Algebraic Chase Decoding of Elliptic Codes Through Computing the Gröbner Basis

Yunqi Wan †, Li Chen †, Fangguo Zhang ‡ §
† School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou, China
‡ School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
§ Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China
Email: wanyq5@mail2.sysu.edu.cn, chenli55@mail.sysu.edu.cn, isszhfg@mail.sysu.edu.cn

2022 IEEE International Symposium on Information Theory (ISIT) | 978-1-6654-2159-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISIT50566.2022.9834889

***Abstract*—This paper proposes two interpolation-based algebraic Chase decoding for elliptic codes. It is introduced from the perspective of computing the Gröbner basis of the interpolation module, for which two Chase interpolation approaches are utilized. They are Kötter's interpolation and the basis reduction (BR) interpolation. By identifying η unreliable symbols, 2^η decoding test-vectors are formulated, and the corresponding interpolation modules can be defined. The re-encoding further helps transform the test-vectors, facilitating the two interpolation techniques. In particular, Kötter's interpolation is performed for the common elements of the test-vectors, producing an intermediate outcome that is shared by the decoding of all test-vectors. The desired Gröbner bases w.r.t. all test-vectors can be obtained in a binary tree growing fashion, leading to a low complexity but its decoding latency cannot be contained. In contrast, the BR interpolation first performs the common computation in basis construction which is shared by all interpolation modules, and then conducts the module basis construction and reduction for all test-vectors in parallel. It results in a significantly lower decoding latency. Finally, simulation results are also presented to demonstrate the effectiveness of the proposed Chase decoding.**

***Index Terms*—Algebraic geometric codes, Chase decoding, elliptic codes, interpolation, list decoding**

I. Introduction

Algebraic-geometric (AG) codes are linear block codes derived from algebraic curves. Compared with RS codes, general AG codes are longer with a greater error-correction capability. But due to the existence of curve genus, they are not maximum distance separable (MDS) codes with a reduced error-correction efficiency. Elliptic curves have a genus of one, resulting in the codes being either MDS or almost MDS. Hence, they inherit a good tradeoff between error-correction capability and efficiency.

Guruswami and Sudan (GS) [1] proposed list decoding of RS and AG codes, which consists of interpolation and root-finding, where the former dominates the decoding complexity. It is often realized through Kötter's approach [2]. GS decoding of elliptic codes was also proposed by the authors in [3]. Kötter and Vardy [4] generalized the GS algorithm and proposed the algebraic soft decoding (ASD) for RS codes. The other interpolation approach is designed from the perspective of the module basis reduction (BR) [5] [6]. It consists of basis construction and its reduction. The latter can be realized by the Mulders-Storjohann (MS) algorithm [7], or other improved variants [8] [9]. Lee and O'Sullivan proposed the GS decoding and the ASD of Hermitian codes using the BR interpolation in [10] and [11], respectively. Recently, the GS decoding and the ASD of elliptic codes were also proposed by the authors in [12] and [13], respectively. Different from the ASD framework, Bellorado and Kavcic proposed another soft decoding algorithm, namely the low-complexity Chase (LCC) decoding for RS codes [14]. With the same test-vector formulation, LCC decoding of Hermitian codes using Kötter's interpolation was proposed by Wu *et al.* [15]. Based on the BR interpolation, LCC decoding of RS codes was proposed by Xing *et al.* [16].

This paper introduces two interpolation-based algebraic Chase decoding for elliptic codes. It is proposed from the perspective of computing the Gröbner basis of the interpolation module. It is substantiated by either Kötter's interpolation or the BR interpolation. By formulating test-vectors, the corresponding interpolation modules are defined. The re-encoding is introduced to transform these modules, reducing the interpolation complexity. We show that by fully utilizing the similarity among test-vectors, Kötter's interpolation is performed for the common interpolation points, producing an intermediate outcome that is shared by the following interpolation for the uncommon points. The uncommon element interpolation is performed in a binary tree growing fashion, delivering the Gröbner bases for all test-vectors. It exhibits a low complexity but with a decoding latency that cannot be contained. In contrast, the BR interpolation can perform the computation of all Gröbner bases in parallel. It results in a significantly lower latency. Finally, simulation results of the proposed Chase decoding are presented, demonstrating their effectiveness.

II. Background Knowledge

A. Elliptic Codes

Let $\mathbb{F}_q = \{\sigma_0, \sigma_1, \ldots, \sigma_{q-1}\}$ denote the finite field of size q. An affine elliptic curve E over $\mathbb{F}_q$ is defined by a non-singular Weierstrass equation as

$$E : Y^2 + \mathfrak{a}_1 XY + \mathfrak{a}_3 Y = X^3 + \mathfrak{a}_2 X^2 + \mathfrak{a}_4 X + \mathfrak{a}_6, \quad (1)$$

with $\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{a}_3, \mathfrak{a}_4, \mathfrak{a}_6 \in \mathbb{F}_q$. It has a genus $g = 1$. On curve E, there exist at most $q + \lfloor 2\sqrt{q} \rfloor$ affine points $P_j = (x_j, y_j)$ and a point of infinity P_∞. Let $-P_j = (x_j, y_j')$ denote the inverse of P_j. We define the following coordinate sets

$$\mathbb{A} = \{x_j \mid P_j = (x_j, y_j), \forall j\}, \quad (2)$$

$$\mathbb{B}_j = \{y_j, y_j'\}. \tag{3}$$

The coordinate ring of E is $\mathcal{R} = \mathbb{F}_q[X,Y]/\langle E\rangle$. It consists of functions in the form of $\mathfrak{h}_0(x) + \mathfrak{h}_1(x)y$, where $\mathfrak{h}_0(x)$, $\mathfrak{h}_1(x) \in \mathbb{F}_q[x]$, and x and y are the residue classes of X and Y, respectively. The quotient field of $\mathcal{R}$ is called the elliptic function field, denoted as $\mathbb{F}_q(E)$. Given $h \in \mathbb{F}_q(E)$, its order at point P is denoted as $\mathrm{v}_P(h)$. If $\mathrm{v}_P(h) > 0$, h has a zero of order $\mathrm{v}_P(h)$ at P. Otherwise, it has a pole of order $-\mathrm{v}_P(h)$ at P. For elliptic curves, $-\mathrm{v}_{P_\infty}(x) = 2$, $-\mathrm{v}_{P_\infty}(y) = 3$ and $-\mathrm{v}_{P_\infty}(x^\lambda y^\gamma) = 2\lambda + 3\gamma$.

