Algebraic Soft Decoding of Elliptic Codes

Yunqi Wan, Li Chen, *Senior Member, IEEE*, and Fangguo Zhang

***Abstract*— This paper proposes the algebraic soft decoding (ASD) for one-point elliptic codes, where the interpolation problem is solved from the perspective of module basis reduction. In ASD, the interpolation polynomial $\mathcal{Q}(x, y, z)$ is the minimum candidate of a Gröbner basis. Based on a multiplicity matrix, an interpolation ideal can be defined. With the decoding output list size, an equivalent interpolation module can be led to. By further defining the set of interpolation points, a sequence of modules from the elliptic curve coordinate ring can be obtained. Based on the Lagrange interpolation functions over elliptic function field, a basis of the interpolation module can be constructed. The desired Gröbner basis that contains $\mathcal{Q}$ can be determined by reducing the module basis. Re-encoding transform (ReT) is further introduced to reduce the basis reduction complexity. It is also shown that the interpolation can be facilitated by assessing the degree of the Lagrange interpolation polynomials. The decoding complexity is analyzed, which is verified by numerical results. That shows the advantage of this interpolation technique over the conventional Kötter's interpolation. The ASD performance of elliptic codes is also presented.**

***Index Terms*— Algebraic soft decoding, basis reduction, elliptic codes, Gröbner basis, interpolation.**

I. Introduction

ALGEBRAIC-GEOMETRIC (AG) codes were first introduced by Goppa [1]. They are linear block codes constructed based on an algebraic curve, including the popular Reed-Solomon (RS) codes, elliptic codes, Hermitian codes and etc. Their codeword length is defined by the number of rational points on the curve. As a special case, RS codes are constructed based on a straight line. Therefore, the length of an RS code cannot exceed the size of the finite field in which it is defined, limiting its minimum Hamming distance and error-correction capability. In comparison, Hermitian curves defined over the same finite field have more rational points than a straight line. Therefore, Hermitian codes are longer with a greater error-correction capability. But they are not maximum distance separable (MDS) codes due to a larger genus penalty. For an (n, k) AG code with length n and dimension k, its minimum Hamming distance is lower bounded by the designed distance d^*, where $d^* = n-k-g+1$ and g is the genus of the curve. In contrast, elliptic curves have a genus of one. They are either MDS or almost MDS codes, which show a good tradeoff between codeword length and the code's distance property.

For an (n, k) RS code, the Berlekamp-Massey (BM) algorithm [2], the Euclidean algorithm [3], and the Welch-Berlekamp algorithm [4] are the conventional unique decoding algorithms. They can correct up to $\left\lfloor \frac{n-k}{2} \right\rfloor$ errors. Extending from the Welch-Berlekamp algorithm, the hard-decision list decoding approach was proposed by Sudan for decoding low rate codes [5]. It is an interpolation-based decoding. By constructing a polynomial that passes through a given set of interpolation points with a certain multiplicity, Guruswami and Sudan [6] later improved it to decode all rate RS and AG codes, namely, the Guruswami-Sudan (GS) algorithm. It can correct errors beyond the half distance bound. By introducing a (received symbol) reliability – (interpolation point) multiplicity transform, Kötter and Vardy [7] generalized the GS algorithm and proposed the algebraic soft decoding (ASD) for RS codes. Based on the multiplicity matrix $\mathbf{M}$, the interpolation ideal over a bivariate polynomial ring can be defined. The interpolation determines a minimum polynomial $\mathcal{Q}(x, y)$ of the ideal, while the message polynomial can be retrieved from finding its y-roots, which is also called the root-finding [8], [9]. Between the interpolation and the root-finding processes, the former dominates the complexity and it can be realized by Kötter's iterative polynomial construction [10]. Lee and O'Sullivan proposed another interpolation approach for ASD of RS codes from the perspective of Gröbner bases of modules [11]. It forms a basis of the module that is defined by further including a constraint for the interpolation ideal. The basis will be further reduced, yielding a Gröbner basis that contains the desired interpolation polynomial $\mathcal{Q}$. This is called the basis reduction (BR) interpolation. In particular, the basis reduction can be realized by the Mulders-Storjohann (MS) algorithm [12], or other facilitating enhancements such as the Alekhnovich algorithm [13] or the Giorgi-Jeannerod-Villard (GJV) algorithm [14]. Moreover, the re-encoding transform (ReT) [15] and the progressive interpolation [16] can also reduce complexity for both the BR and Kötter's interpolation.

The well-known BM algorithm was generalized to multivariate domain to decode AG codes by Sakata [17]. It is called the BMS algorithm. By using the majority voting to find the unknown syndromes, Feng and Rao [18] proposed a decoding algorithm for AG codes that can correct up to $\left\lfloor \frac{d^*-1}{2} \right\rfloor$ errors. Combining the majority voting with the BMS algorithm,

Manuscript received March 30, 2021; revised September 7, 2021 and November 7, 2021; accepted January 17, 2022. Date of publication January 25, 2022; date of current version March 17, 2022. 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. An earlier version of this paper was presented in part at the 2021 IEEE International Symposium on Information Theory, Melbourne, Australia [DOI: 10.1109/ISIT45174.2021.9518148]. The associate editor coordinating the review of this article and approving it for publication was H. Mahdavifar. *(Corresponding author: Li Chen.)*

Yunqi Wan and Li Chen are with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: wanyq5@mail2.sysu.edu.cn; chenli55@mail.sysu.edu.cn).

Fangguo Zhang is with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: isszhfg@mail.sysu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCOMM.2022.3146278.

Digital Object Identifier 10.1109/TCOMM.2022.3146278

Sakata *et al.* [19] presented a more efficient decoding algorithm for AG codes, which has a complexity of $O(gn^2)$. On the other hand, Høholdt and Nielsen [20] presented a mathematical framework for GS decoding of Hermitian codes. GS decoding of elliptic codes was recently proposed by the authors in [21]. The above work in GS decoding of AG codes were realized by characterizing the function field over an algebraic curve and subsequently defining the zero basis of each affine point on the curve. Kötter's interpolation has been extended to decode these two AG codes. Lee and O'Sullivan proposed the BR interpolation for GS decoding of Hermitian codes [22]. For AG codes, the BR interpolation not only requires less decoding computation than Kötter's interpolation, but also eliminates the need for pre-computing the corresponding coefficients [21], [23]. By generalizing the Alekhnovich algorithm, Beelen and Brander reduced the interpolation complexity for a class of AG codes that are constructed from the Miura-Kamiya curves [24]. Nielsen and Beelen further presented the basis reduction technique for power decoding as well as GS decoding of Hermitian codes [25], in which the GJV algorithm is applied. The BR interpolation for GS decoding of elliptic codes was recently proposed by the authors in [26]. In the soft decoding domain, the ASD of Hermitian codes were proposed by Chen *et al.* [27] and Lee *et al.* [28], in which Kötter's interpolation and the BR interpolation were applied, respectively. The Chase decoding of Hermitian codes using Kötter's interpolation was proposed by Wu *et al.* [29].

This paper proposes the ASD of one-point elliptic codes using the BR interpolation technique. The ASD is a list decoding algorithm. The soft received information is first transformed into a multiplicity matrix $\mathbf{M}$, based on which an interpolation ideal $\mathcal{I}_{\mathbf{M}}$ can be defined. The minimum element of the ideal is the desired interpolation polynomial $\mathcal{Q}(x, y, z)$. With a predefined decoding output list size (OLS) l ($l \geq \deg_z \mathcal{Q}$), module $\mathcal{I}_{\mathbf{M},l}$ can be further defined. It contains the trivariate polynomials that satisfy the prescribed interpolation constraints with their z-degree not greater than l. In order to formulate the generators of $\mathcal{I}_{\mathbf{M},l}$, a sequence of modules from the elliptic curve coordinate ring are defined. By characterizing the zero basis of each affine point, bases of these modules can be defined. The formulation is cemented by further defining the Lagrange interpolation functions over an elliptic function field. The module basis can be represented by a matrix whose entries are $\mathbb{F}_q[x]$-coefficients of the basis polynomials. Following a $(1, k)$-weighted degree embedding mapping, the matrix can be transformed into a weighted form. The matrix can be further reduced into the desired Gröbner basis through row operations. The interpolation polynomial $\mathcal{Q}$ is the minimum candidate of the Gröbner basis with respect to the $(1, k)$-revlex order. In order to further reduce the interpolation complexity, the ReT is introduced to reduce the degree of matrix entries. This work also shows that the BR interpolation complexity can be further reduced by assessing degree of the Lagrange interpolation polynomials. The complexity of the BR interpolation will be analyzed, which will be verified by numerical results. Although the BR interpolation exhibits an asymptotic complexity of $O(l^5 n^2)$ that is the same as the conventional Kötter's interpolation, the BR interpolation requires less computation in practice. Our results will demonstrate that a substantial complexity reduction can be obtained over Kötter's interpolation. Moreover, our analysis will also show that both the BR interpolation and the ReT are more effective in yielding a low complexity for high rate code. Finally, the ASD performance of elliptic codes will be evaluated by simulations, which also demonstrate elliptic codes' performance advantage over RS codes.

II. Background Knowledge

This section presents the prerequisites of the paper, including elliptic curves, elliptic codes and the algebraic soft decoding.

