S}^{(\nu)}$, the previous value of the row pointer κ is stored in an array R as $R[\nu]$. We prove the initialization rule for the FIA solving Problem 2 in the following proposition.

Algorithm 1: Algorithm Solving Problem 2

Input: Bivariate polynomial $S(x,y) = \sum_{t=0}^{\ell} S^{(t)}(x)y^t$;
Output: Bivariate polynomial $T(x,y)$;

Data structures:
Bivariate polynomial $T(x,y) = \sum_{t=0}^{\ell} T^{(t)}(x)y^t$;
Column pointers (ν,μ), where $\nu \in [\ell]_0$, $\mu \in [N_\nu]_0$;
Row pointer $\kappa \in [n]_0$;
Array D of n entries in $\mathbb{F}$;
Array R of $\ell+1$ entries in $\mathbb{N}$;
Array A of n entries in $\mathbb{F}[x,y]$;
Variable $\Delta \in \mathbb{F}$, variable *compute* $\in \{\text{TRUE}, \text{FALSE}\}$;

Initialize:
for $i \in [n]_0$ **do**
  $D[i] \leftarrow 0$;
for $i \in [\ell+1]_0$ **do**
  $R[i] \leftarrow 0$;
$(\nu,\mu) \leftarrow (0,0)$, $\kappa \leftarrow 0$, *compute* $\leftarrow$ FALSE;
1 **while** $\kappa < n$ **do**
2 **if** *compute* **then**
3 $\Delta \leftarrow \langle x^{\kappa} \cdot T(x,y), S(x,y)\rangle$;
4 **else**
5 **if** $R[\nu] < 1$ **then**
6 $T(x,y) \leftarrow y^{\nu} \cdot x^{\mu}$;
7 $\Delta \leftarrow S_{\mu}^{(\nu)}$;
8 $\kappa \leftarrow 0$;
9 **else**
10 $T(x,y) \leftarrow x \cdot A[R[\nu]](x,y)$;
11 $\Delta \leftarrow D[R[\nu]]$;
12 $\kappa \leftarrow R[\nu] - 1$;
13 *compute* $\leftarrow$ TRUE;
14 **if** $\Delta = 0$ *or* $D[\kappa] \neq 0$ **then**
15 **if** $\Delta \neq 0$ **then**
16 $T(x,y) \leftarrow T(x,y) - \frac{\Delta}{D[\kappa]} \cdot A[\kappa](x,y)$;
17 $\kappa \leftarrow \kappa + 1$;
18 **else** /* $\Delta \neq 0$ *and* $D[\kappa] = 0$ */
19 $A[\kappa] \leftarrow T(x,y)$;
20 $D[\kappa] \leftarrow \Delta$;
21 $R[\nu] \leftarrow \kappa$;
22 *compute* $\leftarrow$ FALSE;
23 $(\nu,\mu) \leftarrow \mathrm{succ}(\prec_H,(\nu,\mu))$;

Proposition 2 (Initialization Rule): Assume Algorithm 1 examines column (ν,μ) of a syndrome matrix $\mathbf{S} =$

$$\mathbf{S}' = \left(\begin{array}{cccc|cccc|c|c|cc} S_0^{(0)} & S_1^{(0)} & \cdots & S_{k-2}^{(0)} & S_{k-1}^{(0)} & S_0^{(1)} & \cdots & S_{2k-3}^{(0)} & S_{k-2}^{(1)} & \cdots & \cdots & S_{N_{\ell-1}-1}^{(\ell-1)} & S_{N_\ell-1}^{(\ell)} \\ S_1^{(0)} & S_2^{(0)} & \cdots & S_{k-1}^{(0)} & S_k^{(0)} & S_1^{(1)} & \cdots & S_{2(k-2)}^{(0)} & S_{k-1}^{(1)} & \cdots & \cdots & S_{N_{\ell-1}}^{(\ell-1)} & S_{N_\ell}^{(\ell)} \\ \vdots & \vdots & \iddots & \vdots & \vdots & \vdots & \iddots & \vdots & \vdots & \iddots & \cdots & \vdots & \vdots \\ S_{n-1}^{(0)} & S_n^{(0)} & \cdots & S_{n+k-3}^{(0)} & S_{n-k-2}^{(0)} & S_{n-1}^{(1)} & \cdots & S_{n+2(k-2)}^{(0)} & S_{n-k-3}^{(1)} & \cdots & \cdots & S_{n+N_{\ell-1}-2}^{(\ell-1)} & S_{n+N_\ell-2}^{(\ell)} \end{array}\right) \tag{25}$$

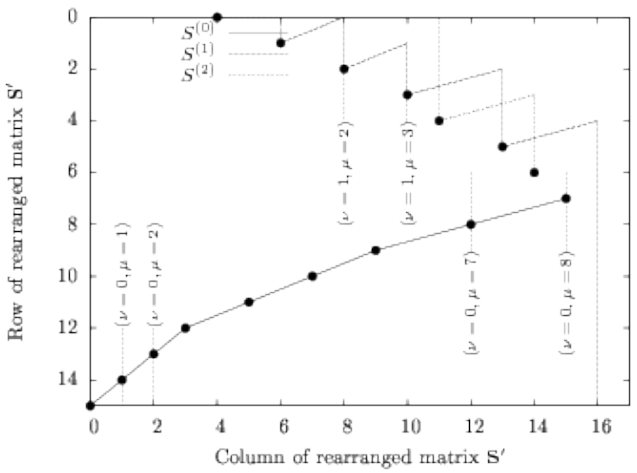


Fig. 2. Illustration of the row pointer κ of Algorithm 1 applied to a horizontal band of three Hankel matrices $\mathbf{S}^{(0)}$, $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$. The columns of the 16×18 matrix $\mathbf{S}'$ are arranged under $\prec_H$-ordering. The three lines $S^{(0)}$, $S^{(1)}$ and $S^{(2)}$ trace the row pointer for each sub-matrix $\mathbf{S}^{(0)}$, $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$.

TABLE I
COLUMN-INDEX $C_{\nu,\mu}$ AND COLUMN POINTER (ν,μ) OF THE RE-ARRANGED MATRIX $\mathbf{S}'$ OF THE REFORMULATED SUDAN INTERPOLATION STEP FOR A $\mathcal{GRS}(16,4)$ CODE WITH DECODING RADIUS $\tau=7$ AND LIST SIZE $\ell=2$.

Column $C_{\nu,\mu}$	**and Column pointers (ν,μ)**		
0	(0,0)	9	(0,6)
1	(0,1)	10	(1,3)
2	(0,2)	11	(2,0)
3	(0,3)	12	(0,7)
4	(1,0)	13	(1,4)
5	(0,4)	14	(2,1)
6	(1,1)	15	(0,8)
7	(0,5)	16	(1,5)
8	(1,2)	17	(2,2)

$||\mathbf{S}^{(0)}\,\mathbf{S}^{(1)}\,\cdots\,\mathbf{S}^{(\ell)}||$ as defined in (23) (or equivalently the bivariate polynomial $S(x,y)$). Assume that a true discrepancy is obtained in row κ_ν.

Let $(\overline{\nu},\overline{\mu}) = \mathrm{succ}(\prec_H,(\nu,\mu))$. Hence, Algorithm 1 can examine column $(\overline{\nu},\overline{\mu})$ at row $\kappa_{\overline{\nu}}-1$ with the initial value $\overline{T}(x,y) = x\cdot T(x,y)$, where $\kappa_{\overline{\nu}}$ is the index of the row, where the last true discrepancy in the $\overline{\nu}$th sub-matrix $\mathbf{S}^{(\overline{\nu})}$ was calculated. The polynomial $T(x,y)$ is the stored intermediate polynomial for $\mathbf{S}^{(\overline{\nu})}$, i.e., $\kappa_{\overline{\nu}} = R[\overline{\nu}]$ and $T(x,y) = A[\kappa_{\overline{\nu}}]$.