***Definition** 1 ([17]):* Let n_P denote an integer corresponding to P, $D = \sum_{P\in E} n_P[P]$ is a divisor of E. If $h \in \mathbb{F}_q(E)$ and $h \neq 0$, the divisor of h is $\mathrm{div}(h) = \sum_{P\in E} \mathrm{v}_P(h)[P]$.

Let $\mathfrak{L}(D)$ denote the Riemann-Roch space defined by the divisor D. For $\mathfrak{L}(u[P_\infty]) = \{h \in \mathbb{F}_q(E) | \mathrm{div}(h) + u[P_\infty] \succeq 0\} \bigcup \{0\}$, there exists a pole basis consisting of $\{\phi_a = 1 \mid a = 0\} \cup \{\phi_a = x^\lambda y^\gamma \mid a = 2\lambda + 3\gamma - 1, a \in (0,u), \lambda \in \mathbb{N}, \gamma \in \{0,1\}\}$, where " $\succeq$ " indicates that the coefficients of $\mathrm{div}(h) + u[P_\infty]$ are nonnegative and $\mathbb{N}$ denotes the set of nonnegative integers. It holds that $-\mathrm{v}_{P_\infty}(\phi_a) < -\mathrm{v}_{P_\infty}(\phi_{a+1})$. For each P_j, there exists a zero basis $\{\psi_{P_j,0}, \psi_{P_j,1}, \ldots, \psi_{P_j,u-1}\}$ of $\mathfrak{L}(u[P_\infty])$ such that $\psi_{P_j,\mu}(x_j, y_j) = 0$ and $\mathrm{v}_{P_j}(\psi_{P_j,\mu}) = \mu$. Given a pole basis monomial ϕ_a, we have $\phi_a = \sum_{\mu\in\mathbb{N}} \xi_{a,P_j,\mu}\psi_{P_j,\mu}$, where $\xi_{a,P_j,\mu} \in \mathbb{F}_q$ is the corresponding coefficient between ϕ_a and $\psi_{P_j,\mu}$ [2] [18]. Hence, $\mathcal{R} = \bigcup_{u=0}^{\infty} \mathfrak{L}(u[P_\infty])$. Given $h \in \mathcal{R}$, it can be written as $h = \sum_{a\in\mathbb{N}} \zeta_a\phi_a$, where $\zeta_a \in \mathbb{F}_q$ and $-\mathrm{v}_{P_\infty}(h) = \max\{-\mathrm{v}_{P_\infty}(\phi_a) \mid \zeta_a \neq 0\}$.

Let $f = f_0\phi_0 + f_1\phi_1 + \cdots + f_{k-1}\phi_{k-1} \in \mathfrak{L}(k[P_\infty])$ denote a message polynomial, an (n,k) elliptic code is defined as

$$\mathcal{C}_E(k[P_\infty]) = \{(f(P_0), f(P_1), \ldots, f(P_{n-1})), \forall f\}, \tag{4}$$

where $\underline{c} = (c_0, c_1, \ldots, c_{n-1}) \in \mathcal{C}_E(k[P_\infty])$ is a codeword. With $g = 1$, the minimum Hamming distance of the code is lower bounded as $d \geq n - k - g + 1 = n - k$.

B. GS Decoding of Elliptic Codes

Let $\mathcal{R}[z]$ denote the polynomial ring over $\mathcal{R}$ and $\mathcal{R}[z]_l = \{Q \in \mathcal{R}[z] \mid \deg_z Q \leq l\}$. Given $\underline{r} = (r_0, r_1, \ldots, r_{n-1}) \in \mathbb{F}_q^n$ as a received word, the set of n interpolation points is

$$\mathbf{P} = \{(P_0, r_0), (P_1, r_1), \ldots, (P_{n-1}, r_{n-1})\}. \tag{5}$$

A polynomial Q in $\mathcal{R}[z]_l$ can be written as $Q = \sum_{a\in\mathbb{N}}\sum_{b\leq l} Q_{ab}\phi_a z^b$, where $Q_{ab} \in \mathbb{F}_q$. Its (μ,ν)-Hasse derivative evaluation at (P_j, r_j) is defined as [3]

$$\mathcal{D}_{\mu\nu}^{(P_j,r_j)}(Q) = \sum_{a\in\mathbb{N}}\sum_{b=\nu}^{l} Q_{ab}\binom{b}{\nu}\xi_{a,P_j,\mu}r_j^{b-\nu}. \tag{6}$$

If $Q(P_j, r_j) = 0$, Q interpolates (P_j, r_j). Furthermore, if $\mathcal{D}_{\mu\nu}^{(P_j,r_j)}(Q) = 0$, $\forall\ \mu + \nu < m$, Q interpolates the point with a zero of multiplicity m.

The GS decoding consists of *interpolation* and *root-finding*. The former determines a nonzero minimum polynomial $\mathcal{Q}(x,y,z) = \sum_{\varrho=0}^{l} \mathcal{Q}_{[\varrho]}(x,y)z^\varrho \in \mathcal{R}[z]_l$, which interpolates points of $\mathbf{P}$ with a multiplicity of m. The latter finds z-roots of $\mathcal{Q}$, which constitute the decoding output list.