A. Elliptic Curves

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

$$Y^2Z + \mathfrak{a}_1XYZ + \mathfrak{a}_3YZ^2 - X^3 - \mathfrak{a}_2X^2Z - \mathfrak{a}_4XZ^2 - \mathfrak{a}_6Z^3 = 0, \quad (1)$$

with $\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{a}_3, \mathfrak{a}_4, \mathfrak{a}_6 \in \mathbb{F}_q$. It has a genus of one. On E, there exists a point of infinity, i.e., $P_\infty = (1, 0, 1)$. With $Z = 1$, E becomes an affine curve

$$Y^2 + \mathfrak{a}_1XY + \mathfrak{a}_3Y - X^3 - \mathfrak{a}_2X^2 - \mathfrak{a}_4X - \mathfrak{a}_6 = 0. \quad (2)$$

The points on the curve are called affine points, and denoted as $P_j = (x_j, y_j)$. Let $\mathcal{P}$ denote the set of all affine points, and $E(\mathbb{F}_q)$ denote the set of $\mathbb{F}_q$-rational points on E, i.e., $E(\mathbb{F}_q) = \mathcal{P} \bigcup \{P_\infty\}$. The $\mathbb{F}_q$-rational points form an additive Abelian group based on the "chord-and-tangent" rule with P_∞ as the identity element [30]. Let δ denote the order of P_j, which is defined as the smallest nonnegative integer δ that satisfies $\delta P_j = P_\infty$. The addition rule is defined as follows.

Definition 1 [30]: Given an elliptic curve E, its points satisfy: (i) $P_j + P_\infty = P_\infty + P_j = P_j$; (ii) $P_\infty = -P_\infty$; (iii) Let $-P_j = (x_j, y_j')$ denote the inverse of P_j, $-P_j = (x_j, -y_j - \mathfrak{a}_1x_j - \mathfrak{a}_3)$ and $P_j + (-P_j) = P_\infty$; (iv) Let $P_{j_0} \neq -P_{j_1}$, and P_{j_2} be the third point of the intersection (counting multiplicities) of either the line defined by P_{j_0} and P_{j_1} (if $P_{j_0} \neq P_{j_1}$), or the tangent line to E at P_{j_0} (if $P_{j_0} = P_{j_1}$), with E. Then, $P_{j_0} + P_{j_1} = -P_{j_2}$.

Note that P_j and $-P_j$ are the only affine points on E with same x-coordinate. Therefore, for E, we can define the following x-coordinate set

$$\mathbb{A} = \{x_j \mid P_j = (x_j, y_j), \forall j\}, \quad (3)$$

where w.r.t. x_j, we can further define

$$\mathbb{B}_j = \{y_j, y_j'\}. \quad (4)$$

Let $\mathbb{F}_q[X, Y]$ denote the bivariate polynomial rings defined over $\mathbb{F}_q$, and $< E >$ denote the ideal generated by E. The coordinate ring of E is

$$\mathcal{R} = \mathbb{F}_q[X, Y]/ < Y^2 + \mathfrak{a}_1XY + \mathfrak{a}_3Y - X^3 - \mathfrak{a}_2X^2 - \mathfrak{a}_4X - \mathfrak{a}_6 > . \quad (5)$$

It is an integral domain. The quotient field of $\mathcal{R}$ is the function field of E, i.e., the elliptic function field and denoted as $\mathbb{F}_q(E)$. Let x and y denote the residue classes of X and Y, respectively. Functions of $\mathcal{R}$ are in the form $\mathfrak{h}_0(x)+\mathfrak{h}_1(x)y$, where $\mathfrak{h}_0(x)$, $\mathfrak{h}_1(x) \in \mathbb{F}_q[x]$. Given $h \in \mathbb{F}_q(E)$, its order at a rational point P is denoted as $\mathrm{v}_P(h)$ [30]. There exists a function Λ that enables $\mathrm{v}_P(\Lambda)=1$ and $h=\Lambda^{\mathrm{v}_P(h)}h'$, where $\mathrm{v}_P(h')=0$. Λ is called a local parameter in P. For the affine points of order two, $y-y_j$ is a local parameter, while $x-x_j$ is a local parameter for the others. If $\mathrm{v}_P(h)>0$, h has a zero of order $\mathrm{v}_P(h)$ at P. Otherwise, if $\mathrm{v}_P(h)<0$, 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 2 [30]: For each rational point P, define a formal symbol $[P]$. Let n_P denote an integer that corresponds to P and $D=\sum_{P\in E(\mathbb{F}_q)} n_P[P]$ denote a divisor of E with degree $\deg(D)=\sum_{P\in E(\mathbb{F}_q)} n_P$ and sum $\mathrm{sum}(D)=\sum_{P\in E(\mathbb{F}_q)} n_P P$.

Definition 3 [30]: If $h \in \mathbb{F}_q(E)$ and $h\neq 0$, the divisor of h is defined as $\mathrm{div}(h)=\sum_{P\in E(\mathbb{F}_q)} \mathrm{v}_P(h)[P]$. $\mathrm{div}(h)$ is also called the principal divisor of E.

Theorem 1 [30]: Given a divisor D of E, it is a principal divisor if and only if $\deg(D)=0$ and $\mathrm{sum}(D)=P_\infty$.

B. Elliptic Codes

Let $\mathfrak{L}(D)$ denote the Riemann-Roch space defined by a 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 basis consisting of

$$\{\phi_a=1 \mid a=0\}\cup\{\phi_a=x^\lambda y^\gamma \mid a=2\lambda+3\gamma-1, \quad a\in(0,u),\lambda\in\mathbb{N},\gamma\in\{0,1\}\} \tag{6}$$

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 $-\mathrm{v}_{P_\infty}(\phi_a)<-\mathrm{v}_{P_\infty}(\phi_{a+1})$, The above basis is called the pole basis. Consequently, $\mathcal{R}=\bigcup_{u=0}^{\infty}\mathfrak{L}(u[P_\infty])$. If $h\in\mathcal{R}$, it can be written as $h=\sum\zeta_a\phi_a$, where $\zeta_a\in\mathbb{F}_q$, and $-\mathrm{v}_{P_\infty}(h)=\max\{-\mathrm{v}_{P_\infty}(\phi_a)\mid\zeta_a\neq 0\}$. Moreover, for each affine point P_j, there exists a zero basis

$$\{\psi_{P_j,b}(x,y)\mid \mathrm{v}_{P_j}(\psi_{P_j,b})=b,\ b\in\mathbb{N}\} \tag{7}$$

of $\mathfrak{L}(u[P_\infty])$. Note that each ϕ_a can be written as

$$\phi_a=\sum_{b\in\mathbb{N}}\xi_{a,P_j,b}\psi_{P_j,b}, \tag{8}$$

where $\xi_{a,P_j,b}\in\mathbb{F}_q$ is the corresponding coefficient between ϕ_a and $\psi_{P_j,b}$ [20], [23].

Given a message vector $\underline{f}=(f_0,f_1,\ldots,f_{k-1})\in\mathbb{F}_q^k$, it can be written as

$$f(x,y)=f_0\phi_0+f_1\phi_1+\cdots+f_{k-1}\phi_{k-1}, \tag{9}$$

where $f\in\mathfrak{L}(k[P_\infty])$. The encoding of an (n,k) one-point elliptic code $\mathcal{C}_E(k[P_\infty])$ can be performed by

$$\underline{c}=(f(P_0),f(P_1),\ldots,f(P_{n-1})), \tag{10}$$

where $\underline{c}\in\mathbb{F}_q^n$. It has the minimum Hamming distance $d\geq d^*=n-k$. Note that an (n,k) elliptic code is MDS if and only if for any $\{P_{j_1},P_{j_2},\ldots,P_{j_k}\}\subseteq\mathcal{P}$, $[P_{j_1}]+[P_{j_2}]+\cdots+[P_{j_k}]-k[P_\infty]$ is not a principal divisor [31]. The above description shows that the number of affine points on curve E defines the length of the elliptic code. Over $\mathbb{F}_q$, there exists an elliptic curve E on which the number of rational points can reach the Hasse-Weil bound [32], i.e., $|E(\mathbb{F}_q)|=q+1+\lfloor 2\sqrt{q}\rfloor$. It should be pointed out that using the affine points of order two will make the module basis construction cumbersome. Given an elliptic curve, there exist at most three such points. In this work, we first find the curve that reaches the Hasse-Weil bound but does not contain affine points of order two. This can be realized by choosing the curve coefficients $\mathfrak{a}_1,\mathfrak{a}_2,\mathfrak{a}_3,\mathfrak{a}_4$ and $\mathfrak{a}_6$, appropriately. The code will then be constructed based on the curve. Excluding P_∞ for encoding, the elliptic codes will have length $n=q+\lfloor 2\sqrt{q}\rfloor$.

C. Algebraic Soft Decoding

The ASD decoding consists of the reliability transform, the interpolation and the root-finding. This subsection introduces the mathematical foundation of the ASD for elliptic codes.

Let $\mathcal{R}[z]$ denote the polynomial ring over $\mathcal{R}$. For monomial $\phi_a z^b\in\mathcal{R}[z]$, its $(1,\varpi)$-weighted degree is $\deg_{1,\varpi}(\phi_a z^b)=-\mathrm{v}_{P_\infty}(\phi_a)+\varpi b$. Given two distinct monomials $\phi_{a_1}z^{b_1}$ and $\phi_{a_2}z^{b_2}$, we can arrange them in the following $(1,\varpi)$-revlex order. That says $\mathrm{ord}(\phi_{a_1}z^{b_1})<\mathrm{ord}(\phi_{a_2}z^{b_2})$, if $\deg_{1,\varpi}(\phi_{a_1}z^{b_1})<\deg_{1,\varpi}(\phi_{a_2}z^{b_2})$, or $\deg_{1,\varpi}(\phi_{a_1}z^{b_1})=\deg_{1,\varpi}(\phi_{a_2}z^{b_2})$ and $b_1<b_2$. Hence, for a polynomial $Q=\sum_{a,b}Q_{ab}\phi_a z^b\in\mathcal{R}[z]$, its $(1,\varpi)$-weighted degree and leading order can be defined as $\deg_{1,\varpi}(Q)=\max\{\deg_{1,\varpi}(\phi_a z^b)\mid Q_{ab}\neq 0\}$ and $\mathrm{lod}(Q)=\max\{\mathrm{ord}(\phi_a z^b)\mid Q_{ab}\neq 0\}$. Given two distinct polynomials $Q_1,Q_2\in\mathcal{R}[z]$, we claim $Q_1<Q_2$ if $\mathrm{lod}(Q_1)<\mathrm{lod}(Q_2)$. In ASD of an (n,k) elliptic code, $\varpi=-\mathrm{v}_{P_\infty}(\phi_{k-1})=k$.