Proof: In terms of the inner product (see Definition 4), we have:

$$\langle x^i T(x,y), S(x,y)\rangle = 0, \quad i\in[\kappa_{\overline{\nu}}]_0.$$

Let us write $T(x,y) = \sum_{t=0}^{\overline{\nu}}\sum_{j=0}^{\overline{\mu}} T_j^{(t)} x^j y^t$. We have $\overline{T}(x,y) = \sum_{t=0}^{\overline{\nu}}\sum_{j=1}^{\overline{\mu}+1} T_{j-1}^{(t)} x^j y^t$ and we compute:

$$\begin{aligned}
\langle x^i\overline{T}(x,y), S(x,y)\rangle &= \sum_{t=0}^{\overline{\nu}}\sum_{j=1}^{\overline{\mu}+1} \overline{T}_j^{(t)}\cdot S_{i,j}^{(t)} \\
&= \sum_{t=0}^{\overline{\nu}}\sum_{j=1}^{\overline{\mu}+1} T_{j-1}^{(t)}\cdot S_{i,j}^{(t)} \\
&= \sum_{t=0}^{\overline{\nu}}\sum_{j=0}^{\overline{\mu}} T_j^{(t)}\cdot S_{i+1,j}^{(t)} \quad \textit{(Hankel)} \\
&= \langle x^{i+1}T(x,y), S(x,y)\rangle
\end{aligned}$$

which is zero for the rows of index $i\in[\kappa_{\overline{\nu}}-1]_0$. ∎

Similarly to the FIA for one Hankel matrix we can start examining a new $\overline{\mu}$th column of the sub-matrix $\mathbf{S}^{(\overline{\nu})}$ in row $\kappa_{\overline{\nu}}-1$. Note that the previous value of the row pointer $\kappa_{\overline{\nu}}$ is stored in $R[\overline{\nu}]$.

Before Algorithm 1 enters a new column, the coefficients of the intermediate bivariate connection polynomial $T(x,y)$ give us the vanishing linear combination of the sub-matrix consisting of the first κ_ν rows and $C_{\nu,\mu}$ previous columns of the rearranged matrix $\mathbf{S}'$ (see (25)). The following theorem summarizes the properties of Algorithm 1.

Theorem 2 (Algorithm 1): Let $\mathbf{S} = ||\mathbf{S}^{(0)}\,\mathbf{S}^{(1)}\,\cdots\,\mathbf{S}^{(\ell)}||$ be the $n\times\sum_{t=0}^{\ell}N_t$ matrix as defined in (23) and $S(x,y)$ the associated bivariate syndrome polynomial for the reformulated Sudan interpolation problem. Algorithm 1 returns a bivariate polynomial $T(x,y)$ such that:

$$\langle x^\kappa T(x,y), S(x,y)\rangle = 0, \quad \kappa\in[n]_0.$$

The time complexity of Algorithm 1 is $\mathcal{O}\left(\ell n^2\right)$.

Proof: The correctness of Algorithm 1 follows from the correctness of the basic FIA (see Theorem 1) and from the correctness of the initialization rule (Proposition 2) considering that Algorithm 1 deals with the column-permuted version $\mathbf{S}'$ of the original matrix $\mathbf{S} = ||\mathbf{S}^{(0)}\,\mathbf{S}^{(1)}\,\cdots\,\mathbf{S}^{(\ell)}||$.

The proof of the complexity of Algorithm 1 is as follows. We trace the triple:

$$\big((\nu,\mu),(\kappa_0,\kappa_1,\ldots,\kappa_\ell),\delta\big).$$

where (ν,μ) is the current column pointer of Algorithm 1 examining the μth column of the νth sub-matrix $\mathbf{S}^{(\nu)}$. The variables κ_0,κ_1,...,κ_ℓ are the values of the last row reached in the sub-matrices $\mathbf{S}^{(0)},\mathbf{S}^{(1)},\ldots,\mathbf{S}^{(\ell)}$. These values are stored in the array R in Algorithm 1. The value δ is the number of already encountered true discrepancies of Algorithm 1. Assume (ν,μ) is the current column pointer of Algorithm 1. The two following events in Algorithm 1 can happen:
1) Either, there is no true discrepancy, then Algorithm 1 stays in the same column and κ_ν increases by one. The triple becomes

$$\big((\nu,\mu),(\kappa_0,\kappa_1,\ldots,\kappa_\nu\leftarrow\kappa_\nu+1,\ldots,\kappa_\ell),\delta\big).$$

2) Or, there is a true discrepancy, then Algorithm 1 examines column $(\overline{\nu},\overline{\mu}) = \mathrm{succ}(\prec_H,(\nu,\mu))$ and the triple becomes

$$\big((\overline{\nu},\overline{\mu}),(\kappa_0,\kappa_1,\ldots,\kappa_{\overline{\nu}}\leftarrow\kappa_{\overline{\nu}}-1,\ldots,\kappa_\ell),\delta\leftarrow\delta+1\big).$$

For both cases, the sum Iter over the triple is

$$\mathsf{Iter} = C_{\nu,\mu} + \left(\sum_{t \in [\ell+1]_0} \kappa_t \right) + \delta, \tag{28}$$

when Algorithm 1 examines the (ν, μ)th column of the matrix $||\mathbf{S}^{(0)}\, \mathbf{S}^{(1)} \cdots \mathbf{S}^{(\ell)}||$. From (27), we have $C_{\mathrm{succ}(\prec_H, (\nu,\mu))} = C_{(\nu,\mu)} + 1$. The sum Iter increases by one in each iteration of Algorithm 1. The initial value of Iter is zero and the last value can be bounded by:

$$\mathsf{Iter} < \mathcal{O}(n) + \mathcal{O}(\ell n) + \mathcal{O}(n) \leq \mathcal{O}(\ell n).$$

Each discrepancy computation costs $O(n)$ and Algorithm 1 does not have to examine more than the $(n+1)$th columns of the $n \times \sum_{t=0}^{\ell} N_t$ matrix $||\mathbf{S}^{(0)}\, \mathbf{S}^{(1)} \cdots \mathbf{S}^{(\ell)}||$. Thus, the total cost of Algorithm 1 is $O(\ell n^2)$. ∎

In the following, we illustrate the values of the row pointer κ of Algorithm 1, when applied to a syndrome matrix $\mathbf{S} = ||\mathbf{S}^{(0)}\, \mathbf{S}^{(1)}\, \mathbf{S}^{(2)}||$ that consists of three Hankel matrices.

C. Example: Sudan Decoding of a Generalized Reed–Solomon Code with Adapted FIA

We consider a $\mathcal{GRS}(16, 4)$ code over GF(17). For a decoding radius $\tau = 7 = \lfloor (n-k)/2 \rfloor + 1$, the list size is $\ell = 2$. The degrees of the three univariate polynomials $Q^{(0)}(x)$, $Q^{(1)}(x)$ and $Q^{(2)}(x)$ are limited to $(N_0, N_1, N_2) = (9, 6, 3)$ and we have more unknowns than interpolation constraints $(N_0 + N_1 + N_2 > n)$.

Figure 2 illustrates the row pointer of Algorithm 1 when the 16×18 syndrome matrix $||\mathbf{S}^{(0)}\, \mathbf{S}^{(1)}\, \mathbf{S}^{(2)}||$ is examined. The columns of the syndrome matrix are virtually rearranged according to the $\prec_H$-ordering and Algorithm 1 scans the rearranged matrix $\mathbf{S}'$ column by column. The column-index $C_{\nu,\mu}$ (see (27)) and the corresponding column pointer (ν, μ) are listed in Table I.

The three zig–zag lines $S^{(0)}$, $S^{(1)}$ and $S^{(2)}$ in Figure 2 trace the value of the row pointer κ for the three sub-matrices $\mathbf{S}^{(0)}$, $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$, which have a Hankel structure. The dots indicate the case, where a true discrepancy occurs. After the kth column (here $k-1 = 3$), every second column corresponds to the same sub-matrix.

After column 10 of the rearranged matrix $\mathbf{S}'$, every third column of $\mathbf{S}'$ corresponds to the same sub-matrix $\mathbf{S}^{(\nu)}$. Let us investigate two cases, where a true discrepancy in Algorithm 1 occurs. They are marked in column $C_{0,7} = 12$ and $C_{0,8} = 15$ of the re-arranged $\mathbf{S}'$ in Figure 2. In between column 12 and 15 one column of the sub-matrices $\mathbf{S}^{(1)}$ and $\mathbf{S}^{(2)}$ is examined by Algorithm 1. In column $(0, 8)$, Algorithm 1 starts investigating the second row, because the true discrepancy in column $(0, 7)$ occurred in the third row (according to Proposition 2).

D. The FIA for a Vertical Band of Hankel Matrices

The FIA can also be adapted to a matrix consisting of Hankel matrices arranged vertically. This case has been considered for example in [2], [32]. The basic idea for such a vertical band of Hankel matrices is the same as in the previous case. The rows of each sub-matrix of Hankel structure are scanned in a similar interleaving order as the columns of the previous case.