For monomial $\phi_a z^b \in \mathcal{R}[z]$, its $(1,\varpi)$-weighted degree is defined as $\deg_{1,\varpi}(\phi_a z^b) = -\mathrm{v}_{P_\infty}(\phi_a) + \varpi b$. Therefore, given two distinct monomials $\phi_{a_1}z^{b_1}, \phi_{a_2}z^{b_2} \in \mathcal{R}[z]$, they can be arranged in the $(1,\varpi)$-revlex order [12]. This order also enables each polynomial in $\mathcal{R}[z]_l$ to be identified by its leading monomial (the monomial with the highest order). Hence, polynomials in $\mathcal{R}[z]_l$ can be ordered by their leading monomials. In decoding an (n,k) elliptic code, $\varpi = -\mathrm{v}_{P_\infty}(\phi_{k-1}) = k$. Therefore, the desired interpolation polynomial $\mathcal{Q}(x,y,z)$ is a nonzero minimal polynomial under the $(1,k)$-revlex order.

***Definition** 2:* The interpolation module $\mathcal{I}_{\mathbf{P},l}$ is the space of all polynomials Q over $\mathcal{R}[z]_l$. They have a zero of multiplicity m at the interpolation points of $\mathbf{P}$.

The interpolation polynomial $\mathcal{Q}$ can be founded through computing a Gröbner basis of $\mathcal{I}_{\mathbf{P},l}$. Let $\mathrm{ind}(Q) = (\gamma, b)$ denote the index of Q if the leading monomial of Q is $x^\lambda y^\gamma z^b$. The following Lemma gives a simple criterion for verifying the desired Gröbner basis.

***Lemma** 1 ([11]):* Given a basis $\{M_t(x,y,z) \mid 0 \leq t \leq 2l+1\}$ of an $\mathbb{F}_q[x]$-submodule $\mathcal{I}_{\mathbf{P},l}$. Under the $(1,\varpi)$-revlex order, if $\mathrm{ind}(M_t) \neq \mathrm{ind}(M_{t'})$, $\forall t \neq t'$, $\{M_t \mid 0 \leq t \leq 2l+1\}$ is a Gröbner basis of $\mathcal{I}_{\mathbf{P},l}$.

The desired polynomial $\mathcal{Q}$ is the minimum candidate of the above Gröbner basis, which can be computed by either Kötter's approach [3] or the BR approach [12].

III. Test-Vectors Formulation

This section introduces the test-vector formulation for the proposed LCC decoding. Re-encoding transform is applied to the test-vectors, underpinning the low-complexity interpolation.

A. Formulation

Assume that codeword $\underline{c} = (c_0, c_1, \ldots, c_{n-1})$ is transmitted through a discrete memoryless channel. Given received vector $\underline{\mathrm{r}} = (\mathrm{r}_0, \mathrm{r}_1, \ldots, \mathrm{r}_{n-1}) \in \mathbb{R}^n$, a reliability matrix $\mathbf{\Pi}$ can be obtained. Its entries $\pi_{ij} = \Pr[\mathrm{r}_j \mid c_j = \sigma_i]$ are the symbol wise channel transition probabilities, where $0 \leq i \leq q-1$ and $0 \leq j \leq n-1$. Let $i_j^{\mathrm{I}} = \mathrm{argmax}\{\pi_{ij} \mid \forall i\}$ and $i_j^{\mathrm{II}} = \mathrm{argmax}\{\pi_{ij} \mid i \neq i_j^{\mathrm{I}}\}$ denote the row indices of the largest and the second largest entries in column j of $\mathbf{\Pi}$. For c_j, the two most likely decisions are $r_j^{\mathrm{I}} = \sigma_{i_j^{\mathrm{I}}}$ and $r_j^{\mathrm{II}} = \sigma_{i_j^{\mathrm{II}}}$, respectively. The symbol wise reliability metric is defined as

$$\gamma_j = \frac{\pi_{i_j^{\mathrm{I}} j}}{\pi_{i_j^{\mathrm{II}} j}}, \tag{7}$$

where $\gamma_j \in (1, +\infty)$ [14]. The decision on c_j is more reliable if γ_j is greater, and vice versa. By sorting the above n reliability metrics in a descending order, a new symbol index sequence $j_0, j_1, \ldots, j_{n-1}$ can be yielded. Let $\Theta = \{j_0, j_1, \ldots, j_{n-\eta-1}\}$ denote the index set of the $n-\eta$ most reliable symbols. Its complementary set is $\Theta^c = \{j_{n-\eta}, j_{n-\eta+1}, \ldots, j_{n-1}\}$. Subsequently, 2^η test-vectors can be formulated as

$$\underline{r}_u = (r_{j_0}^{(u)}, r_{j_1}^{(u)}, \ldots, r_{j_{n-\eta-1}}^{(u)}, r_{j_{n-\eta}}^{(u)}, \ldots, r_{j_{n-1}}^{(u)}), \tag{8}$$

where $r_j^{(u)} = r_j^{\mathrm{I}}$, if $j \in \Theta$; and $r_j^{(u)} = r_j^{\mathrm{I}}$ or r_j^{II}, if $j \in \Theta^c$. As a result, 2^η sets of interpolation points can be formed as

$$\mathbf{P}^{(u)} = \{(P_{j_0}, r_{j_0}^{(u)}), (P_{j_1}, r_{j_1}^{(u)}), \ldots, (P_{j_{n-1}}, r_{j_{n-1}}^{(u)})\}, \quad (9)$$

where $1 \le u \le 2^\eta$.