Definition 4: A polynomial $Q\in\mathcal{R}[z]$ has an interpolation multiplicity of m at point (P_j,σ_i) if it can be written as $\sum_{a+b\geq m}Q_{ab}\Lambda_j^a(z-\sigma_i)^b$, where Λ_j is a local parameter in P_j. The multiplicity is denoted as $\mathrm{mult}_{(P_j,\sigma_i)}(Q)$.

Definition 5: A multiplicity matrix $\mathbf{M}$ is a matrix of size $q\times n$ with entry m_{ij} representing the interpolation multiplicity for point (P_j,σ_i).

1) Reliability Transform: Suppose that a codeword $\underline{c}=(c_0,c_1,\ldots,c_{n-1})$ is transmitted through a discrete memoryless channel. Given a received vector $\underline{r}=(r_0,r_1,\ldots,r_{n-1})\in\mathbb{R}^n$, a reliability matrix $\mathbf{\Pi}$ of size $q\times n$ can be obtained. Its entry

$$\pi_{ij}=\Pr[r_j\mid c_j=\sigma_i] \tag{11}$$

is the symbol wise channel observations. In this paper, the elliptic codes are defined over finite fields of characteristics two, and binary phase-shift keying (BPSK) modulation is used for simulation over the additive white Gaussian noise (AWGN) channel. Hence, assuming the bits (binary representation) of a codeword symbol are independent, the above symbol wise reliabilities can be computed from the corresponding bit wise reliabilities.

Matrix $\mathbf{\Pi}$ will be transformed into a multiplicity matrix $\mathbf{M}$ of the same size [7]. Note that the $\mathbf{\Pi}\rightarrow\mathbf{M}$ transform

can be parameterized by a predefined decoding OLS l, where $l \geq \deg_z \mathcal{Q}$ [27]. Using the Algorithm A of [7], the $\mathbf{\Pi} \rightarrow \mathbf{M}$ transform updates the entries of $\mathbf{M}$ iteratively, so that the resulting $\mathbf{M}$ can sustain a predefined parameter l. The relationship between l and $\mathbf{M}$ will later be introduced in Theorem 3. The $\mathbf{\Pi} \rightarrow \mathbf{M}$ transform will terminate once $\mathbf{M}$ can sustain a predefined l.

2) Interpolation and Root-Finding: Given $\mathbf{M}$, we can define ideal $\mathcal{I}_{\mathbf{M}}$ as a set of all polynomials over $\mathcal{R}[z]$ that have a zero of multiplicity at least $m_{ij}(m_{ij} \neq 0)$ at the point (P_j, σ_i), which is

$$\mathcal{I}_{\mathbf{M}} = \{Q \in \mathcal{R}[z] \mid \mathrm{mult}_{(P_j,\sigma_i)}(Q) \geq m_{ij} \quad \text{for } 0 \leq i \leq q-1, 0 \leq j \leq n-1\}. \quad (12)$$

Interpolation aims to find the minimum polynomial $\mathcal{Q}$ over $\mathcal{I}_{\mathbf{M}}$.

Let $i_j = \{i \mid \sigma_i = c_j\}$, the codeword score of $\mathbf{M}$ is defined as

$$\mathrm{S_M}(\underline{c}) = \sum_{j=0}^{n-1} m_{i_j j}.$$

For a given decoding event, if the ASD can recover its message polynomial, then the decoding is claimed successful. The following Theorem reveals a sufficient condition for a successful ASD.

Theorem 2: Given an (n, k) elliptic code and the interpolation polynomial $\mathcal{Q} \in \mathcal{I}_{\mathbf{M}}$. If

$$\mathrm{S_M}(\underline{c}) > \deg_{1,k}(\mathcal{Q}), \quad (13)$$

then $\mathcal{Q}(x, y, f) = 0$, or equivalently $(z - f) \mid \mathcal{Q}$.

Proof: Since $\mathcal{Q} \in \mathcal{I}_{\mathbf{M}}$, for P_j, $\mathcal{Q}$ can be written as

$$\mathcal{Q} = \sum_{a+b_i \geq m_{ij}} h_a \prod_{i=0}^{q-1} (z - \sigma_i)^{b_i}, \quad (14)$$

where $h_a \in \mathcal{R}$ and $\mathrm{v}_{P_j}(h_a) \geq a$. Replacing z in (14) by f yields $\mathcal{Q}(x, y, f) = \sum_{a+b_i \geq m_{ij}} h_a \prod_{i=0}^{q-1} (f - \sigma_i)^{b_i}$. If $f(P_j) = \sigma_i$ for each P_j, then $\mathrm{v}_{P_j}(\mathcal{Q}(x, y, f)) \geq a + b_i \geq m_{ij}$. Hence, $\mathcal{Q}(x, y, f)$ has at least $\mathrm{S_M}(\underline{c})$ zeroes over the n affine points, i.e., $\sum_{j=0}^{n-1} \mathrm{v}_{P_j}(\mathcal{Q}(x, y, f)) = \mathrm{S_M}(\underline{c})$. Since $f \in \mathfrak{L}(k[P_\infty])$, $\deg_{1,k} \mathcal{Q}(x, y, f) \leq \deg_{1,k}(\mathcal{Q}(x, y, z))$. Based on (13), since $\deg_{1,k}(\mathcal{Q}(x, y, f)) = -\mathrm{v}_{P_\infty}(\mathcal{Q}(x, y, f))$, $\sum_{j=0}^{n-1} \mathrm{v}_{P_j}(\mathcal{Q}(x, y, f)) > -\mathrm{v}_{P_\infty}(\mathcal{Q}(x, y, f))$, i.e., $\mathcal{Q}(x, y, f)$ has a zero order that is greater than its pole order. Hence, $\mathcal{Q}(x, y, f) = 0$. ∎

Theorem 2 implies that it is essential to determine a polynomial with the minimum $(1, k)$-weighted degree over $\mathcal{I}_{\mathbf{M}}$, so that a successful ASD can be ensured. Among $Q \in \mathcal{I}_{\mathbf{M}}$ with the same weighted degree, the one with the smallest z-degree is more beneficial to reduce complexity of the root-finding. Therefore, the goal of the interpolation is to construct the minimum polynomial $\mathcal{Q}$ under the $(1, k)$-revlex order over $\mathcal{I}_{\mathbf{M}}$. Since the decoding outputs are z-roots of $\mathcal{Q}$, the designed decoding OLS $l \geq \deg_z(\mathcal{Q})$, which will be characterized as follows.

Given $\mathbf{M}$, let

$$\mathfrak{C}_{\mathbf{M}} = \sum_{i=0}^{q-1} \sum_{j=0}^{n-1} \binom{m_{ij}+1}{2} \quad (15)$$

denote the interpolation cost of the matrix. It is also the number of interpolation constraints.

Theorem 3 [33]: For an (n, k) elliptic code, given $\mathbf{M}$, the error-correction capability is determined by the decoding OLS l as

$$l = \left\lfloor \sqrt{\frac{2\mathfrak{C}_{\mathbf{M}}}{k} + \frac{1}{4}} - \frac{1}{2} \right\rfloor. \quad (16)$$

Theorem 2 shows that if $\mathrm{S_M}(\underline{c}) > \deg_{1,k}(\mathcal{Q})$, message polynomial f can be decoded by finding z-roots of $\mathcal{Q}$, which can be realized by the recursive coefficient search algorithm [8], [27]. This recursive process determines the message polynomial coefficients successively by modifying the interpolation polynomial $\mathcal{Q}$. It exhibits a complexity of $O(l^2 n^2)$. If multiple z-roots are found, the estimated message $\hat{f}$ is selected such that its corresponding codeword after modulation has the minimum Euclidean distance to the received vector $\underline{r}$.

3) Asymptotic Error-Correction Capability: As l increases, the ASD has a stronger error-correction capability. Its asymptotic error-correction capability can be characterized when l tends to infinity. Adopting the notation in [7], let $\mathcal{N}_{1,k}(\mathcal{D})$ denote the number of trivariate monomials $\phi_a z^b$ with the $(1, k)$-weighted degree at most $\mathcal{D}$, i.e.,

$$\mathcal{N}_{1,k}(\mathcal{D}) = |\{\phi_a z^b \mid \deg_{1,k}(\phi_a z^b) \leq \mathcal{D}\}|. \quad (17)$$

Further let $\Delta_{1,k}(\mathfrak{C})$ denote the minimum value of $\mathcal{D}$ so that $\mathcal{N}_{1,k}(\mathcal{D})$ is greater than $\mathfrak{C}$ as

$$\Delta_{1,k}(\mathfrak{C}) = \min\{\mathcal{D} \mid \mathcal{N}_{1,k}(\mathcal{D}) > \mathfrak{C}\}. \quad (18)$$

It can be seen that as $\mathfrak{C} \rightarrow \infty$, $\Delta_{1,k}(\mathfrak{C}) \rightarrow \infty$.