The obtained time complexity for a vertical band of s Hankel matrices, where each sub-matrix consist of N columns, is $\mathcal{O}\left(sN^2\right)$.

V. Guruswami–Sudan Interpolation Step with a Block-Hankel Matrix

A. The Guruswami–Sudan Interpolation Step for Generalized Reed–Solomon Codes

We consider again a Generalized Reed–Solomon code with support set $\alpha_1, \alpha_2, \ldots, \alpha_n$, multipliers $v'_1, v'_2, \ldots, v'_n$ and dimension k, as introduced in Section II. Let $v_1, v_2, \ldots, v_n$ according to (3) be the multipliers of the dual Generalized Reed–Solomon code.

Let $(r_1, r_2, \ldots, r_n)$ be the received word. The Guruswami–Sudan decoding principle [3]–[5] improves the previous algorithms by introducing an additional parameter s, which is the order of multiplicity for the n points $(\alpha_1, r_1/v'_1), (\alpha_2, r_2/v'_2), \ldots, (\alpha_n, r_n/v'_n)$. The parameter s influences the decoding radius τ and the list size ℓ. The relationship between these parameters has been discussed in many publications (see e.g. [33]).

Problem 3 (Guruswami–Sudan Interpolation Step [3]): Let the aimed decoding radius τ, the multiplicity s and the received word $(r_1, r_2, \ldots, r_n)$ be given. The Guruswami–Sudan interpolation step determines a polynomial $Q(x, y) = \sum_{t=0}^{\ell} Q^{(t)}(x) y^t \in \mathbb{F}[x, y]$, such that

1) $Q(x, y) \neq 0$,
2) $\mathrm{mult}\left(Q(x, y), (\alpha_i, r_i/v'_i)\right) \geq s, \quad \forall i \in [n]$,
3) $\mathrm{wdeg}_{1,\text{k-1}}\, Q(x, y) < s(n - \tau)$.

As in the previous section, let N_t denote the degree of the $\ell + 1$ univariate polynomials $Q^{(t)}(x)$. From Condition 3) of Problem 3 we get:

$$\deg Q^{(t)}(x) < N_t \stackrel{\text{def}}{=} s(n-\tau) - t(k-1), \quad t \in [\ell+1]_0. \tag{29}$$

B. Univariate Reformulation of the Guruswami–Sudan Interpolation Problem and A Block-Hankel Matrix

We reformulate the Guruswami–Sudan interpolation problem to obtain not one, but a system of several Extended Key Equations. The corresponding homogeneous linear system has a Block-Hankel form.

Proposition 3 (Univariate Reformulation): Let the integers s, τ, ℓ and the received vector $(r_1, r_2, \ldots, r_n)$ be given. Let $R(x)$ be the Lagrange interpolation polynomial, such that $R(\alpha_i) = r_i/v'_i$, $i \in [n]$. Let $G(x) = \prod_{i=1}^{n} (x - \alpha_i)$. A polynomial $Q(x, y)$ satisfies Conditions 2) and 3) of Problem 3, if and only if there exist s polynomials $B^{(b)}(x) \in \mathbb{F}[x]$ such that

$$Q^{[b]}(x, R(x)) = B^{(b)}(x) \cdot G(x)^{s-b}, \tag{30}$$

and $\deg B^{(b)}(x) < \ell(n-k) - s\tau + b$, $b \in [s]_0$.

Note that $Q^{[b]}(x, y)$ denotes the bth Hasse derivative of the bivariate polynomial $Q(x, y)$ with respect to the variable y (see Definition 3).

We first prove the following lemma.

Lemma 2: Let $(\alpha_i, r_i) \in \mathbb{F}^2$ be given, and let $R(x) \in \mathbb{F}[x]$ be any polynomial such that $R(\alpha_i) = r_i$. A polynomial $Q(x,y)$ has multiplicity at least s at (α_i, r_i) if and only if $(x-\alpha_i)^{s-b} | Q^{[b]}(x, R(x))$, $\forall\, b \in [s]_0$.

Proof: After translation to the origin, we can assume that $(\alpha_i, r_i) = (0,0)$, and $R(0) = 0$, i.e., $x | R(x)$. Let $Q(x,y) = \sum_i Q_i(x,y)$, where $Q_i(x,y)$ is homogeneous of degree i.

We first suppose that $Q(x,y)$ has at least a multiplicity s at $(0,0)$, i.e., $Q_i(x,y) = 0$, for $i < s$. Hence, we have

$$Q^{[b]}(x, R(x)) = \sum_{i \geq s-b} Q_i^{[b]}(x, R(x)).$$

For $b < s$, the polynomials $Q_i^{[b]}(x,y)$ have no terms of degree less than $s-b$, and with $x | R(x)$, we have $x^{s-b} | Q_i^{[b]}(x, R(x))$. It follows, that x^{s-b} divides $Q^{[b]}(x, R(x))$ for all $b \in [s]_0$.

Suppose for the converse that $x^{s-b} | Q^{[b]}(x, R(x))$. That is, $Q^{[b]}(x, R(x)) = x^{s-b} U^{(b)}(x)$, for some polynomials $U^{(b)}(x)$ and $b \in [s]_0$. Using Taylor's formula with the Hasse derivatives [22, p. 89] we have:

$$\begin{aligned} Q(x,y) &= \sum_b Q^{[b]}(x, R(x)) \cdot (y - R(x))^b \\ &= \sum_{b<s} Q^{[b]}(x, R(x)) \cdot (y - R(x))^b \\ &\quad + \sum_{b \geq s} Q^{[b]}(x, R(x)) \cdot (y - R(x))^b \\ &= \sum_{b<s} x^{s-b} U^{(s-b)}(x) \cdot (y - R(x))^b \\ &\quad + \sum_{b \geq s} Q^{[b]}(x, R(x)) \cdot (y - R(x))^b. \end{aligned}$$

Now, $(y - R(x))^b$ has only terms of degree higher than b, since $x | R(x)$. Thus, we have no terms of degree less than s in $Q(x,y)$. ∎

Proof of Proposition 3: From the previous lemma, we know that $(x-\alpha_i)^{s-b} | Q^{[b]}(x, R(x))$, $b \in [s]_0$, $i \in [n]$. Since all $(x-\alpha_i)$'s are distinct the Chinese Remainder Theorem for univariate polynomials implies that $G(x)^{s-b} | Q^{[b]}(x, R(x))$. The degree condition follows easily. ∎

Proposition 3 enables to rewrite the s equations (30) more explicitly:

$$\begin{aligned} \sum_{t=b}^{\ell} \binom{t}{b} Q^{(t)}(x) \cdot R(x)^{t-b} = \\ B^{(b)}(x) \cdot G(x)^{s-b}, \quad b \in [s]_0. \end{aligned} \tag{31}$$

As usual, let the reciprocal polynomials be:

$$\begin{aligned} \overline{R}(x) &= x^{n-1} \cdot R(x^{-1}), \\ \overline{G}(x) &= x^n \cdot G(x^{-1}) = \prod_{i=1}^{n} (1 - \alpha_i x), \\ \overline{B}^{(b)}(x) &= x^{\ell(n-k) - s\tau - b - 1} \cdot B(x^{-1}), \\ \Lambda^{(t)}(x) &= x^{N_t - 1} \cdot Q^{(t)}(x^{-1}). \end{aligned}$$

Inserting them into (31), leads to:

$$\begin{aligned} \sum_{t=b}^{\ell} \binom{t}{b} \Lambda^{(t)}(x) \cdot x^{(\ell-t)(n-k)} \cdot \overline{R}(x)^{t-b} \\ = \overline{B}^{(b)}(x) \cdot \overline{G}(x)^{s-b}, \end{aligned} \tag{32}$$

Since $G(x)$ is relatively prime to $x^{(\ell-b)(n-k)}$, it admits an inverse modulo $x^{(\ell-b)(n-k)}$. The Taylor series of $\overline{R}(x)^{t-b} / \overline{G}(x)^{s-b}$ is denoted by $T^{(b,t)}(x)$. Then (32) leads to s equations:

$$\begin{aligned} \sum_{t=b+1}^{\ell} \binom{t}{b} \Lambda^{(t)}(x) \cdot x^{(\ell-t)(n-k)} \cdot T^{(b,t)}(x) \\ \equiv \overline{B}^{(b)}(x) \bmod x^{(\ell-b)(n-k)}, \quad b \in [s]_0. \end{aligned}$$

where each equation is denoted by $\mathrm{EKE}(b)$. Note that the degree of $\overline{B}^{(b)}(x)$ can be greater than $(\ell-b)(n-k)$ and it is not clear how to properly truncate this identity, as in [9], [10], noted in Note 1, or as in the case of the classical Key Equation (see Section III).