B. Re-encoding Transform

Re-encoding further transforms $2\left\lfloor \frac{k-1}{2} \right\rfloor$ points of $\mathbf{P}^{(u)}$ to have a zero z-coordinate, reducing the interpolation complexity. Let $\mathbf{P}_\Theta^{(u)}$ and $\mathbf{P}_{\Theta^c}^{(u)}$ denote the set of interpolation points defined by Θ and Θ^c, respectively, and $\mathbf{P}^{(u)} = \mathbf{P}_\Theta^{(u)} \cup \mathbf{P}_{\Theta^c}^{(u)}$. For each set of interpolation points, they can be categorized into $\left\lfloor \frac{n}{2} \right\rfloor$ pairs, each of which share the same x-coordinate. Hence, $\left\lfloor \frac{k-1}{2} \right\rfloor$ pairs of interpolation points will be chosen for re-encoding, which are called the re-encoding points. To best reduce the interpolation complexity, the $\left\lfloor \frac{k-1}{2} \right\rfloor$ pairs of the re-encoding points should be selected from $\mathbf{P}_\Theta^{(u)}$. Therefore, $\mathbf{P}_\Theta^{(u)}$ should contain at least $\left\lfloor \frac{k-1}{2} \right\rfloor$ different x-coordinates. Let Γ denote the index set of the re-encoding points and $\Gamma^c = \{0, 1, \ldots, n-1\} \setminus \Gamma$. Further let $\mathbf{P}_\Gamma^{(u)}$ and $\mathbf{P}_{\Gamma^c}^{(u)}$ denote the set of the interpolation points defined by Γ and Γ^c, respectively, and $\mathbf{P}^{(u)} = \mathbf{P}_\Gamma^{(u)} \cup \mathbf{P}_{\Gamma^c}^{(u)}$. It should be ensured that $\Gamma \subseteq \Theta$. Without loss of generality, Γ can be written as

$$\Gamma = \{j_0, j_1, \ldots, j_{2\left\lfloor \frac{k-1}{2} \right\rfloor - 1}\}. \quad (10)$$

Since $j_{2\iota}$ and $j_{2\iota+1}$ satisfy $P_{j_{2\iota}} = -P_{j_{2\iota+1}}$, where $\iota = 0, 1, \ldots, \left\lfloor \frac{k-1}{2} \right\rfloor - 1$. Therefore, the re-encoding points are

$$\mathbf{P}_\Gamma^{(u)} = \{(P_{j_0}, r_{j_0}^{(u)}), \ldots, (P_{j_{2\left\lfloor \frac{k-1}{2} \right\rfloor - 1}}, r_{j_{2\left\lfloor \frac{k-1}{2} \right\rfloor - 1}}^{(u)})\}. \quad (11)$$

Remark 1: In order to let all test-vectors share at least $\left\lfloor \frac{k-1}{2} \right\rfloor$ pairs of common symbols, the number of unreliable symbols should satisfy $\eta \le n - 2\left\lfloor \frac{k-1}{2} \right\rfloor$. As a result, the corresponding η interpolation points will exhibit at most $\left\lfloor \frac{n}{2} \right\rfloor - \left\lfloor \frac{k-1}{2} \right\rfloor$ different x-coordinates.

Let us define

$$\mathbb{A}_\Gamma = \{x_j \mid j \in \Gamma\}, \quad (12)$$

$$\mathbb{A}_{\Gamma^c} = \mathbb{A} \setminus \mathbb{A}_\Gamma. \quad (13)$$

Based on Theorem 10 of [12], the re-encoding polynomial $\mathcal{K}_\Gamma$ can be defined as

$$\mathcal{K}_\Gamma = \sum_{j \in \Gamma} r_j^{(u)} \prod_{\alpha \in \mathbb{A}_\Gamma \setminus \{x_j\}} \frac{x - \alpha}{x_j - \alpha} \prod_{\beta \in \mathbb{B}_j \setminus \{y_j\}} \frac{y - \beta}{y_j - \beta}. \quad (14)$$

Note that it can be seen as the Lagrange interpolation polynomial, and $\mathcal{K}_\Gamma(P_j) = r_j^{(u)}$ if $j \in \Gamma$. Consequently, all test-vectors can be transformed by

$$\underline{r}_u \mapsto \underline{z}_u : z_j^{(u)} = r_j^{(u)} - \mathcal{K}_\Gamma(P_j), \forall j. \quad (15)$$

They can be written as

$$\underline{z}_u = (z_{j_0}^{(u)}, z_{j_1}^{(u)}, \ldots, z_{j_{n-1}}^{(u)}). \quad (16)$$

Therefore, if $j \in \Gamma$, $z_j^{(u)} = 0$. Among all test-vectors, at least $2\left\lfloor \frac{k-1}{2} \right\rfloor$ common positions will be zero. The corresponding set of interpolation points can be represented as

$$\mathbf{P}_u = \{(P_0, z_0^{(u)}), (P_1, z_1^{(u)}), \ldots, (P_{n-1}, z_{n-1}^{(u)})\}. \quad (17)$$

The interpolation module $\mathcal{I}_{\mathbf{P}_u}$ will be further transformed. First, the following lemma needs to be introduced.

Lemma 2: Let $Q = Q^{(0)} + Q^{(1)}z \in \mathcal{I}_{\mathbf{P}_u}$ and $\mathcal{G}_\Gamma = \prod_{\alpha \in \mathbb{A}_\Gamma} (x - \alpha)$, then $\mathcal{G}_\Gamma \mid Q^{(0)}$.

Proof: Since $Q \in \mathcal{I}_{\mathbf{P}_u}$, for $(P_j, z_j^{(u)}) \in \mathbf{P}_u$ with $j \in \Gamma$, Q can be written as $Q = Q^{(0)\prime}\Lambda_j + Q^{(1)}z$, where $\Lambda_j = x - x_j$. Therefore, for all $j \in \Gamma$, we can obtain $\mathcal{G}_\Gamma \mid Q^{(0)}$. ∎

Based on Lemma 2, let $Q^{(0)} = \mathcal{G}_\Gamma \tilde{Q}^{(0)}$, we have $Q = \tilde{Q}^{(0)}\mathcal{G}_\Gamma + Q^{(1)}z \in \mathcal{I}_{\mathbf{P}_u}$. With the mapping of

$$\Phi : z \mapsto z\mathcal{G}_\Gamma, \quad (18)$$

$Q \in \mathcal{I}_{\mathbf{P}_u}$ can be transformed into

$$Q(x, y, z\mathcal{G}_\Gamma) = \tilde{Q}^{(0)}\mathcal{G}_\Gamma + Q^{(1)}z\mathcal{G}_\Gamma = \mathcal{G}_\Gamma(\tilde{Q}^{(0)} + Q^{(1)}z). \quad (19)$$