Lemma 4: $\mathcal{N}_{1,k}(\mathcal{D}) \geq \frac{\mathcal{D}^2}{2k}$.

Proof: Based on (17), $\mathcal{N}_{1,k}(\mathcal{D}) = |\{\phi_a z^0 \mid \deg_{1,k}(\phi_a z^0) \leq \mathcal{D}\}| + |\{\phi_a z^1 \mid \deg_{1,k}(\phi_a z^1) \leq \mathcal{D}\}| + \cdots + |\{\phi_a z^{\lfloor \frac{\mathcal{D}}{k} \rfloor} \mid \deg_{1,k}(\phi_a z^{\lfloor \frac{\mathcal{D}}{k} \rfloor}) \leq \mathcal{D}\}| = \sum_{i=0}^{\lfloor \frac{\mathcal{D}}{k} \rfloor} |\{\phi_a z^i \mid \deg_{1,k}(\phi_a) \leq \mathcal{D} - ik\}|$. Note that if $\mathcal{D} - ik \neq 0$, $|\{\phi_a z^i \mid \deg_{1,k}(\phi_a) \leq \mathcal{D} - ik\}| = \mathcal{D} - ik$; otherwise, $|\{\phi_a z^i \mid \deg_{1,k}(\phi_a) \leq 0\}| = 1$. Since $\mathcal{D} - k\lfloor \frac{\mathcal{D}}{k} \rfloor \geq 0$, $\mathcal{N}_{1,k}(\mathcal{D}) \geq \frac{1}{2}\left(2\mathcal{D} - k\lfloor \frac{\mathcal{D}}{k} \rfloor\right)\left(\lfloor \frac{\mathcal{D}}{k} \rfloor + 1\right) \geq \frac{\mathcal{D}}{2}\left(\lfloor \frac{\mathcal{D}}{k} \rfloor + 1\right) \geq \frac{\mathcal{D}^2}{2k}$. ∎

With the above Lemma 4, we can conclude the asymptotic error-correction capability of the ASD for elliptic codes as follows.

Theorem 5: The ASD manages to output the correct message f if

$$\sum_{j=0}^{n-1} \pi_{i_j j} > \sqrt{k \sum_{i=0}^{q-1} \sum_{j=0}^{n-1} \pi_{ij}^2}. \quad (19)$$

Proof: Based on Lemma 4 and eq. (18), we know $\Delta_{1,k}(\mathfrak{C}) > \sqrt{2k\mathcal{N}_{1,k}(\Delta_{1,k}(\mathfrak{C}))} = \sqrt{2k\mathfrak{C}}$. Therefore, eq. (13)

can be written as

$$\sum_{j=0}^{n-1} m_{ijj} > \sqrt{k \sum_{i=0}^{q-1} \sum_{j=0}^{n-1} m_{ij}(m_{ij}+1)}. \tag{20}$$

Let $\mathfrak{m} = \sum_{i=0}^{q-1} \sum_{j=0}^{n-1} m_{ij}$, we can realize that when $l \to \infty$, $\mathfrak{C} \to \infty$, $\mathfrak{m} \to \infty$, and $\frac{m_{ij}}{\mathfrak{m}} \to \frac{\pi_{ij}}{n}$ [7]. Moreover, when $\mathfrak{m} \to \infty$, $\frac{n}{\mathfrak{m}} \to 0$. Therefore, when $l \to \infty$, $m_{ij} = \frac{\mathfrak{m}}{n}\pi_{ij}$. Hence, eq. (19) can be reached. ■

The above Theorem shows that the asymptotic error-correction capability of the ASD is determined by the reliability matrix $\mathbf{\Pi}$. This is in line with the characterizations for RS codes [7] and Hermitian codes [27].

III. The Basis Reduction Interpolation

This section proposes the BR interpolation for the ASD, which is to construct the interpolation polynomial $\mathcal{Q}$ through module basis construction and its reduction. A complexity reduction approach is further introduced.

A. Module Basis and Gröbner Basis

Note that for $\mathcal{I}_{\mathbf{M}}$, its minimum polynomial $\mathcal{Q}$ will also be the minimum candidate in its Gröbner basis.[1] Therefore, $\mathcal{Q}$ can be obtained by computing the Gröbner basis of $\mathcal{I}_{\mathbf{M}}$.

Let $\mathcal{R}[z]_l = \{Q \in \mathcal{R}[z] \mid \deg_z(Q) \le l\}$ denote a free module over $\mathbb{F}_q[x]$ of rank $2(l+1)$. It has a free basis of $\{1, y, z, yz, \ldots, z^l, yz^l\}$. Let us define

$$\mathcal{I}_{\mathbf{M},l} = \mathcal{I}_{\mathbf{M}} \cap \mathcal{R}[z]_l,$$

which is a submodule of $\mathcal{R}[z]_l$ over $\mathbb{F}_q[x]$. Since the $(1,k)$-revlex order is applied in both $\mathcal{R}[z]$ and $\mathcal{R}[z]_l$, $\mathcal{Q}$ is the minimum candidate of $\mathcal{I}_{\mathbf{M},l}$ and its Gröbner basis.

Let $\mathrm{ind}(Q) = (\gamma, b)$ if the leading monomial of Q is $x^\lambda y^\gamma z^b$ under the $(1,k)$-revlex order. The following Lemma gives a simple criterion for verifying a Gröbner basis.

Lemma 6 [28]: Assume that $\{M_t(x,y,z) \mid 0 \le t \le 2l+1\}$ generates submodule $\mathcal{I}_{\mathbf{M},l}$. If under the $(1,k)$-revlex order $\mathrm{ind}(M_t) \ne \mathrm{ind}(M_{t'})$, $\forall t \ne t'$, then $\{M_t(x,y,z) \mid 0 \le t \le 2l+1\}$ is a Gröbner basis of $\mathcal{I}_{\mathbf{M},l}$.

Therefore, the interpolation polynomial $\mathcal{Q}$ can be computed by first constructing a basis for submodule $\mathcal{I}_{\mathbf{M},l}$. The basis will be further reduced into the Gröbner basis, in which $\mathcal{Q}$ is the minimum candidate.

In order to describe the basis reduction, the following definitions are needed. Consider a square matrix $\Xi \in \mathbb{F}_q[x]^{2l\times 2l}$. Let Ξ_t and $\Xi_{t,s}$ denote its row-t and the its entry of row-t column-s, respectively, the degree of Ξ_t is defined as

$$\deg(\Xi_t) = \max\{\deg(\Xi_{t,s}) \mid 0 \le s \le 2l+1\}, \tag{21}$$

the leading position of Ξ_t is

$$\mathrm{LP}(\Xi_t) = \max\{s \mid \deg(\Xi_{t,s}) = \deg(\Xi_t)\}, \tag{22}$$

and the degree of Ξ is

$$\deg(\Xi) = \max\{\deg(\Xi_t) \mid 0 \le t \le 2l+1\}. \tag{23}$$

[1]In this work, the Gröbner basis is defined under the $(1,k)$-revlex order.

Definition 6: Given a square matrix Ξ over $\mathbb{F}_q[x]$, it is in the weak Popov form if and only if $\mathrm{LP}(\Xi_t) \ne \mathrm{LP}(\Xi_{t'}), \forall t \ne t'$.

It should be pointed out that using elementary row operations, Ξ can be transformed into a matrix Ξ' which is in the weak Popov form, and without changing the row space of the matrices.

Example 1: Given the following matrix

$$\Xi = \begin{bmatrix} x+x^4 & 0 & 0 & 0 \\ 0 & x+x^4 & 0 & 0 \\ 1+x^3 & 1+x^3 & 1 & 0 \\ x^3+x^6 & 0 & 0 & 1 \end{bmatrix}$$

over $\mathbb{F}_4[x]$. Applying row operations, it can be transformed into

$$\Xi' = \begin{bmatrix} x+x^4 & 0 & 0 & 0 \\ 1+x^3 & 1+x^3 & 1 & 0 \\ 0 & 0 & x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It can be seen that the row space of Ξ and Ξ' are the same. Since $\mathrm{LP}(\Xi_0) = 0$, $\mathrm{LP}(\Xi_1) = 1$, $\mathrm{LP}(\Xi_2) = 1$ and $\mathrm{LP}(\Xi_3) = 0$, Ξ is not in weak Popov form. However, for Ξ', all its leading positions are different. It is in weak Popov form. ■

Note that a module basis can be represented by a matrix whose entries are $\mathbb{F}_q[x]$-coefficients of the basis polynomials, i.e., for each polynomial $Q(x,y,z) = Q^{(0)}(x) + Q^{(1)}(x)y + \cdots + Q^{(2l+1)}(x)yz^l \in \mathcal{R}[z]_l$, it can be represented as a vector $(Q^{(0)}(x), Q^{(1)}(x), \ldots, Q^{(2l+1)}(x))$. Performing a $(1,k)$-weighted degree embedding mapping, the matrix can be transformed into a weighted form. Based on Lemma 6 and Definition 6, when the weighted matrix is reduced into the weak Popov form, it can be demapped back into a set of polynomials which form a Gröbner basis of the interpolation module. The desired interpolation polynomial $\mathcal{Q}(x,y,z)$ is the minimum candidate of the Gröbner basis.