In the following, we consider the complete system of $\binom{s+1}{2} n$ homogeneous linear equations. We have $\deg Q^{[b]}(x, R(x)) = s(n-\tau) + \ell(n-k) - b(n-1)$. We obtain s equations for the bth derivative with the following truncation:

$$\begin{aligned} \sum_{t=b}^{\ell} \binom{t}{b} \Lambda^{(t)}(x) \cdot x^{(\ell-t)(n-k)} \cdot T^{(b,t)}(x) \\ \equiv \overline{B}^{(b)}(x) \bmod x^{s(n-\tau)+\ell(n-k)-b(n-1)}, \quad b \in [s]_0. \end{aligned} \tag{33}$$

Let us write $\mathrm{EKE}_0(b)$ for the bth equation as above.

Proposition 4: Let $d = n - k + 1$ be the minimum distance of the considered $\mathcal{GRS}(n,k)$ code. Let b be such that $s\tau - bd \geq 0$. If $\Lambda^{(b+1)}(x), \ldots, \Lambda^{(\ell)}(x)$ is a solution to $\mathrm{EKE}(b)$, then there exists $\Lambda^{(b)}(x)$ such that $\Lambda^{(b)}(x), \Lambda^{(b+1)}(x), \ldots, \Lambda^{(\ell)}(x)$ is a solution to $\mathrm{EKE}_0(b)$.

Proof: Let us consider (31). We isolate $Q^{(b)}(x)$ and get

$$\begin{aligned} Q^{(b)}(x) + \sum_{t=b+1}^{\ell} \binom{t}{b} Q^{(t)}(x) \cdot R(x)^{t-b} \\ = B^{(b)}(x) \cdot G(x)^{s-b}. \end{aligned} \tag{34}$$

and thus $Q^{(b)}(x)$ is the remainder of the Euclidean division of $\sum_{t=b+1}^{\ell} \binom{t}{b} Q^{(t)}(x) R(x)^{t-b}$ by $G(x)^{s-b}$, as long as $\deg Q^{(b)}(x) < \deg G(x)^{s-b}$, which gives $s(n-\tau) - b(k-1) \leq (s-b)n$, i.e., $s\tau - bd \geq 0$. ∎

Note 2: We denote $b_0 = \lfloor (s\tau)/d \rfloor$. Actually, we can consider (32) and substitute the $\Lambda^{(b)}(x)$, for $b \in [b_0]$, successively. This is possible for the case of the first order system ($s = 1$), noted in Note 1. In the more general Guruswami–Sudan case, we can obtain a reduced system with $\Lambda^{(b_0+1)}(x), \ldots, \Lambda^{(\ell)}(x)$, but it seems that this reduced system lost its Block-Hankel structure. Thus, there are no benefits of reducing the number of unknowns. We could not find a proper interpretation of the quantity $b_0 = \lfloor (s\tau)/d \rfloor$.

With (33), we now can define the syndrome polynomials for the reformulated Guruswami–Sudan interpolation problem.

Definition 7 (Syndromes for Guruswami–Sudan): The syndrome polynomials $S^{(0,0)}(x), S^{(0,1)}(x), \ldots, S^{(0,\ell)}(x), S^{(1,1)}(x), \ldots, S^{(s-1,\ell)}(x)$ with $S^{(b,t)}(x) = \sum_{i=0}^{(s-b)n+N_t-1} S_i^{(b,t)} x^i$ are given by:

$$S_i^{(b,t)} = T_{i+(b+1+t(n-1)-sn)}^{(b,t)}, \quad \begin{matrix} b \in [s]_0, \\ t = b, \ldots, \ell, \end{matrix} \tag{35}$$

where $T^{(b,t)}(x)$ denotes the power series of $\overline{R}(x)^{t-b}/\overline{G}(x)^{s-b}$.

The (sth order) **Extended Key Equations** are

$$\sum_{t=b}^{\ell} \Lambda^{(t)}(x) \cdot S^{(b,t)}(x) \equiv \overline{B}^{(b)}(x) \bmod x^{s(n-\tau)+\ell(n-k)-b(n-1)}, \tag{36}$$

with $\deg \overline{B}^{(b)}(x) < \ell(n-k) - s\tau + b$, $b \in [s]_0$.

The explicit expression for $S_i^{(b,t)}$ is difficult to obtain. We claim that it will not be easier to compute $S_i^{(b,t)}$ with such a formula than by calculating the power series expansion of $T^{(b,t)}(x) = \overline{R}(x)^{t-b}/\overline{G}(x)^{s-b}$, which is fast to compute by computer algebra techniques.

Considering the high degree terms, we get $\sum_{b=0}^{s-1}(s-b)n = \binom{s+1}{2}n$ homogeneous equations from (36), which can be written as:

$$\sum_{t=b}^{\ell} \sum_{i=0}^{N_t-1} Q_i^{(t)} \cdot S_{j+i}^{(b,t)} = 0, \quad \begin{matrix} j \in [(s-b)n]_0, \\ b \in [s]_0. \end{matrix} \tag{37}$$

These linear equations lead to a Block-Hankel matrix. The syndrome matrix $\mathbf{S} = ||\mathbf{S}^{(t,b)}||$ for all $t \in [\ell+1]_0, b \in [s]_0$ of the reformulated Guruswami–Sudan interpolation problem has the following form:

$$\begin{pmatrix} \mathbf{S}^{(0,0)} & \mathbf{S}^{(0,1)} & \ldots & \ldots & \ldots & \mathbf{S}^{(0,\ell)} \\ \mathbf{0} & \mathbf{S}^{(1,1)} & \ldots & \ldots & \ldots & \mathbf{S}^{(1,\ell)} \\ \vdots & & \ddots & & \vdots & \\ \mathbf{0} & \ldots & \mathbf{0} & \mathbf{S}^{(s-1,s-1)} & \ldots & \mathbf{S}^{(s-1,\ell)} \end{pmatrix}, \tag{38}$$

where each sub–matrix $\mathbf{S}^{(b,t)} = ||S_{i,j}^{(b,t)}||$ is an $n(s-b) \times N_t$ Hankel matrix and $S^{(b,t)}(x) = \sum_{i=0}^{(s-b)n+N_t-1} S_i^{(t,b)} x^i$ are the associated polynomials with $S_{i,j}^{(b,t)} = S_{i+j}^{(b,t)}$. All matrices depend on the received vector $\mathbf{r}$ except the ones on the diagonal: $\mathbf{S}^{(i,i)}$, $i \in [s]_0$.

C. The FIA for the Block-Hankel Matrix of the Reformulated Guruswami–Sudan Interpolation Problem

We adapt the FIA to the Block-Hankel matrix of (38). The structure of this syndrome matrix is a mixture of the syndrome matrix (see Definition 2) of the reformulated Sudan interpolation problem and a vertical arrangement of many Hankel matrices. The extension of the FIA for this case was hinted in [10, Section 5.2]. First of all, let us express the s Key Equations of (37) in terms of the inner product of bivariate polynomials.