Therefore, $Q(x, y, z\mathcal{G}_\Gamma)$ interpolates all points in

$$\mathbf{P}'_u = \{(P_j, \tilde{z}_j^{(u)}) \mid \tilde{z}_j^{(u)} = \frac{z_j^{(u)}}{\mathcal{G}_\Gamma(x_j)}, 0 \le j \le n-1\}. \quad (20)$$

Let us further partition $\mathbf{P}'_u$ into

$$\tilde{\mathbf{P}}_u = \{(P_j, \tilde{z}_j^{(u)}) \in \mathbf{P}'_u \mid j \in \Gamma^c\} \quad (21)$$

and

$$\tilde{\mathbf{P}}_u^c = \{(P_j, \tilde{z}_j^{(u)}) \in \mathbf{P}'_u \mid j \in \Gamma\}, \quad (22)$$

respectively. Therefore, $\tilde{\mathbf{P}}_u^c = \{(P_j, 0) \mid j \in \Gamma\}$. Note that $\mathcal{G}_\Gamma$ passes through all points in $\tilde{\mathbf{P}}_u^c$. Let $\tilde{Q} = \tilde{Q}^{(0)} + \tilde{Q}^{(1)}z$, where $\tilde{Q}^{(1)} = Q^{(1)}$. $\tilde{Q}$ also passes through all points in $\tilde{\mathbf{P}}_u$.

With the mapping Φ, all polynomials in $\mathcal{I}_{\mathbf{P}_u}$ can be expressed in the form of (19). Since $\mathcal{G}_\Gamma$ is uniquely determined for each decoding event, a Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$ will be determined by either Kötter's interpolation or the BR interpolation. Moreover, since $\deg_{1,k}(\mathcal{G}_\Gamma) = 2\left\lfloor \frac{k-1}{2} \right\rfloor$, the weighted degree of z is $k - 2\lfloor \frac{k-1}{2} \rfloor$. That says the desired Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$ is defined under the $(1, k - 2\lfloor \frac{k-1}{2} \rfloor)$-revlex order. What follow are two interpolation approaches for finding the desired Gröbner basis for each decoding event, in which each test-vector will be GS decoded with $m = l = 1$.

IV. Kötter's Interpolation

Regarding each $\tilde{\mathbf{P}}_u$, Kötter's interpolation [3] will be performed to determine a Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$. This interpolation process can be categorized into the common element interpolation and the uncommon element interpolation, respectively.

A. Common Element Interpolation

By (21), there are $n - 2\left\lfloor \frac{k-1}{2} \right\rfloor$ interpolation points in $\tilde{\mathbf{P}}_u$. Therefore, after interpolating these points, a Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$ will be obtained. For all $\tilde{\mathbf{P}}_u$, there are $n - \eta - 2\left\lfloor \frac{k-1}{2} \right\rfloor$ common elements. Let $\Gamma' = \Theta \setminus \Gamma$ denote the index set of those common elements, we also have $\Gamma = \Theta \setminus \Gamma'$ and $\Theta^c = \Gamma^c \setminus \Gamma'$.

$\tilde{\mathbf{P}}_u$ can be further partitioned as

$$\tilde{\mathbf{P}}_u = \tilde{\mathbf{P}}_u^{(1)} \cup \tilde{\mathbf{P}}_u^{(2)}, \tag{23}$$

where

$$\tilde{\mathbf{P}}_u^{(1)} = \{(P_j, \tilde{z}_j^{(u)}) \mid j \in \Gamma'\}, \tag{24}$$

and

$$\tilde{\mathbf{P}}_u^{(2)} = \{(P_j, \tilde{z}_j^{(u)}) \mid j \in \Theta^c\}. \tag{25}$$

Hence, by exploiting the similarity among all test-vectors, the interpolation will be performed once for the common points defined in $\tilde{\mathbf{P}}_u^{(1)}$. Its outcome will be utilized by the following interpolation for the points of $\tilde{\mathbf{P}}_u^{(2)}$.

At the beginning, a group of interpolation polynomials are initialized as

$$\mathbf{G} = \{Q_t \mid t = 0, 1, 2, 3\} = \{1,\ y,\ z,\ yz\}. \tag{26}$$

Based on (6), when $m = 1$, $(\mu, \nu) = (0, 0)$, the interpolation constraints of Q_t w.r.t. point $(P_j, z_j^{(u)})$ can be interpreted as evaluation of Q_t at the point, and denoted as Δ_t. Let $Q_t = Q_t^{(0)} + Q_t^{(1)} z$, where $Q_t^{(0)}$, $Q_t^{(1)} \in \mathcal{R}$, we have

$$\Delta_t := \mathcal{D}_{0,0}^{(P_j, z_j^{(u)})}(Q_t) = Q_t^{(0)}(P_j) + Q_t^{(1)}(P_j) z_j^{(u)}. \tag{27}$$

Therefore, for each interpolation point in $\tilde{\mathbf{P}}_u$, all polynomials in $\mathcal{G}_0$ will be tested as in (27). If $\Delta_t = 0$, Q_t interpolates the point. Otherwise, an update will be needed.

Let $\mathbf{G}_j$ denote a group of the updated polynomials after the jth interpolation constraint (also implied by the jth interpolation point in $\tilde{\mathbf{P}}_u$) is satisfied. Let

$$\mathbf{G}_{j-1}^* = \{Q_t \mid\ \Delta_t \neq 0, Q_t \in \mathbf{G}_{j-1}\}. \tag{28}$$

If $\mathbf{G}_{j-1}^* \neq \emptyset$, let $Q_{t^*} = \min \mathbf{G}_{j-1}^*$. The polynomial group $\mathbf{G}_{j-1}$ will be updated as follows. For $Q_t \in \mathbf{G}_{j-1} \setminus \mathbf{G}_{j-1}^*$, $Q_t' = Q_t$. For $Q_t \in \mathbf{G}_{j-1}^* \setminus \{Q_{t^*}\}$, $Q_t' = Q_t - \frac{\Delta_t^{(j)}}{\Delta_{t^*}^{(j)}} Q_{t^*}$. For Q_{t^*}, $Q_{t^*}' = (x - x_j) Q_{t^*}$.