B. Module Basis Construction

In order to construct the basis for $\mathcal{I}_{\mathbf{M},l}$, the following interpolation point enumeration is needed. Let $\mathcal{S}_j$ denote the multiset of interpolation points (P_j, σ_i) that are defined by P_j

$$\mathcal{S}_j = \{\underbrace{(P_j,\sigma_i),\ldots,(P_j,\sigma_i)}_{m_{ij}} \mid \forall i\}. \tag{24}$$

Its balanced list $\mathcal{S}'_j$ can be further generated by moving one of the most frequent elements in $\mathcal{S}_j$ to $\mathcal{S}'_j$, until $\mathcal{S}_j$ becomes empty. Let $\mathfrak{m}_j = \sum_{i=0}^{q-1} m_{ij}$, we have $|\mathcal{S}'_j| = |\mathcal{S}_j| = \mathfrak{m}_j$. Note that $\mathfrak{m}_j \le l$. $\mathcal{S}'_j$ can be denoted as

$$\mathcal{S}'_j = \{(P_j, z_j^{(0)}), (P_j, z_j^{(1)}), \ldots, (P_j, z_j^{(\mathfrak{m}_j-1)})\}, \tag{25}$$

where $z_j^{(u)} \in \mathbb{F}_q$ and $0 \le u \le \mathfrak{m}_j - 1$. With all balance lists $\mathcal{S}'_0, \mathcal{S}'_1, \ldots, \mathcal{S}'_{n-1}$, let $\underline{z}^{(u)} = (z_0^{(u)}, z_1^{(u)}, \ldots, z_{n-1}^{(u)})$. Furthermore, $\mathcal{S}'_j$ can be partitioned into $\mathcal{S}_j^{(u)} = \{(P_j, z_j^{(0)}), \ldots, (P_j, z_j^{(u-1)})\}$ and $\overline{\mathcal{S}}_j^{(u)} = \{(P_j, z_j^{(u)}), \ldots, (P_j, z_j^{(\mathfrak{m}_j-1)})\}$. Note that in $\overline{\mathcal{S}}_j^{(u)}$, $(P_j, z_j^{(u)})$

is one of the most frequent elements. Let $\mathfrak{m}_j^{(u)}$ denote the multiplicity of $(P_j, z_j^{(u)})$ in $\overline{\mathcal{S}}_j^{(u)}$, we can define

$$\mathcal{J}_u = \{h \in \mathcal{R} \mid \mathrm{v}_{P_j}(h) \geq \mathfrak{m}_j^{(u)}\} \tag{26}$$

as an $\mathbb{F}_q[x]$-submodule of $\mathcal{R}$. Therefore, $\mathcal{J}_u$ satisfies the following property.

Lemma 7: $\mathcal{J}_u \subseteq \mathcal{J}_{u+1}$.

Proof: Let $h \in \mathcal{J}_u$, $\mathrm{v}_{P_j}(h) \geq \mathfrak{m}_j^{(u)}$ for $0 \leq j \leq n-1$. Since $\mathfrak{m}_j^{(u)} \geq \mathfrak{m}_j^{(u+1)}$, $\mathrm{v}_{P_j}(h) \geq \mathfrak{m}_j^{(u+1)}$. Based on eq. (26), $h \in \mathcal{J}_{u+1}$. Therefore, $\mathcal{J}_u \subseteq \mathcal{J}_{u+1}$. ∎

Recall that over an elliptic curve E, $P_j = (\alpha, \beta)$ and $-P_j = (\alpha, \beta')$. Let $P_{\alpha_0}^{(u)} = (\alpha, \beta)$, $P_{\alpha_1}^{(u)} = (\alpha, \beta')$, and $\mu_{\alpha_v}^{(u)} = \mathfrak{m}_j^{(u)}$ for $P_{\alpha_v}^{(u)} = P_j$, where $v = 0, 1$. For each $\alpha \in \mathbb{A}$, we arrange the index v such that

$$\mu_{\alpha_0}^{(u)} \geq \mu_{\alpha_1}^{(u)}. \tag{27}$$

Therefore, $\mathcal{J}_u$ can be written as

$$\mathcal{J}_u = \{h \in \mathcal{R} \mid \mathrm{v}_{P_{\alpha_v}^{(u)}}(h) \geq \mu_{\alpha_v}^{(u)}\}. \tag{28}$$

The following Lemma describes the property of elements in $\mathcal{J}_u$.

Lemma 8: Let $h(x,y) = \sum_{s=0}^{\rho} \mathfrak{h}_s(x) y^s \in \mathcal{J}_u$, then $\prod_{\alpha \in \mathbb{A}} (x-\alpha)^{\mu_{\alpha_\rho}^{(u)}} | \mathfrak{h}_\rho(x)$.

Proof: When $\rho = 0$, $h = \mathfrak{h}_0 \in \mathcal{J}_u$. For $P_{\alpha_0}^{(u)}$, there exists a local parameter Λ such that $\mathfrak{h}_0 = \Lambda^{\mu_{\alpha_0}^{(u)}} \mathfrak{h}_0'$. Since $P_{\alpha_0}^{(u)}$ is not an affine point of order two, Λ can be defined as $x - \alpha$. Therefore, $\prod_{\alpha \in \mathbb{A}} (x-\alpha)^{\mu_{\alpha_0}^{(u)}} | \mathfrak{h}_0$. When $\rho = 1$, for $P_{\alpha_1}^{(u)}$, let us assume that $(x-\alpha)^{\mu_{\alpha_1}^{(u)}} \nmid \mathfrak{h}_1$, h can be written as $h = (x-\alpha)^{\mu} h'$, where $\mu < \mu_{\alpha_1}^{(u)}$, $h' = \mathfrak{h}_0' + \mathfrak{h}_1' y$, and $(x-\alpha) \nmid \mathfrak{h}_0'$ or $(x-\alpha) \nmid \mathfrak{h}_1'$. Since $h \in \mathcal{J}_u$, $\mathrm{v}_{P_{\alpha_1}}(h) \geq \mu_{\alpha_1}^{(u)}$ and $h'(P_{\alpha_1}^{(u)}) = 0$. Since $h'(P_{\alpha_0}^{(u)}) \neq h'(P_{\alpha_1}^{(u)})$, $\mathrm{v}_{P_{\alpha_0}^{(u)}}(h) = \mu < \mu_{\alpha_0}^{(u)}$. It contradicts $\mathrm{v}_{P_{\alpha_0}^{(u)}}(h) = \mu_{\alpha_0}^{(u)}$. Therefore, $\prod_{\alpha \in \mathbb{A}} (x-\alpha)^{\mu_{\alpha_\rho}^{(u)}} | \mathfrak{h}_\rho$. ∎

Let $\mu_\alpha^{(u)} = \mu_{\alpha_0}^{(u)} - \mu_{\alpha_1}^{(u)}$, given $\mathcal{H} \in \mathbb{F}_q[x]$, it satisfies

$$\mathrm{v}_{P_{\alpha_0}^{(u)}}(y - \mathcal{H}(x)) \geq \mu_\alpha^{(u)}, \ \forall \alpha \in \mathbb{A}. \tag{29}$$

Let

$$\nu^{(u)} = \sum_{\alpha \in \mathbb{A}} \mu_\alpha^{(u)}, \tag{30}$$

$\mathcal{H}$ can be written as

$$\mathcal{H}(x) = \sum_{i=0}^{\nu^{(u)}-1} \zeta_i x^i, \tag{31}$$

where $\zeta_i \in \mathbb{F}_q$. Please note that the defined notion of $\mathcal{H}(x)$ is provided in Appendix. Therefore, based on the above Lemma, the basis of $\mathcal{J}_u$ can be obtained as follows.

Theorem 9: $\mathcal{J}_u$ $(0 \leq u \leq l)$ can be generated as an $\mathbb{F}_q[x]$-module by

$$\mathcal{G}_0^{(u)}(x,y) = \prod_{\alpha \in \mathbb{A}} (x-\alpha)^{\mu_{\alpha_0}^{(u)}} \tag{32}$$

and

$$\mathcal{G}_1^{(u)}(x,y) = (y - \mathcal{H}(x)) \prod_{\alpha \in \mathbb{A}} (x-\alpha)^{\mu_{\alpha_1}^{(u)}}. \tag{33}$$

Proof: For each point $P_{\alpha_v}^{(u)}$, $\mathrm{v}_{P_{\alpha_v}^{(u)}}(\mathcal{G}_0^{(u)}) \geq \mu_{\alpha_v}^{(u)}$ and $\mathrm{v}_{P_{\alpha_v}^{(u)}}(\mathcal{G}_1^{(u)}) = \mathrm{v}_{P_{\alpha_v}^{(u)}}(y - \mathcal{H}) + \mu_{\alpha_1}^{(u)} \geq \mu_{\alpha_v}^{(u)}$. Therefore, $\mathcal{G}_0^{(u)}$, $\mathcal{G}_1^{(u)} \in \mathcal{J}_u$. Based on Lemma 8, for each $h = \mathfrak{h}_0 + \mathfrak{h}_1 y \in \mathcal{J}_u$, there exists $\mathfrak{h}_1'$ such that $h' = h - \mathfrak{h}_1' \mathcal{G}_1^{(u)} \in \mathcal{J}_u$. Therefore, there also exists $\mathfrak{h}_0'$ such that $h' = \mathfrak{h}_0' \mathcal{G}_0^{(u)}$. As a result, $h = \mathfrak{h}_1' \mathcal{G}_1^{(u)} + \mathfrak{h}_0' \mathcal{G}_0^{(u)}$, i.e., $\mathcal{J}_u$ can be generated as an $\mathbb{F}_q[x]$-module by $\mathcal{G}_0^{(u)}$ and $\mathcal{G}_1^{(u)}$. ∎

Note that if $\mathfrak{m}_j^{(l)} = 0$ with $0 \leq j < n$, $\mathcal{G}_0^{(l)} = 1$ and $\mathcal{G}_1^{(l)} = y$. Given $Q \in \mathcal{R}[z]$, it can be written as $Q = \sum_{s \in \mathbb{N}} Q_{[s]} z^s$, where $Q_{[s]} \in \mathcal{R}$. The following Lemma reveals the property of the polynomials in $\mathcal{I}_{\mathbf{M},l}$.