Problem 4 (Reformulated Guruswami–Sudan Problem): Let $S^{(b)}(x,y),\ b \in [s]_0$ be s bivariate syndrome polynomials with:

$$S^{(b)}(x,y) = \sum_{t=b}^{\ell} \sum_{i=0}^{N_t+(s-b)n-1} S_i^{(b,t)} x^i y^t, \tag{39}$$

where the coefficients $S_i^{(b,t)}$ are given in Definition 7. We search a nonzero bivariate polynomial $T(x,y)$ that fulfills:

$$\langle x^\kappa T(x,y), S^{(\vartheta)}(x,y) \rangle = 0, \quad \begin{matrix} \vartheta \in [s]_0, \\ \kappa \in [(s-\vartheta)n]_0. \end{matrix} \tag{40}$$

Algorithm 2: Algorithm Solving Problem 4

Input: s bivariate polynomials $S^{(b)}(x,y) = \sum_{t=0}^{\ell} S^{(t,b)}(x) y^t, b \in [s]_0$;
Output: Bivariate polynomial $T(x,y)$;

Data structures:
Bivariate polynomial $T(x,y) = \sum_{t=0}^{\ell} T^{(t)}(x) y^t$;
Column pointers (ν, μ), where $\nu \in [\ell]_0$, $\mu \in [N_\nu]_0$;
Row pointer (ϑ, κ), where $\vartheta \in [s]_0$ and $\kappa \in [(s-\vartheta)n]_0$;
Two–dimensional array $A[(i,j)]$ of $\binom{s+1}{2}n$ entries in $\mathbb{F}[x,y]$ indexed with the row pointer (ϑ, κ);
Two–dimensional array $D[(i,j)]$ of $\binom{s+1}{2}n$ entries in $\mathbb{F}$ indexed with the row pointer (ϑ, κ);
Array R of $\ell+1$ entries containing the row pointer (ϑ, κ);
Variable $\Delta \in \mathbb{F}$, variable *compute* $\in \{\text{TRUE}, \text{FALSE}\}$;

Initialize:
Initialize arrays A, D and C to zero;
$(\nu, \mu) \leftarrow (0,0)$ and $(\vartheta, \kappa) \leftarrow (0,0)$;
compute $\leftarrow$ FALSE;

while $(\vartheta, \kappa) < (s, 0)$ **do**
  if *compute* **then**
    $\Delta \leftarrow \langle x^\kappa \cdot T(x,y), S^{(\vartheta)}(x,y) \rangle$;
  else
    if $R[\nu] < 1$ **then**
      $T(x,y) \leftarrow y^\nu \cdot x^\mu$;
      $\Delta \leftarrow S_\mu^{(0,\nu)}$;
      $(\vartheta, \kappa) \leftarrow (0,0)$;
    else
      $T(x,y) \leftarrow x \cdot A[R[\nu]](x,y)$;
      $\Delta \leftarrow D[R[\nu]]$;
      $(\vartheta, \kappa) \leftarrow R[\nu]$;
      if $\kappa = 0$ **then**
        $(\vartheta, \kappa) \leftarrow (\vartheta - 1, n)$;
        $\Delta \leftarrow 0$;
      $\kappa \leftarrow \kappa - 1$;
    compute $\leftarrow$ TRUE;
  if $\Delta = 0$ *or* $D[(\vartheta, \kappa)] \neq 0$ **then**
    if $\Delta \neq 0$ **then**
      $T(x,y) \leftarrow T(x,y) - \frac{\Delta}{D[(\vartheta,\kappa)]} \cdot A[(\vartheta, \kappa)](x,y)$;
    $(\vartheta, \kappa) \leftarrow \text{succ}(\prec_V, (\vartheta, \kappa))$;
  else /* $\Delta \neq 0$ *and* $D[(\vartheta, \kappa)] = 0$ */
    $A[(\vartheta, \kappa)] \leftarrow T(x,y)$;
    $D[(\vartheta, \kappa)] \leftarrow \Delta$;
    $R[\nu] \leftarrow (\vartheta, \kappa)$;
    compute $\leftarrow$ FALSE;
    $(\nu, \mu) \leftarrow \text{succ}(\prec_H, (\nu, \mu))$;

We adjust the FIA as an algorithm on a row- and column-interleaved version of the Block-Hankel matrix $\mathbf{S}$ of (38). Let us first define an ordering to describe the vertical rearrangement of the rows of the syndrome matrix $\mathbf{S}$ as in (38). Let denote $\prec_V$ the ordering on the rows, indexed by pairs (ϑ, κ), such that:

$$(\vartheta,\kappa) \prec_V (\overline{\vartheta},\overline{\kappa}) \iff \begin{cases} \kappa + \vartheta n < \overline{\kappa} + \overline{\vartheta} n \\ \text{or} \\ \kappa + \vartheta n = \overline{\kappa} + \overline{\vartheta} n \text{ and } \vartheta < \overline{\vartheta}. \end{cases} \tag{41}$$

Let $\mathrm{succ}(\prec_V, (\vartheta,\kappa))$ denote the pair that immediately follows (ϑ,κ) with respect to order defined by $\prec_V$ and let $\mathrm{pred}(\prec_V, (\vartheta,\kappa))$ denote the pair that immediately precedes (ϑ,κ) with respect to order defined by $\prec_V$. Furthermore, let:

$$R_{\vartheta,\kappa} = \big|\{(t,i) \,|\, (t,i) \prec_V (\vartheta,\kappa)\}\big|, \tag{42}$$

which we use to index the rows of the virtually rearranged matrix (similar to the horizontal case). Note that $R_{\mathrm{pred}(\prec_V,(\vartheta,\kappa))} = R_{\vartheta,\kappa} - 1$.

In the following, $\mathbf{S}'$ denotes the rearranged version of the matrix $\mathbf{S}$ of (38), where the columns are ordered under $\prec_H$- and the rows under $\prec_V$-ordering.

Algorithm 2 is the Fundamental Iterative Algorithm tailored to a Block-Hankel matrix as in (38). As in the case of the reformulated Sudan interpolation problem, the columns of the Block-Hankel matrix $\mathbf{S}$ are indexed by a couple (ν,μ), where $\nu \in [\ell+1]_0$ and $\mu \in [N_\nu]_0$. Furthermore, the rows are indexed by a couple (ϑ,κ), where $\vartheta \in [s]_0$ and $\kappa \in [(s-\vartheta)\cdot n]_0$.

Now, the arrays storing the discrepancies and the intermediate polynomials are still indexed by rows, but the indexes of the rows are two-dimensional, leading to two-dimensional arrays. The two-dimensional array A stores the intermediate bivariate polynomials and the two-dimensional array D, stores the discrepancy values. Both arrays A and D are indexed by the row pointer (ϑ,κ). The discrepancy calculation (see Line 20 of Algorithm 2) is adjusted to a Block-Hankel matrix where each sub-horizontal band of Hankel matrices is represented by a bivariate polynomial.

The intermediate bivariate connection polynomial $T^{(\vartheta,\kappa)}(x)$ of Algorithm 2 examining the κth row and the μth column of the (ν,ϑ)th sub-matrix $\mathbf{S}^{(\nu,\vartheta)}$, gives us the vanishing linear combination of the sub-matrix consisting of the first $R_{\vartheta,\kappa}$ rows and the first $C_{\nu,\mu}$ columns of the rearranged syndrome matrix $\mathbf{S}'$.

The row pointer of the sub-block $||\mathbf{S}^{(\nu,0)}\,\mathbf{S}^{(\nu,1)}\,\cdots\,\mathbf{S}^{(\nu,s-1)}||^T$ is stored in the array $R[\nu]$. Note that $\ell+1$ row pointers of the form (ϑ,κ) need to be stored.

The adjusted initialization rule of Algorithm 2 examining the Block-Hankel syndrome matrix as defined in (38) is stated in the following proposition (see Line 16, 21 and 27 of Algorithm 2).

Proposition 5 (Initialization Rule): Assume Algorithm 2 examines column (ν,μ) of a Block-Hankel syndrome matrix $\mathbf{S}$ as defined in (38) or equivalently the s bivariate polynomials $S^{(0)}(x,y), S^{(1)}(x,y), \ldots, S^{(s-1)}(x,y)$ of Problem 4. Assume that a true discrepancy is obtained. Let $(\overline{\nu},\overline{\mu}) = \mathrm{succ}(\prec_H, (\nu,\mu))$ and let (ϑ,κ) be the previously stored value for the index of the last reached row in the sub-matrix of index $\overline{\nu}$, and let $T(x,y)$ be the bivariate polynomial stored for that row. If $(\overline{\vartheta},\overline{\kappa}) = \mathrm{pred}(\prec_V, (\vartheta,\kappa))$, we can start examining column $(\overline{\nu},\overline{\mu})$ of $\mathbf{S}$ at row $(\overline{\vartheta},\overline{\kappa})$ with the initial value $\overline{T}(x,y) = x\cdot T(x,y)$.