Since $|\tilde{\mathbf{P}}_u^{(1)}| = n - \eta - 2\left\lfloor \frac{k-1}{2} \right\rfloor$, after interpolating the points in $\tilde{\mathbf{P}}_u^{(1)}$ as above, $\mathbf{G}_{n-\eta-2\left\lfloor \frac{k-1}{2} \right\rfloor}$ will be obtained as the interpolation outcome w.r.t. these common elements.

B. Uncommon Element Interpolation

Since each symbol in Θ^c has binary decisions, the uncommon element interpolation can be performed in a binary tree growing fashion. Elaborating from $\mathbf{G}_{n-\eta-2\left\lfloor \frac{k-1}{2} \right\rfloor}$, points in $\tilde{\mathbf{P}}_u^{(2)}$ will be interpolated one by one.

The binary tree has $\eta + 1$ layers. At layer $\eta + 1$, there are 2^η polynomial sets, which correspond to the 2^η test-vectors, respectively. The minimum polynomial $\tilde{\mathcal{Q}}_u$ will be chosen from $\mathbf{G}^{(u)}_{n-2\left\lfloor \frac{k-1}{2} \right\rfloor}$, which is a Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$. $\tilde{\mathcal{Q}}_u$ can be written as

$$\tilde{\mathcal{Q}}_u = \tilde{\mathcal{Q}}_u^{(0)} + \tilde{\mathcal{Q}}_u^{(1)} z, \tag{29}$$

where $\tilde{\mathcal{Q}}_u^{(0)}, \tilde{\mathcal{Q}}_u^{(1)} \in \mathcal{R}$. Based on the mapping Φ, the interpolation polynomial $\mathcal{Q}_u$ in $\mathcal{I}_{\mathbf{P}_u}$ can be further obtained by

$$\mathcal{Q}_u = \mathcal{G}_\Gamma \tilde{\mathcal{Q}}_u \left(x, y, \frac{z}{\mathcal{G}_\Gamma}\right) = \mathcal{G}_\Gamma \tilde{\mathcal{Q}}_u^{(0)} + \tilde{\mathcal{Q}}_u^{(1)} z, \tag{30}$$

which interpolates all the points of (17). Let f_u' denote the z-roots of $\mathcal{Q}_u$, it can be written as

$$f_u' = -\frac{\mathcal{G}_\Gamma \tilde{\mathcal{Q}}_u^{(0)}}{\tilde{\mathcal{Q}}_u^{(1)}}. \tag{31}$$

Eq. (31) can be realized by the recursive coefficient search algorithm [19] [20]. Estimation of the message polynomial f_u can be further obtained by

$$\hat{f}_u = f_u' + \mathcal{K}_\Gamma. \tag{32}$$

After the binary interpolation tree has fully expanded, the interpolation polynomials $\mathcal{Q}_u$ of all 2^η test-vectors can be obtained. Each $\mathcal{Q}_u$ yields at most one estimated message and its codeword. The estimated codeword that has the smallest Euclidean distance to the received vector $\underline{\mathrm{r}}$ will be identified. Its corresponding message will be the decoding output $\hat{f}$.

V. The BR Interpolation

The BR interpolation [12] can also be utilized to determine a Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$, forming an algebraic Chase decoding that has a remarkable latency advantage over the above LCC using Kötter's interpolation. Similarly, the BR interpolation can be categorized into the common computation (in basis construction), and the remaining uncommon computation (in basis construction and their reduction).

A. Common Computation

Let us define the following variants of $\tilde{\mathbf{P}}_u^{(1)}$ and $\tilde{\mathbf{P}}_u^{(2)}$ as

$$\tilde{\mathbf{P}}_u^{(1)'} = \{(P_j, 0) \mid j \in \Gamma'\}, \tag{33}$$

$$\tilde{\mathbf{P}}_u^{(2)'} = \{(P_j, 0) \mid j \in \Theta^c\}. \tag{34}$$

***Theorem** 3 ([12]):* $\mathcal{I}_{\tilde{\mathbf{P}}_u}$ can be generated as an $\mathbb{F}_q[x]$-module by the following basis

$$\mathcal{M}_{\tilde{\mathbf{P}}_u} = \{\tilde{M}_0^{(u)} = \mathcal{G}_{\Gamma^c}, \tilde{M}_1^{(u)} = y\mathcal{G}_{\Gamma^c}, \tilde{M}_2^{(u)} = z - \mathcal{K}_{\Gamma^c}, \tilde{M}_3^{(u)} = y(z - \mathcal{K}_{\Gamma^c})\}, \tag{35}$$

where

$$\mathcal{G}_\Gamma = \prod_{\alpha \in \mathbb{A}_\Gamma} (x - \alpha) \tag{36}$$

and

$$\mathcal{K}_{\Gamma^c} = \sum_{j \in \Gamma^c} \frac{z_j^{(u)}}{\mathcal{G}_\Gamma(x_j)} \prod_{\alpha \in \mathbb{A}_{\Gamma^c} \setminus \{x_j\}} \frac{x - \alpha}{x_j - \alpha} \prod_{\beta \in \mathbb{B}_j \setminus \{y_j\}} \frac{y - \beta}{y_j - \beta}. \tag{37}$$