Lemma 10: Let $Q(x,y,z) = \sum_{s=0}^{\varrho} Q_{[s]}(x,y) z^s \in \mathcal{I}_{\mathbf{M},l}$, then $Q_{[\varrho]}(x,y) \in \mathcal{J}_\varrho$.

Proof: Since $Q \in \mathcal{I}_{\mathbf{M},l}$, for affine point P_j, Q can be written as

$$Q = \sum_{a + b_i \geq m_{ij}} h_a \prod_{i=0}^{q-1} (z - \sigma_i)^{b_i},$$

where $h_a \in \mathcal{R}$, $\mathrm{v}_{P_j}(h_a) \geq a$ and $\sum_{i=0}^{q-1} b_i \leq \varrho$. Therefore, $\mathrm{v}_{P_j}(h_a) \geq \max\{m_{ij} - b_i \mid 0 \leq i \leq q-1\}$. When $\varrho = 0$, $\mathrm{v}_{P_j}(h_a) \geq \max\{m_{ij}\}$ and $Q_{[0]} \in \mathcal{J}_0$. When $\varrho = 1$, for $\sum_{i=0}^{q-1} b_i = 1$, let $b_0 = 1$, we have $\mathrm{v}_{P_j}(h_a) \geq \max\{m'_{ij} \mid m'_{0j} = m_{0j} - 1 \text{ and } m'_{ij} = m_{ij}, 1 \leq i \leq q-1\}$. If $\mathfrak{m}_j^{(1)} = \mathfrak{m}_j^{(0)}$, $Q_{[1]} \in \mathcal{J}_0$ and $\mathcal{J}_0 \subseteq \mathcal{J}_1$. Otherwise, $Q_{[1]} \in \mathcal{J}_1$. Following the same deduction manner, the conclusion can be reached. ∎

To define a basis for $\mathcal{I}_{\mathbf{M},l}$, the following function is needed

$$\mathcal{K}_{\underline{z}^{(u)}}(x,y) = \sum_{j=0}^{n-1} z_j^{(u)} \mathcal{L}_j(x,y), \tag{34}$$

where

$$\mathcal{L}_j(x,y) = \prod_{\alpha \in \mathbb{A} \setminus \{x_j\}} \frac{x-\alpha}{x_j - \alpha} \prod_{\beta \in \mathbb{B}_j \setminus \{y_j\}} \frac{y-\beta}{y_j - \beta}, \tag{35}$$

which is the Lagrange interpolation function over $\mathbb{F}_q(E)$. Note that if $j = j'$, then $\mathcal{L}_j(P_{j'}) = 1$; otherwise, $\mathcal{L}_j(P_{j'}) = 0$. Hence, $\mathcal{K}_{\underline{z}^{(u)}}(P_j) = z_j^{(u)}$ for $0 \leq u < \mathfrak{m}_j$. Based on eqs. (32)-(34), the generators of $\mathcal{I}_{\mathbf{M},l}$ can be defined as follows.

Theorem 11: $\mathcal{I}_{\mathbf{M},l}$ can be generated as an $\mathbb{F}_q[x]$-module by $\mathcal{M}$

$$= \{M_t(x,y,z) \mid M_t(x,y,z) = \mathcal{G}_v^{(u)}(x,y) \prod_{\epsilon=0}^{u-1} (z - \mathcal{K}_{\underline{z}^{(\epsilon)}}(x,y)), \ t = v + 2u, v = 0, 1 \text{ and } 0 \leq u \leq l\}. \tag{36}$$

Proof: Based on Theorem 9, we have $\mathcal{G}_v^{(u)} \in \mathcal{J}_u$, i.e., $\mathrm{v}_{P_j}(\mathcal{G}_v^{(u)}) \geq \mathfrak{m}_j^{(u)}$. Let $m_{ij}^{(u)}$ denote the multiplicity of (P_j, σ_i) in $\overline{\mathcal{S}}_j^{(u)}$. Therefore, $\mathrm{mult}_{(P_j,\sigma_i)}(\mathcal{G}_v^{(u)}) \geq m_{ij}^{(u)}$. Based on eqs. (25) and (33), $\mathrm{mult}_{(P_j,\sigma_i)}(\prod_{\epsilon=0}^{u-1}(z - \mathcal{K}_{\underline{z}^{(\epsilon)}})) \geq m_{ij} - m_{ij}^{(u)}$. Therefore, $M_t \in \mathcal{I}_{\mathbf{M},l}$. Based on Lemma 10, for $Q = \sum_{s=0}^{l} Q_{[s]} z^s \in \mathcal{I}_{\mathbf{M},l}$, $Q_{[s]} \in \mathcal{J}_s$. Therefore, there exist $\mathfrak{h}_0^{(l)}$,

$\mathfrak{h}_1^{(l)} \in \mathbb{F}_q[x]$ such that $Q_{[l]} = \mathfrak{h}_0^{(l)}\mathcal{G}_0^{(l)} + \mathfrak{h}_1^{(l)}\mathcal{G}_1^{(l)}$. It enables $Q^{(l-1)} = Q - (\mathfrak{h}_0^{(l)}M_{2l} + \mathfrak{h}_1^{(l)}M_{2l+1})$ with $\deg_z Q^{(l-1)} \leq l-1$ and $Q_{[l-1]}^{(l-1)} \in \mathcal{J}_{l-1}$. Again, there exist $\mathfrak{h}_0^{(l-1)}$, $\mathfrak{h}_1^{(l-1)} \in \mathbb{F}_q[x]$ such that $Q^{(l-2)} = Q^{(l-1)} - (\mathfrak{h}_0^{(l-1)}M_{2l-2} + \mathfrak{h}_1^{(l-1)}M_{2l-1})$ with $\deg_z Q^{(l-2)} \leq l-2$ and $Q_{[l-2]}^{(l-2)} \in \mathcal{J}_{l-2}$. Following the same deduction, there exist $\mathfrak{h}_0^{(1)}$, $\mathfrak{h}_1^{(1)} \in \mathbb{F}_q[x]$ which enable $Q^{(0)} = Q^{(1)} - (\mathfrak{h}_0^{(1)}M_2 + \mathfrak{h}_1^{(1)}M_3)$. Therefore, $Q^{(0)} \in \mathcal{J}_0$, i.e., there exist $\mathfrak{h}_0^{(0)}$, $\mathfrak{h}_1^{(0)} \in \mathbb{F}_q[x]$ such that $Q^{(0)} = \mathfrak{h}_0^{(0)}M_0 + \mathfrak{h}_1^{(0)}M_1$. Consequently, if $Q \in \mathcal{I}_{\mathbf{M},l}$, it can be expressed as an $\mathbb{F}_q[x]$-linear combination of M_t. ■

It can be seen that for M_t, $\mathcal{G}_v^{(u)}$ and $\prod_{\epsilon=0}^{u-1}(z - \mathcal{K}_{\underline{z}^{(\epsilon)}})$ interpolate all points of $\overline{\mathcal{S}}_j^{(u)}$ and $\mathcal{S}_j^{(u)}$, respectively. Therefore, with $\mathcal{H}$, the generators for the basis $\mathcal{M}$ of the $\mathbb{F}_q[x]$-module $\mathcal{I}_{\mathbf{M},l}$ can be computed.

C. Module Basis Reduction

Basis $\mathcal{M}$ will be further reduced, yielding the Gröbner basis that is denoted by $\mathcal{M}'$. It contains the desired interpolation polynomial $\mathcal{Q}$.

Since $\mathcal{I}_{\mathbf{M},l}$ is a submodule of $\mathcal{R}[z]_l$ over $\mathbb{F}_q[x]$, for a polynomial $Q \in \mathcal{I}_{\mathbf{M},l}$, it can be written as $Q = Q^{(0)} + Q^{(1)}y + \cdots + Q^{(2l+1)}yz^l$, where $Q^{(0)}, Q^{(1)}, \ldots, Q^{(2l+1)} \in \mathbb{F}_q[x]$, or alternatively as $Q = (Q^{(0)}, Q^{(1)}, \ldots, Q^{(2l+1)})(1, y, \ldots, yz^l)^T$. Similarly, the basis polynomials M_t can also be written as $M_t = (M_t^{(0)}, M_t^{(1)}, \cdots, M_t^{(2l+1)})(1, y, \ldots, yz^l)^T$. Basis $\mathcal{M}$ can be presented as a matrix $\mathbf{V} \in \mathbb{F}_q[x]^{2(l+1)\times 2(l+1)}$ by letting

$$\mathbf{V}_t = (M_t^{(0)}, M_t^{(1)}, \ldots, M_t^{(2l+1)}), \tag{37}$$

where $\mathbf{V}_{t,s} = M_t^{(s)}(x)$. Inversely,

$$M_t = \mathbf{V}_t \cdot (1, y, \ldots, yz^l)^T. \tag{38}$$

The MS algorithm [12] can reduce $\mathbf{V}$ into the weak Popov form. First, let us define the mapping $\Psi_{\underline{w}}$ [25]: $\mathbb{F}_q[x]^{2(l+1)} \rightarrow \mathbb{F}_q[x]^{2(l+1)}$

$$\Psi_{\underline{w}} : \mathbf{V}_t \mapsto \mathbf{V}_t^* = \mathbf{V}_t \cdot \mathrm{diag}(x^{w_0}, x^{w_1}, \ldots, x^{w_{2l+1}}), \tag{39}$$

where $\underline{w} = (w_0, w_1, \ldots, w_{2l+1})$ and $w_s = \left\lfloor \frac{k\lfloor \frac{s}{2} \rfloor + 3(s \bmod 2)}{2} \right\rfloor$. With the mapping, matrix $\mathbf{V}$ is transformed into

$$\mathbf{V}^* = \Psi_{\underline{w}}(\mathbf{V}) = (\Psi_{\underline{w}}(\mathbf{V}_0), \Psi_{\underline{w}}(\mathbf{V}_1), \ldots, \Psi_{\underline{w}}(\mathbf{V}_{2l+1}))^T, \tag{40}$$

where $\mathbf{V}_{t,s}^* = \mathbf{V}_{t,s}x^{w_s}$. Row operations will then be performed on $\mathbf{V}^*$ to reduce it into the weak Popov form that was defined by Definition 6 earlier. It is denoted as $\mathbf{V}^{*\prime}$. The corresponding matrix $\mathbf{V}'$ can be obtained by $\Psi_{\underline{w}}^{-1}$ as

$$\mathbf{V}_t^{*\prime} \mapsto \mathbf{V}_t' = \mathbf{V}_t^{*\prime} \cdot \mathrm{diag}(x^{-w_0}, x^{-w_1}, \ldots, x^{-w_{2l+1}}). \tag{41}$$