Proof: In terms of the inner product (see Definition 4), we have:

$$\langle x^{i_1} T(x,y), S^{(i_2)}(x,y)\rangle = 0, \quad \forall\,(i_2,i_1) \prec_V (\overline{\vartheta},\overline{\kappa}). \tag{43}$$

Let us write $T(x,y) = \sum_{t=0}^{\overline{\nu}} \sum_{j=0}^{\overline{\mu}} T_j^{(t)} x^j y^t$ and $\overline{T}(x,y) = \sum_{t=0}^{\overline{\nu}} \sum_{j=0}^{\overline{\mu}+1} \overline{T}_j^{(t)} x^j y^t$, with $\overline{T}_j^{(t)} = T_{j-1}^{(t)}$, for $j > 0$, and $\overline{T}_0^{(t)} = 0$. Due to the structure of the Block-Hankel matrix $\mathbf{S}$, we have the following identities:

$$\begin{aligned} &\langle x^{i_1}\overline{T}(x,y), S^{(i_2)}(x,y)\rangle \\ &\quad= \sum_{t=0}^{\overline{\nu}} \sum_{j=1}^{\overline{\mu}+1} \overline{T}_j^{(t)} \cdot S_{i_1,j}^{(i_2,t)} \\ &\quad= \sum_{t=0}^{\overline{\nu}} \sum_{j=1}^{\overline{\mu}+1} T_{j-1}^{(t)} \cdot S_{i_1,j}^{(i_2,t)} \\ &\quad= \sum_{t=0}^{\overline{\nu}} \sum_{j=0}^{\overline{\mu}} T_j^{(t)} \cdot S_{i_1,j+1}^{(i_2,t)} \\ &\quad= \sum_{t=0}^{\overline{\nu}} \sum_{j=0}^{\overline{\mu}} T_j^{(t)} \cdot S_{i_1+1,j}^{(i_2,t)} \quad \textit{(Hankel)} \\ &\quad= \langle x^{i_1+1} T(x,y), S^{(i_2)}(x,y)\rangle \end{aligned}$$

which is zero for every $(i_2,i_1) \prec_V (\overline{\vartheta},\overline{\kappa})$. ∎

Theorem 3 (Algorithm 2): Let $\mathbf{S}$ be the $\binom{s+1}{2} n \times \sum_{t=0}^{\ell} N_t$ syndrome Block-Hankel matrix of the reformulated Guruswami–Sudan interpolation problem as in (38) and let $S^{(b)}(x,y), b \in [s]_0$ be the corresponding bivariate syndrome polynomials as defined in Problem 4. Then Algorithm 2 outputs a bivariate polynomial $T(x,y)$, such that:

$$\langle x^{\kappa} T(x,y), S^{(\vartheta)}(x,y)\rangle = 0, \quad \begin{aligned} &\vartheta \in [s]_0, \\ &\kappa \in [(s-\vartheta)n]_0. \end{aligned}$$

The time complexity of Algorithm 2 is $\mathcal{O}\left(\ell s^4 n^2\right)$.

Proof: The correctness is as usual, considering that we deal with the row- and column-permuted version $\mathbf{S}'$ of the Block-Hankel matrix $\mathbf{S}$ and that the initialization rule is correct.

In the following, we analyze the complexity of Algorithm 2. As in Section IV, we describe the state of Algorithm 2 with the following triple:

$$\big((\nu,\mu), ([\vartheta,\kappa]_0, [\vartheta,\kappa]_1, \ldots, [\vartheta,\kappa]_\ell), \delta\big), \tag{44}$$

where (ν,μ) is the current column pointer of Algorithm 2, when examining the μth column of the horizontal band of s vertically arranged Hankel matrices $||\mathbf{S}^{(\nu,0)}\,\mathbf{S}^{(\nu,1)}\,\cdots\,\mathbf{S}^{(\nu,s-1)}||^T$. The index $[\vartheta,\kappa]_\nu$ is the last considered row in the horizontal band of s sub-matrices

$||\mathbf{S}^{(\nu,0)}\,\mathbf{S}^{(\nu,1)}\,\ldots\,\mathbf{S}^{(\nu,s-1)}||^T$. These values are stored in the array R of Algorithm 2. As for Algorithm 1, δ denotes the number of already encountered true discrepancies. Assume (ν,μ) is the current column pointer of Algorithm 2. The same two cases as before can happen:

1) Either, there is no true discrepancy, then Algorithm 2 remains in the same column (ν,μ) of the sub-matrices $||\mathbf{S}^{(\nu,0)}\,\mathbf{S}^{(\nu,1)}\,\ldots\,\mathbf{S}^{(\nu,s-1)}||^T$ and the triple becomes:

$$\big((\nu,\mu),([\vartheta,\kappa]_0,[\vartheta,\kappa]_1,\ldots,\\ [\vartheta,\kappa]_\nu \leftarrow \mathsf{next}[\prec_V,([\vartheta,\kappa]_\nu)],\ldots,[\vartheta,\kappa]_\ell),\delta\big),$$

2) Or, a true discrepancy is encountered and the triple becomes:

$$\big((\overline{\nu},\overline{\mu}),([\vartheta,\kappa]_0,[\vartheta,\kappa]_1,\ldots,\\ [\vartheta,\kappa]_{\overline{\nu}} \leftarrow \mathsf{prev}[\prec_V,([\vartheta,\kappa]_{\overline{\nu}})],\ldots,[\vartheta,\kappa]_\ell),\delta+1\big),$$

where $(\overline{\nu},\overline{\mu}) \leftarrow \mathrm{succ}(\prec_H,(\nu,\mu))$. In both cases, the sum Iter of the triple is:

$$\mathsf{Iter} = C_{\nu,\mu} + \left(\sum_{t\in[\ell+1]_0} R_{[\vartheta,\kappa]_t}\right) + \delta, \tag{45}$$

when Algorithm 2 examines the (ν,μ)th column of the Block-Hankel matrix $\mathbf{S}$ of (38) and it increases by one in each iteration. The initial value of Iter is zero, and the final value can be bounded by

$$\begin{aligned}\mathsf{Iter} &\le \binom{s+1}{2}n + \sum_{t=0}^{\ell}\binom{s+1}{2}n + \binom{s+1}{2}n\\ &\le \mathcal{O}\left(\ell s^2 n\right).\end{aligned}$$

The number of iterations of Algorithm 2 is bounded by $\mathcal{O}\left(\ell s^2 n\right)$.

This gives a total of $\mathcal{O}\left(\ell s^4 n^2\right)$, since the discrepancy calculation requires $\mathcal{O}\left(s^2 n\right)$. ∎

D. Example: Guruswami–Sudan Decoding of a Generalized Reed–Solomon Code with Adapted FIA

We consider the case of multiplicity $s = 2$ for the $\mathcal{GRS}(16,4)$ code. The corresponding list size is $\ell = 4$. The decoding radius is now $\tau = 8$. The degrees of the univariate polynomials $Q^{(t)}(x)$ are $(N_0,N_1,N_2,N_3,N_4) = (16,13,10,7,4)$.

The Block-Hankel syndrome matrix $\mathbf{S}$

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}^{(0,0)} & \mathbf{S}^{(0,1)} & \mathbf{S}^{(0,2)} & \mathbf{S}^{(0,3)} & \mathbf{S}^{(0,4)} \\ \mathbf{0} & \mathbf{S}^{(1,1)} & \mathbf{S}^{(1,2)} & \mathbf{S}^{(1,3)} & \mathbf{S}^{(1,4)} \end{pmatrix}$$

is a $(3n = 48) \times (\sum_{t=0}^{4} N_t = 50)$ matrix. It consists of nine non-zero Hankel matrices and one all-zero matrix $\mathbf{S}^{(1,0)}$ arranged in two horizontal bands of five Hankel matrices. The values of the row pointer (ϑ,κ) of Algorithm 2 for the Block-Hankel matrix are traced in Figure 3. The five zig–zag lines $S^{(\vartheta,0)},S^{(\vartheta,1)},\ldots,S^{(\vartheta,4)}$ in Figure 3 trace the row pointer (ϑ,κ), when Algorithm 2 examines the five sub-blocks $||\mathbf{S}^{(0,0)}\,\mathbf{S}^{(1,0)}||^T$, $||\mathbf{S}^{(0,1)}\,\mathbf{S}^{(1,1)}||^T$, $\ldots$, $||\mathbf{S}^{(0,4)}\,\mathbf{S}^{(1,4)}||^T$. In Table II, the column $C_{\nu,\mu}$ and the column pointer (ν,μ) according to $\prec_H$ for the syndrome matrix of the $\mathcal{GRS}(16,4)$ code are listed. Additionally to the horizontal ordering $\prec_H$ of the columns (as in the Sudan case), now the rows are ordered according to $\prec_V$. The row-index $R_{\vartheta,\kappa}$ and the row pointer (ϑ,κ) are shown in Table III. Let us consider three cases,

TABLE II
COLUMN-INDEX $C_{\nu,\mu}$ AND COLUMN POINTER (ν,μ) FOR THE BLOCK-HANKEL SYNDROME MATRIX OF A $\mathcal{GRS}(16,4)$ CODE WITH MULTIPLICITY $s=2$ AND LIST SIZE $\ell=4$.