For each set of transformed interpolation points $\tilde{\mathbf{P}}_u$ of (21), the corresponding basis can be constructed as in (35). However, there are at least $n - \eta - 2\left\lfloor \frac{k-1}{2} \right\rfloor$ common interpolation points among all the sets. Based on (23)-(25), in all the basis of $\mathcal{M}_{\tilde{\mathbf{P}}_u}$, $\tilde{M}_0^{(u)}$ and $\tilde{M}_1^{(u)}$ are the common candidates. Let

$$\mathcal{K}_{\Gamma^c} = \mathcal{K}_{\Gamma^c}^{(0)} + \mathcal{K}_{\Gamma^c}^{(1)}, \tag{38}$$

where

$$\mathcal{K}_{\Gamma^c}^{(0)} = \sum_{j\in\Gamma'} \frac{z_j^{(u)}}{\mathcal{G}_\Gamma(x_j)} \prod_{\alpha\in\mathbb{A}_{\Gamma^c}\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta\in\mathbb{B}_j\setminus\{y_j\}} \frac{y-\beta}{y_j-\beta}, \quad (39)$$

and

$$\mathcal{K}_{\Gamma^c}^{(1)} = \sum_{j\in\Theta^c} \frac{z_j^{(u)}}{\mathcal{G}_\Gamma(x_j)} \prod_{\alpha\in\mathbb{A}_{\Gamma^c}\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta\in\mathbb{B}_j\setminus\{y_j\}} \frac{y-\beta}{y_j-\beta}. \quad (40)$$

In $\mathcal{M}_{\tilde{\mathbf{P}}_u}$, the remaining $\tilde{M}_2^{(u)}$ and $\tilde{M}_3^{(u)}$ can be rewritten as

$$\tilde{M}_2^{(u)} = z - \mathcal{K}_{\Gamma^c}^{(0)} - \mathcal{K}_{\Gamma^c}^{(1)} \quad (41)$$

and

$$\tilde{M}_3^{(u)} = y(z - \mathcal{K}_{\Gamma^c}^{(0)} - \mathcal{K}_{\Gamma^c}^{(1)}), \quad (42)$$

respectively. Therefore, for all candidates on $\mathcal{M}_{\tilde{\mathbf{P}}_u}$, $\mathcal{K}_{\Gamma^c}^{(1)}$ is the only different element. The common part of all bases can be computed once, yielding $\mathcal{M}_{\tilde{\mathbf{P}}}^*$ as

$$\mathcal{M}_{\tilde{\mathbf{P}}}^* = \{\tilde{M}_0^{(u)*}, \tilde{M}_1^{(u)*}, \tilde{M}_2^{(u)*}, \tilde{M}_3^{(u)*}\}, \quad (43)$$

where $\tilde{M}_0^{(u)*} = \mathcal{G}_{\Gamma^c}$, $\tilde{M}_1^{(u)*} = y\mathcal{G}_{\Gamma^c}$, $\tilde{M}_2^{(u)*} = z - \mathcal{K}_{\Gamma^c}^{(0)}$, $\tilde{M}_3^{(u)*} = y(z - \mathcal{K}_{\Gamma^c}^{(0)})$. Therefore, for each $\mathcal{I}_{\tilde{\mathbf{P}}_u}$, they share the same basis $\mathcal{M}_{\tilde{\mathbf{P}}}^*$. Note that the polynomials of $\mathcal{M}_{\tilde{\mathbf{P}}}^*$ pass through the points in $\tilde{\mathbf{P}}_u^{(1)} \cup \tilde{\mathbf{P}}_u^{(2)'}$.

B. Uncommon Computation

Based on $\mathcal{M}_{\tilde{\mathbf{P}}}^*$, by performing the uncommon computation in basis construction and its reduction, a Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$ can be obtained. Based on (41) and (42), the uncommon basis construction is to compute $\mathcal{K}_{\Gamma^c}^{(1)}$. Note that for $(P_j, \tilde{z}_j^{(u)}) \in \tilde{\mathbf{P}}_u^{(2)}$, $\mathcal{K}_{\Gamma^c}^{(1)}(P_j) = \tilde{z}_j^{(u)}$, and for $(P_j, 0) \in \tilde{\mathbf{P}}_u^{(1)}$, $\mathcal{K}_{\Gamma^c}^{(1)}(P_j) = 0$. $\mathcal{K}_{\Gamma^c}^{(1)}$ can be seen as the Lagrange interpolation polynomial that passes through all points in $\tilde{\mathbf{P}}_u^{(2)} \cup \tilde{\mathbf{P}}_u^{(1)'}$. Therefore, based on (40)-(42), for each $\tilde{\mathbf{P}}_u$, its basis $\mathcal{M}_{\tilde{\mathbf{P}}_u}$ can be obtained.

The MS algorithm will further reduce $\mathcal{M}_{\tilde{\mathbf{P}}_u}$ into $\mathcal{M}'_{\tilde{\mathbf{P}}_u}$ that satisfies $\mathrm{ind}(\tilde{M}_t^{(u)}) \neq \mathrm{ind}(\tilde{M}_{t'}^{(u)})$, $\forall t \neq t'$. Based on Lemma 1, $\mathcal{M}'_{\tilde{\mathbf{P}}_u}$ is a Gröbner basis of $\mathcal{I}_{\tilde{\mathbf{P}}_u}$. The minimum polynomial $\tilde{\mathcal{Q}}_u$ will be further chosen from $\mathcal{M}'_{\tilde{\mathbf{P}}_u}$. The remaining root-finding and decoding output selection processes will be the same as that described in the end of Section IV.

VI. Simulation Results

This section shows the decoding frame error rate (FER) of the proposed Chase decoding of elliptic codes. It was obtained over the additive white Gaussian noise (AWGN) channel using BPSK. The decoding complexity and latency are also shown. They were measured as the average number of finite field multiplications and the average simulation running time in decoding a codeword.

Fig. 1 shows the proposed Chase decoding performance of the $(80, 59)$ elliptic code. Performances of the GS decoding with $m = 1$ and the ASD with $l = 2$ are shown as comparison benchmarks. The simulation results show that as η increases, the Chase decoding performance can be improved.

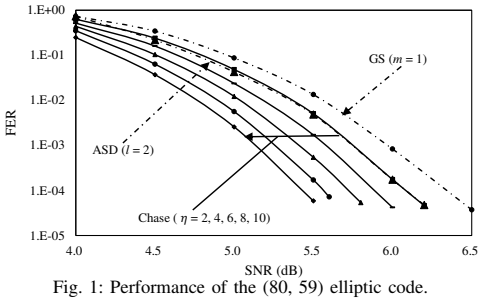

Fig. 1: Performance of the (80, 59) elliptic code.