Lemma 12 [34]: Assume that $\mathcal{M} = \{M_t(x, y, z) \mid 0 \leq t \leq 2l+1\}$ generates an $\mathbb{F}_q[x]$-module $\mathcal{I}_{\mathbf{M},l}$ and $\mathbf{V}$ is its matrix representation as in (37). If $\Psi_{\underline{w}}(\mathbf{V})$ is in the weak Popov form, then $\mathcal{M}$ is a Gröbner basis of $\mathcal{I}_{\mathbf{M},l}$ with respect to the $(1, k)$-revlex order.

Therefore, the desired Gröbner basis $\mathcal{M}'$ can be further obtained as in (38). The interpolation polynomial $\mathcal{Q}$ can be found in $\mathcal{M}'$.

Example 2: Continue Example 1. Assume that $\mathcal{I}_{\mathbf{M},l} =< M_0, M_1, M_2, M_3 >$, where $M_0 = x + x^4$, $M_1 = (x + x^4)y$, $M_2 = (1 + x^3) + (1 + x^3)y + z$ and $M_3 = (x^3 + x^6) + yz$. Let $k = 4$, then $\underline{w} = (0, 1, 2, 3)$. They can be represented in a matrix $\mathbf{V}$ as eq. (37), and further transformed into $\mathbf{V}^* = \Psi_{\underline{w}}(\mathbf{V}) = \mathbf{V} \cdot \mathrm{diag}(x^0, x^1, x^2, x^3)$ as

$$\mathbf{V}^* = \begin{bmatrix} x + x^4 & 0 & 0 & 0 \\ 0 & x^2 + x^5 & 0 & 0 \\ 1 + x^3 & x + x^4 & x^2 & 0 \\ x^3 + x^6 & 0 & 0 & x^3 \end{bmatrix}.$$

Applying row operations on $\mathbf{V}^*$, it can be transformed into

$$\mathbf{V}' = \begin{bmatrix} x + x^4 & 0 & 0 & 0 \\ 1 + x^3 & x + x^4 & x^2 & 0 \\ 0 & 0 & x^3 & 0 \\ 0 & 0 & 0 & x^3 \end{bmatrix}$$

It can be seen that $\mathbf{V}'$ is in weak Popov form. By performing $\Psi_{\underline{w}}^{-1}(\mathbf{V}') = \mathbf{V}' \cdot \mathrm{diag}(x^0, x^{-1}, x^{-2}, x^{-3})$, a Gröbner basis of $\mathcal{I}_{\mathbf{M},l}$ is $M_0' = x + x^4$, $M_1' = (1 + x^3) + (1 + x^3)y + z$, $M_2' = xz$ and $M_3' = yz$. ■

Based on Theorem 2, message polynomial f can be further decoded by finding z-roots of $\mathcal{Q}$ [8]. Summarizing the Section, the ASD algorithm that utilizes the BR interpolation can be presented as in Algorithm 1, where $\hat{f}$ denotes the estimation of f.

Algorithm 1 The ASD Algorithm

Input: $\mathbf{\Pi}$ and l;
Output: $\hat{f}$;

1: Compute $\mathbf{M}$ that sustains l;
2: Create balanced lists $\mathcal{S}_j'$ as in (24) and (25);
3: Formulate the module basis $\mathcal{M}$ as in (36);
4: Map $\mathcal{M}$ to $\mathbf{V}^*$ as in (37) and (39);
5: Reduce $\mathbf{V}^*$ using the MS algorithm, yielding its weak Popov form $\mathbf{V}^{*\prime}$;
6: Demap $\mathbf{V}^{*\prime}$ as in (38) and (41), yielding the Göbner basis $\mathcal{M}'$;
7: Choose the minimum candidate of $\mathcal{M}'$ as $\mathcal{Q}$;
8: Determine the z-roots of $\mathcal{Q}$ in estimating $\hat{f}$.

D. Reducing Interpolation Complexity

Complexity of the above ASD algorithm can be reduced by assessing the degree of $\mathcal{K}_{\underline{z}^{(u)}}$ [35]. Recalling (34), it can be written as

$$\begin{aligned} \mathcal{K}_{\underline{z}^{(u)}} &= (z_0^{(u)}, z_1^{(u)}, \ldots, z_{n-1}^{(u)})(\mathcal{L}_0, \mathcal{L}_1, \ldots, \mathcal{L}_{n-1})^T \\ &= (z_0^{(u)}, z_1^{(u)}, \ldots, z_{n-1}^{(u)}) \cdot \Upsilon \cdot (\phi_0, \phi_1, \ldots, \phi_{n-2}, \phi_n)^T, \end{aligned} \tag{42}$$

where $\Upsilon \in \mathbb{F}_q^{n\times n}$ is the coefficient matrix of the Lagrange interpolation polynomials.

Lemma 13: Given $\mathcal{K}_{\underline{z}^{(u)}}(x,y)$ that is defined as in (34), $\underline{z}^{(u)}$ is a codeword if and only if $\deg_{1,k}\left(\mathcal{K}_{\underline{z}^{(u)}}(x,y)\right) \leq k$.

Proof: If $\deg_{1,k}(\mathcal{K}_{\underline{z}^{(u)}}) \leq k$, $\mathcal{K}_{\underline{z}^{(u)}} \in \mathfrak{L}(k[P_\infty])$ and $(\mathcal{K}_{\underline{z}^{(u)}}(P_0), \mathcal{K}_{\underline{z}^{(u)}}(P_1), \ldots, \mathcal{K}_{\underline{z}^{(u)}}(P_{n-1})) \in \mathcal{C}_E(k[P_\infty])$, i.e., $\underline{z}^{(u)}$ is a codeword.

If $\underline{z}^{(u)}$ is a codeword, there exists a corresponding message polynomial $f \in \mathfrak{L}(k[P_\infty])$. To prove its uniqueness, let

$$\mathcal{K}'_{\underline{z}^{(u)}} = \mathcal{K}_{\underline{z}^{(u)}} - f.$$

Based on eq. (42), $\mathcal{K}'_{\underline{z}^{(u)}} = \zeta_0\phi_n + \zeta_1\phi_{n-2} + \cdots + \zeta_{n-1}\phi_0$, where $\zeta_i \in \mathbb{F}_q$. Therefore, $\deg_{1,k}(\mathcal{K}'_{\underline{z}^{(u)}}) \leq n+1$. According to Theorem 1, if $\deg_{1,k}(\mathcal{K}'_{\underline{z}^{(u)}}) = n+1$, $\mathrm{div}(\mathcal{K}'_{\underline{z}^{(u)}}) = [P'] + \sum_{\alpha\in\mathbb{A}}([P^{(u)}_{\alpha_0}] + [P^{(u)}_{\alpha_1}]) - (n+1)[P_\infty]$. Since $P^{(u)}_{\alpha_0} = -P^{(u)}_{\alpha_1}$, $P' = P_\infty$. Since $\mathcal{K}'_{\underline{z}^{(u)}}$ does not contain monomial ϕ_{n-1}, $\deg_{1,k}(\mathcal{K}'_{\underline{z}^{(u)}}) \leq n-1$, i.e., $\zeta_0 = 0$. Since $\mathcal{K}'_{\underline{z}^{(u)}}(P_i) = 0$, $0 \leq i < n$, $\mathcal{K}'_{\underline{z}^{(u)}}$ has a zero order that is greater than its pole order. As a result, $\mathcal{K}'_{\underline{z}^{(u)}} = 0$, i.e., $\mathcal{K}_{\underline{z}^{(u)}} = f$ and $\deg_{1,k}(\mathcal{K}_{\underline{z}^{(u)}}) \leq k$. ∎

Let us present Υ as $\Upsilon = [\Upsilon_0\Upsilon_1]$, where $\Upsilon_0 \in \mathbb{F}_q^{n\times k}$ and $\Upsilon_1 \in \mathbb{F}_q^{n\times(n-k)}$. Based on Lemma 13, for elliptic codes, Υ_1^T is the parity-check matrix. Consequently, in ASD of elliptic codes, if $\deg_{1,k}(\mathcal{K}_{\underline{z}^{(u)}}) \leq k$, then $\underline{z}^{(u)}[\Upsilon_1] = \underline{0}$ and $\mathcal{K}_{\underline{z}^{(u)}}$ is a message candidate. Note that $\underline{0}$ denotes an all zero vector. The maximum likelihood (ML) criterion [36] can be further applied to assess whether $\underline{z}^{(u)}$ is an ML codeword. If so, the decoding can be terminated and outputs $\mathcal{K}_{\underline{z}^{(u)}}$ as a message candidate. Consequently, the following basis construction and reduction can be skipped.