	Column $C_{\nu,\mu}$ and Column pointer (ν,μ)								
0	(0,0)	10	(1,3)	20	(2,3)	30	(0,12)	40	(0,14)
1	(0,1)	11	(2,0)	21	(3,0)	31	(1,9)	41	(1,11)
2	(0,2)	12	(0,7)	22	(0,10)	32	(2,6)	42	(2,8)
3	(0,3)	13	(1,4)	23	(1,7)	33	(3,3)	43	(3,5)
4	(1,0)	14	(2,1)	24	(2,4)	34	(4,0)	44	(4,2)
5	(0,4)	15	(0,8)	25	(3,1)	35	(0,13)	45	(0,15)
6	(1,1)	16	(1,5)	26	(0,11)	36	(1,10)	46	(1,12)
7	(0,5)	17	(2,2)	27	(1,8)	37	(2,7)	47	(2,9)
8	(1,2)	18	(0,9)	28	(2,5)	38	(3,4)	48	(3,6)
9	(0,6)	19	(1,6)	29	(3,2)	39	(4,1)	49	(4,3)

where a true discrepancy in Algorithm 2 occurred. The first case are the most left two points in Figure 3. The value of the column pointer (ν,μ) is $(0,2)$ and $(0,3)$. Algorithm 3 examines the first band of the two Hankel matrices $||\mathbf{S}^{(0,0)}\,\mathbf{S}^{(1,0)}||^T$ traced by line $S^{(\vartheta,0)}$. For the first pair no columns were virtually interchanged and the horizontal distance is one.

The second two points with the values of the column pointer $(0,5)$ and $(0,6)$ indicate a true discrepancy of Algorithm 3, when the second band of the two Hankel matrices $||\mathbf{S}^{(0,1)}\,\mathbf{S}^{(1,1)}||^T$ is examined. The values are traced by the line $S^{(\vartheta,1)}$ in Figure 3. For the second pair $((0,5),(0,6))$, the columns of the first and second vertical band of Hankel matrices are mixed and therefore the horizontal distance is two.

TABLE III
ROW-INDEX $R_{\vartheta,\kappa}$ AND ROW POINTER (ϑ,κ) OF ALGORITHM 2 FOR BLOCK-HANKEL SYNDROME MATRIX OF A $\mathcal{GRS}(16,4)$ CODE WITH MULTIPLICITY $s=2$ AND LIST SIZE $\ell=4$.

Row $R_{\vartheta,\kappa}$ and Row pointer (ϑ,κ)	
0 - 16	(0,0 - 16)
17	(1,0)
18	(0,17)
19	(1,1)
20	(0,18)
⋮	⋮
46	(0,31)
47	(1,15)

The third considered case, where a true discrepancy occurs, are the most right two points in Figure 3 indicated by values $(1,10)$ and $(1,11)$ of the row pointer (ϑ,κ). Algorithm 2 examines the band of the two Hankel matrices $||\mathbf{S}^{(0,3)}\,\mathbf{S}^{(1,3)}||^T$ and restarts (at the point $(1,10)$) with the previous stored value of the row pointer (at $(1,11)$). In between four other horizontal bands of matrices were examined.

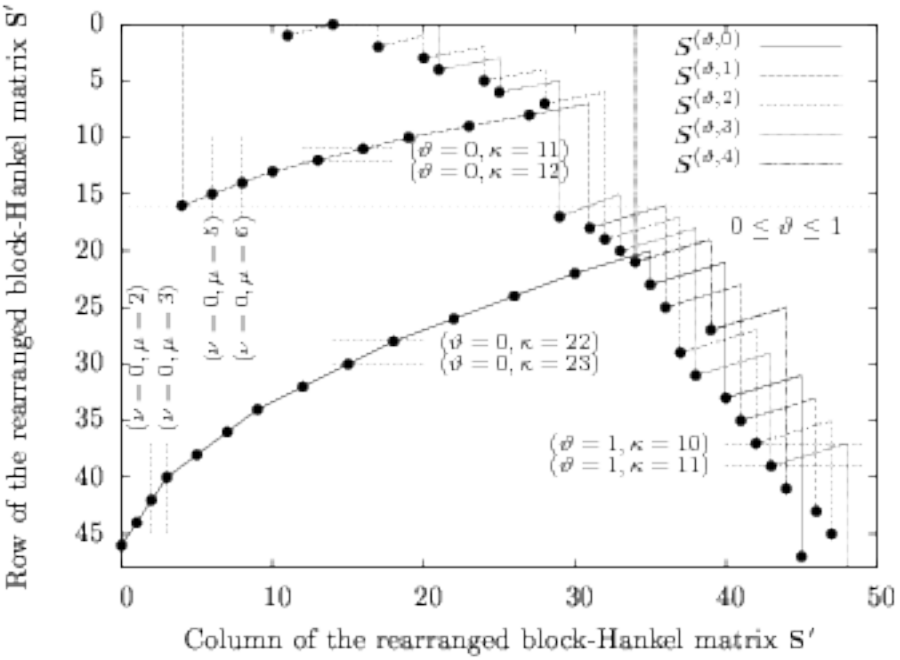


Fig. 3. Illustration of the row pointer (ϑ, κ) of Algorithm 2 applied to a 48×50 Block-Hankel matrix $\mathbf{S}$. The matrix consists of two vertically arranged bands of five horizontally arranged Hankel matrices. The first band consists of 32 rows and the second one of 16. The plotted matrix $\mathbf{S}'$ consists of the rearranged columns and rows of the matrix $\mathbf{S}$ under $\prec_H$- respective $\prec_V$-ordering . The mixture of rows of the two vertical lines starts in line 16 (marked by the dotted horizontal line). The five zig–zag lines $S^{(\vartheta,0)}, S^{(\vartheta,1)}, \ldots, S^{(\vartheta,4)}$ trace the row pointer for the five sub-blocks $||\mathbf{S}^{(0,0)}\,\mathbf{S}^{(1,0)}||^T$, $||\mathbf{S}^{(1,0)}\,\mathbf{S}^{(1,1)}||^T$, $\ldots$, $||\mathbf{S}^{(4,0)}\,\mathbf{S}^{(4,1)}||^T$ of two vertically arranged Hankel matrices.

VI. Conclusion

We reformulated the Guruswami–Sudan interpolation conditions (for a multiplicity higher than one) for Generalized Reed–Solomon codes into a set of univariate polynomial equations, which can partially be seen as Extended Key Equations. The obtained set of homogeneous linear equations has a Block-Hankel structure. We adapted the Fundamental Iterative Algorithm of Feng and Tzeng to this special structure and achieved a significant reduction of the time complexity.

As mentioned in Note 2, the set of equations can be further reduced, under the observation that the diagonal terms are constant, i.e., they do not depend on the received word. This reduction leads to a loss of the Block-Hankel structure and therefore would destroy the quadratic complexity. We note that Beelen and Høholdt [34] mentioned this reduction for the Guruswami–Sudan interpolation step for Algebraic Geometric codes, to get a smaller interpolation problem, but the system does not appear to be Block-Hankel.

We conclude that we identified the quantity $\lfloor (s\tau)/d \rfloor$ (see Note 2) without having found an interpretation of that number.

VII. Acknowledgment

The authors thank Vladimir Sidorenko and Martin Bossert for fruitful discussions.

We thank the anonymous referees for their valuable comments that improved the presentation of this paper.

References

[1] D. Augot and A. Zeh, "On the Roth and Ruckenstein Equations for the Guruswami-Sudan Algorithm," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, 2008, pp. 2620–2624. [Online]. Available: http://dx.doi.org/10.1109/ISIT.2008.4595466

[2] A. Zeh, C. Gentner, and M. Bossert, "Efficient List-Decoding of Reed-Solomon Codes with the Fundamental Iterative Algorithm," in *Information Theory Workshop, 2009. ITW 2009. IEEE*, October 2009. [Online]. Available: http://dx.doi.org/10.1109/ITW.2009.5351241

[3] V. Guruswami and M. Sudan, "Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, 1999. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=782097

[4] V. Guruswami, *List Decoding of Error-Correcting Codes*, ser. Lecture Notes in Computer Science. New York: Springer, 2004, no. 3282.