In particular, it can be seen that for the code, the Chase decoding can significantly outperform the ASD and the GS decoding.

TABLE I: Interpolation Complexity of Chase Decoding

η	Kötter's interpolation		BR interpolation		
	Comm. Inter.	Uncomm. Inter.	Comm. Comput.	Uncomm. Comput.	Basis Red.
2	4.89×10^3	5.49×10^3	5.27×10^3	1.08×10^3	8.75×10^3
4	3.99×10^3	2.47×10^4	5.86×10^3	7.85×10^3	3.61×10^4
6	3.17×10^3	1.00×10^5	6.10×10^3	4.53×10^4	1.49×10^5

Table I shows the interpolation complexity of Chase decoding the $(80, 59)$ elliptic codes using Kötter's interpolation and the BR interpolation, respectively. It can be seen that the two interpolation approaches yield a similar complexity, while the former is slightly simpler. Despite our earlier research [12] has shown that the BR interpolation is less complex than Kötter's interpolation, under the LCC decoding paradigm, Kötter's interpolation of all test-vectors can be performed in a binary tree growing fashion, eliminating the redundant interpolation computation and resulting in a complexity advantage over the BR interpolation. However, the BR interpolation enables the uncommon computation in basis construction and reduction of all decoding events to be performed in parallel. It yields a significant advantage in decoding latency, which is demonstrated below as in Table II.

TABLE II: Decoding Latency (ms) Comparison

η	Kötter's interpolation		BR interpolation	
	(80, 39)	(80, 59)	(80, 39)	(80, 59)
2	5.270	6.679	5.040	5.850
4	9.242	11.456	5.161	5.867
6	24.970	29.870	5.201	5.881

Table II shows the latency (in ms) in decoding the $(80, 39)$ and the $(80, 59)$ elliptic codes. These results were obtained by simulating the proposed Chase decoding approaches using C programming language on the Intel core i7-10710U CPU and Windows 10. By performing the uncommon computation in basis construction and reduction in parallel, latency of the BR interpolation substantiated Chase decoding does not vary remarkably with different η values. Its latency is only defined by that of decoding a single test-vector. In contrast, latency of Kötter's interpolation substantiated Chase decoding increases with the η values.

Acknowledgement

This work is sponsored by the National Natural Science Foundation of China (NSFC) with project IDs 62071498 and 61972429, and the Guangdong Major Project of Basic and Applied Basic Research with project ID 2019B030302008.

REFERENCES

[1] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, Sep. 1999.

[2] T. Høholdt and R. Nielsen, "Decoding Hermitian codes with Sudan's algorithm," in *Pro. Int. Symp. Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, vol. 1719. Germany, Berlin:Springer-Verlag, 1999, pp. 260–269.

[3] Y. Wan, L. Chen, and F. Zhang, "Design of Guruswami-Sudan list decoding for elliptic codes," in *Proc. IEEE Inf. Theory Workshop*, Visby, Sweden, Aug. 2019, pp. 1–5.

[4] R. Kötter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.

[5] H. O'Keeffe and P. Fitzpatrick, "Gröbner basis solutions of constrained interpolation problems," *Linear algebra app.*, vol. 351, pp. 533–551, 2002.

[6] K. Lee and M. O'Sullivan, "List decoding of Reed-Solomon codes from a Gröbner basis perspective," *J. Symb. Comput.*, vol. 43, no. 9, pp. 645–658, 2008.

[7] T. Mulders and A. Storjohann, "On lattice reduction for polynomial matrices," *J. Symb. Comput.*, vol. 35, no. 4, pp. 377–401, 2003.

[8] M. Alekhnovich, "Linear diophantine equations over polynomials and soft decoding of Reed-Solomon codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2257–2265, 2005.

[9] P. Giorgi, C. P. Jeannerod, and G. Villard, "On the complexity of polynomial matrix computations," in *Proc. Int. Symp. Symb. Algebr. Comput.*, 2003, pp. 135–142.

[10] K. Lee and M. O'Sullivan, "List decoding of Hermitian codes using Gröbner bases," *J. Symb. Comput.*, vol. 44, no. 12, pp. 1662–1675, Dec. 2009.

[11] ——, "Algebraic soft-decision decoding of Hermitian codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2587–2600, Jun. 2010.

[12] Y. Wan, L. Chen, and F. Zhang, "Guruswami-Sudan decoding of elliptic codes through module basis reduction," *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 7197–7209, Nov. 2021.

[13] ——, "Algebraic soft decoding of elliptic codes," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1522–1534, Mar. 2022.

[14] J. Bellorado and A. Kavcic, "Low-complexity soft-decoding algorithms for Reed-Solomon codes–part I: An algebraic soft-in hard-out Chase decoder," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 945–959, 2010.

[15] S. Wu, L. Chen, and M. Johnston, "Interpolation-based low-complexity Chase decoding algorithms for Hermitian codes," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1376–1385, 2018.

[16] J. Xing, L. Chen, and M. Bossert, "Low-complexity Chase decoding of Reed-Solomon codes using module," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6012–6022, 2020.

[17] H. Stichtenoth, *Algebraic Function Fields and Codes*. Berlin, Germany: Springer-Verlag, 2009.

[18] L. Chen, "Design of an efficient list decoding system for Reed-Solomon and algebraic-geometric codes," Ph.D. dissertation, Dept. Electron. Comput. Eng., Newcastle Univ., Newcastle-upon-Tyne, U.K., 2008.

[19] X. Wu and P. Siegel, "Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2579–2587, Sep. 2001.

[20] L. Chen, R. Carrasco, M. Johnston, and E. Chester, "Efficient factorisation algorithm for list decoding algebraic-geometric and Reed-Solomon codes," in *Proc. IEEE Int. Conf. Commun.*, Glasgow, UK, 2007, pp. 851–856.