[5] ——, *Algorithmic Results in List Decoding*. Now Publishers Inc, January 2007.

[6] M. Sudan, "Decoding of Reed-Solomon Codes beyond the Error-Correction Bound," *Journal of Complexity*, vol. 13, no. 1, pp. 180–193, March 1997. [Online]. Available: http://dx.doi.org/10.1006/jcom.1997.0439

[7] I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960. [Online]. Available: http://dx.doi.org/10.1137/0108018

[8] R. Koetter, "On Algebraic Decoding of Algebraic-Geometric and Cyclic Codes," Ph.D. dissertation, University of Linköping, 1996.

[9] R. M. Roth and G. Ruckenstein, "Efficient Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance," *Information Theory, IEEE Transactions on*, vol. 46, no. 1, pp. 246–257, 2000. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=817522

[10] G. Ruckenstein, "Error Decoding Strategies for Algebraic Codes," Ph.D. dissertation, Technion, 2001. [Online]. Available: http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi/2001/PHD/PHD-2001-01

[11] M. Alekhnovich, "Linear Diophantine Equations Over Polynomials and Soft Decoding of Reed-Solomon Codes," *Information Theory, IEEE Transactions on*, vol. 51, no. 7, pp. 2257–2265, 2005. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1459042

[12] P. Trifonov, "Efficient Interpolation in the Guruswami–Sudan Algorithm," *Information Theory, IEEE Transactions on*, vol. 56, no. 9, pp. 4341 –4349, Sep. 2010. [Online]. Available: http://dx.doi.org/10.1109/TIT.2010.2053901

[13] S. Sakata, "Finding a minimal polynomial vector set of a vector of

nD arrays," in *Proceedings of Applied Algebra, Algebraic Algorithms and Error–Correcting Codes (AAECC '91)*, ser. LNCS, H. F. Mattson, T. Mora, and T. R. N. Rao, Eds., vol. 539. Berlin, Germany: Springer, Oct. 1991, pp. 414–425. [Online]. Available: http://dx.doi.org/10.1007/3-540-54522-0_129

[14] ——, "On Fast Interpolation Method for Guruswami-Sudan List Decoding of One-Point Algebraic-Geometry Codes," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, ser. Lecture Notes in Computer Science, S. Boztaş and I. E. Shparlinski, Eds. Berlin, Heidelberg: Springer, October 2001, vol. 2227, ch. 18, pp. 172–181. [Online]. Available: http://dx.doi.org/10.1007/3-540-45624-4_18

[15] G. L. Feng and K. K. Tzeng, "A Generalization of the Berlekamp-Massey Algorithm for Multisequence Shift-Register Synthesis with Applications to Decoding Cyclic Codes," *Information Theory, IEEE Transactions on*, vol. 37, no. 5, pp. 1274–1287, 1991. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=133246

[16] E. R. Berlekamp, *Algebraic coding theory*. McGraw-Hill, 1968.

[17] J. Massey, "Shift-Register Synthesis and BCH Decoding," *Information Theory, IEEE Transactions on*, vol. 15, no. 1, pp. 122–127, January 2003. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1054260

[18] P. Beelen and T. Høholdt, "A Syndrome Formulation of the Interpolation Step in the Guruswami-Sudan Algorithm," in *ICMCTA*, ser. Lecture Notes in Computer Science, A. I. Barbero, Ed., vol. 5228. Springer, 2008, pp. 20–32. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87448-5_3

[19] P. Beelen and K. Brander, "Key Equations for List Decoding of Reed–Solomon Codes and How to Solve Them," *Journal of Symbolic Computation*, vol. 45, no. 7, pp. 773–786, Jul. 2010.

[20] E. R. Berlekamp and L. Welch, "Error correction of algebraic block codes," US Patent Number 4,633,470.

[21] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes (North-Holland Mathematical Library)*. North Holland, June 1988.

[22] R. Roth, *Introduction to Coding Theory*. Cambridge University Press, March 2006.

[23] H. Hasse, "Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenem Konstantenkörper bei beliebiger Charakteristik." *J. Reine Angew. Math.*, vol. 175, pp. 50–54, 1936.

[24] P. Gemmell and M. Sudan, "Highly resilient correctors for polynomials," *Information Processing Letters*, vol. 43, no. 4, pp. 169–174, Sep. 1992.

[25] T. Yaghoobian and I. F. Blake, "Two new decoding algorithms for Reed-Solomon codes," *Applicable Algebra in Engineering, Communication and Computing*, vol. 5, no. 1, pp. 23–43, January 1994. [Online]. Available: http://dx.doi.org/10.1007/BF01196623

[26] D. Dabiri and I. F. Blake, "Fast parallel algorithms for decoding Reed-Solomon codes based on remainder polynomials," *Information Theory, IEEE Transactions on*, vol. 41, no. 4, pp. 873–885, 1995. [Online]. Available: http://dx.doi.org/10.1109/18.391235

[27] J. Justesen and T. Hoholdt, *A Course in Error-Correcting Codes (EMS Textbooks in Mathematics)*. European Mathematical Society, February 2004.

[28] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes," *Information and Control*, vol. 27, no. 1, pp. 87–99, 1975.

[29] J. Dornstetter, "On the Equivalence Between Berlekamp's and Euclid's Algorithms," *Information Theory, IEEE Transactions on*, vol. 33, no. 3, pp. 428–431, 1987. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1057299

[30] A. E. Heydtmann and J. M. Jensen, "On the Equivalence of the Berlekamp-Massey and the Euclidean Algorithms for Decoding," *Information Theory, IEEE Transactions on*, vol. 46, no. 7, pp. 2614–2624, 2000. [Online]. Available: http://dx.doi.org/10.1109/18.887869

[31] 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 *AAECC*, ser. Lecture Notes in Computer Science, M. Bras-Amorós and T. Høholdt, Eds., vol. 5527. Springer, 2009, pp. 32–42. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02181-7

[32] G. Schmidt, V. R. Sidorenko, and M. Bossert, "Collaborative Decoding of Interleaved Reed-Solomon Codes and Concatenated Code Designs," *Information Theory, IEEE Transactions on*, vol. 55, no. 7, pp. 2991–3012, 2009. [Online]. Available: http://dx.doi.org/10.1109/TIT.2009.2021308

[33] R. J. Mceliece, "The Guruswami–Sudan Decoding Algorithm for Reed–Solomon Codes," *Interplanetary Network Progress Report*, vol. 153, pp. 1–60, January 2003. [Online]. Available: http://ipnpr.jpl.nasa.gov/progress_report/42-153/153F.pdf

[34] P. Beelen and T. Hoholdt, "The Decoding of Algebraic Geometry Codes," in *Series on Coding Theory and Cryptology : Advances in Algebraic Geometry Codes ; volume 5*. World Scientific, 2008, pp. 49–98.

Alexander Zeh studied electrical engineering at the University of Applied Science in Stuttgart, with the main topic automation technology. He received his Dipl.-Ing. (BA) degree in 2004. He continued his studies at Universität Stuttgart until 2008, where he received is Dipl.-Ing. in electrical engineering. He participated in the double-diploma program with Télécom ParisTech (former ENST) from 2006 to 2008 and he received also a french diploma. Currently he is a Ph.D. student at the Institute of Telecommunications and Applied Information Theory, University of Ulm, Germany and at the Computer Science Department (LIX), École Polytechnique ParisTech, Paris, France. His current research interests include coding and information theory, signal processing, telecommunications and the implementation of fast algorithms on FPGAs.

Christian Gentner studied electrical engineering at the University of Applied Science in Ravensburg, with the main topic communication technology. He received his Dipl.-Ing. (BA) degree in 2006. He continued his studies at the University of Ulm until 2009, where he received his M.Sc. in telecommunication and media technology. He is currently working towards the Ph.D. degree at the Institute of Communications and Navigation of the German Aerospace Center (DLR), Germany. His current research interests include multi-sensor navigation, propagation effects and non-line-of-sight identification and mitigation as well as the implementation of these algorithms on FPGAs.

Daniel Augot studied Mathematics and Computer Science at the University Pierre and Marie Curie in Paris. He received the Master degree in theoretical computer science in 1989. He was a Ph.D. student of Pascale Charpin and graduated in 1993. He was then hired as a researcher at INRIA–Rocquencourt. In 2009, he became a senior researcher at INRIA–Saclay-Île-de-France and École Polytechnique. His major research interests are coding theory, cryptography and their interplay.