IV. The Re-Encoding Transformed Basis Reduction Interpolation

In the above BR interpolation, the complexity is dominated by the basis reduction process that reduces $\mathbf{V}^*$ into the weak Popov form $\mathbf{V}^{*\prime}$. ReT on the interpolation points can help reduce entry degree of $\mathbf{V}^*$, facilitating the BR interpolation [37].

A. Interpolation Points Transform

Re-encoding can transform the interpolation points, resulting in the candidates of submodule $\mathcal{I}_{\mathbf{M},l}$ share a common divisor. It can be removed so that complexity of the following basis reduction can be reduced. For elliptic curves, $|\mathbb{A}| = \left\lfloor \frac{n}{2} \right\rfloor$, where $\mathbb{A}$ was defined in eq. (3). There are $\left\lfloor \frac{k-1}{2} \right\rfloor$ pairs of interpolation points, each of which share the same x-coordinate. They are chosen for re-encoding, which are called the re-encoding points. In order to reduce the BR complexity in the best capacity, the re-encoding points should correspond to a large multiplicity. Based on the creation of balanced lists, we know $(P_j, z_j^{(0)})$ has the largest multiplicity in all interpolation points defined by P_j. Therefore, the $\left\lfloor \frac{k-1}{2} \right\rfloor$ pairs of re-encoding points should be chosen from the set

$$\{(P_0, z_0^{(0)}), (P_1, z_1^{(0)}), \ldots, (P_{n-1}, z_{n-1}^{(0)})\}. \tag{43}$$

Recall Theorem 9, $\mathcal{G}_0^{(u)} = \prod_{\alpha\in\mathbb{A}}(x-\alpha)^{\mu^{(u)}_{\alpha_0}}$ and $\mathcal{G}_1^{(u)} = (y-\mathcal{H})\prod_{\alpha\in\mathbb{A}}(x-\alpha)^{\mu^{(u)}_{\alpha_1}}$. Since $\mu^{(u)}_{\alpha_0} \geq \mu^{(u)}_{\alpha_1}$, $\prod_{\alpha\in\mathbb{A}}(x-\alpha)^{\mu^{(u)}_{\alpha_1}}$ is the greatest common divisor of $\mathcal{G}_0^{(u)}$ and $\mathcal{G}_1^{(u)}$. Therefore, for $\alpha \in \mathbb{A}$, $\mu^{(0)}_{\alpha_1}$ are sorted in a descending order and the $\left\lfloor \frac{k-1}{2} \right\rfloor$ largest values are identified. Correspondingly, the $\left\lfloor \frac{k-1}{2} \right\rfloor$ pairs of interpolation points are chosen as the re-encoding points. Let Γ denote the index set of the re-encoding points and $\bar{\Gamma} = \{0,1,\ldots,n-1\} \setminus \Gamma$. Hence, Γ can be written as

$$\Gamma = \{j_1, j'_1, \ldots, j_{\lfloor\frac{k-1}{2}\rfloor}, j'_{\lfloor\frac{k-1}{2}\rfloor}\}, \tag{44}$$

where j_i and j'_i $(i = 1,2,\ldots,\lfloor\frac{k-1}{2}\rfloor)$ satisfy $P_{j'_i} = -P_{j_i}$. Therefore, the re-encoding points are

$$(P_{j_1}, z^{(0)}_{j_1}), (P_{j'_1}, z^{(0)}_{j'_1}), \ldots, (P_{j'_{\lfloor\frac{k-1}{2}\rfloor}}, z^{(0)}_{j'_{\lfloor\frac{k-1}{2}\rfloor}}). \tag{45}$$

Furthermore, let $\mathbb{A}_\Gamma$ denote the set of x-coordinates of the re-encoding points. The re-encoding polynomial $\mathcal{K}_\Gamma$ can be defined as

$$\mathcal{K}_\Gamma(x,y) = \sum_{j\in\Gamma} z_j^{(0)} \mathcal{L}_\Gamma^{(j)}(x,y), \tag{46}$$

where

$$\mathcal{L}_\Gamma^{(j)}(x,y) = \prod_{\alpha\in\mathbb{A}_\Gamma\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta\in\mathbb{B}_j\setminus\{y_j\}} \frac{y-\beta}{y_j-\beta}. \tag{47}$$

Note that for $j \in \Gamma$, since $\mathcal{L}_\Gamma^{(j)}(P_j) = 1$, $\mathcal{K}_\Gamma(P_j) = z_j^{(0)}$. Consequently, z-coordinate of all interpolation points in the balanced lists can be transformed as

$$\tilde{z}_j^{(u)} = z_j^{(u)} - \mathcal{K}_\Gamma(P_j). \tag{48}$$

As a result, a balanced list $\mathcal{S}'_j$ will become

$$\tilde{\mathcal{S}}'_j = \{(P_j, \tilde{z}_j^{(u)}) \mid 0 \leq u < \mathfrak{m}_j\}. \tag{49}$$

B. Basis Construction

Let us define $\Gamma_u = \{j_i \mid z^{(u)}_{j_i} = z^{(0)}_{j_i} \text{ and } z^{(u)}_{j'_i} = z^{(0)}_{j'_i}, j_i \in \Gamma\}$ and $\bar{\Gamma}_u = \{0,1,\ldots,n-1\} \setminus \Gamma_u$. Therefore, $\tilde{z}_j^{(u)} = 0$ if $j \in \Gamma_u$. Accordingly, let $\mathbb{A}_{\Gamma_u}$ and $\mathbb{A}_{\bar{\Gamma}_u}$ denote the set of x-coordinates in Γ_u and $\bar{\Gamma}_u$, respectively. To describe the basis construction in the case of ReT, the following notations are needed. Let us further define

$$\mathcal{G}_\Gamma(x) = \prod_{\alpha\in\mathbb{A}_\Gamma} (x-\alpha). \tag{50}$$

It can be factorized into $\mathcal{G}_\Gamma(x) = \mathcal{G}_{\Gamma_u}(x)\mathcal{G}_{\Gamma\setminus\Gamma_u}(x)$, where

$$\mathcal{G}_{\Gamma_u}(x) = \prod_{\alpha\in\mathbb{A}_{\Gamma_u}} (x-\alpha), \tag{51}$$

and

$$\mathcal{G}_{\Gamma\setminus\Gamma_u}(x) = \prod_{\alpha\in\mathbb{A}_\Gamma\setminus\mathbb{A}_{\Gamma_u}} (x-\alpha). \tag{52}$$

Based on the above transform, the new multiplicity matrix $\mathbf{M}'$ is obtained with entries m'_{ij}. Therefore, $m'_{ij} = m_{i'j}$, where $\sigma_i + \mathcal{K}_\Gamma(P_j) = \sigma_{i'}$. Let

$$\mathcal{I}_{\mathbf{M}',l} = \{Q \in \mathcal{R}[z]_l \mid \text{mult}_{(P_j,\sigma_i)}(Q) \geq m'_{ij} \quad \text{for } 0 \leq i \leq q-1, 0 \leq j \leq n-1\}. \tag{53}$$

It denotes a set of all Q which have a zero of multiplicities m'_{ij} at the interpolation points (P_j, σ_i). Let us further define

$$\mathcal{G}(x) = \prod_{\alpha \in \mathbb{A}_\Gamma} (x-\alpha)^{\mu^{(0)}_{\alpha_1}}. \tag{54}$$

Therefore,

$$\mathcal{G}(x) = \prod_{u=0}^{l} \mathcal{G}_{\Gamma_u}(x). \tag{55}$$

The following Lemma reveals the property of the candidates of $\mathcal{I}_{\mathbf{M}',l}$ when the ReT is applied.

Lemma 14: If $Q(x,y,z) \in \mathcal{I}_{\mathbf{M}',l}$, then $\mathcal{G} \mid Q(x,y,z\mathcal{G}_\Gamma)$.

Proof: Based on Theorem 11, Q can be written as $Q = \sum_{t=0}^{2l+1} \mathfrak{h}_t M_t$. Therefore, $Q(x,y,z\mathcal{G}_\Gamma) = \sum_{t=0}^{2l+1} \mathfrak{h}_t M_t(x,y,z\mathcal{G}_\Gamma)$. For each M_t, $M_t(x,y,z\mathcal{G}_\Gamma)$ can be expressed as

$$M_t(x,y,z\mathcal{G}_\Gamma) = \mathcal{G}_v^{(u)} \prod_{\epsilon=0}^{u-1} (z\mathcal{G}_\Gamma - \mathcal{K}_{\underline{\tilde{z}}^{(\epsilon)}}).$$

Based on eq. (33), for $\underline{\tilde{z}}^{(\epsilon)} = (\tilde{z}_0^{(\epsilon)}, \tilde{z}_1^{(\epsilon)}, \ldots, \tilde{z}_{n-1}^{(\epsilon)})$, $\mathcal{K}_{\underline{\tilde{z}}^{(\epsilon)}}$ can be written as

$$\begin{aligned}
&\mathcal{K}_{\underline{\tilde{z}}^{(\epsilon)}} \\
&= \sum_{j \in \Gamma_\epsilon} 0 \cdot \mathcal{L}_j + \sum_{j \in \bar{\Gamma}_\epsilon} \tilde{z}_j^{(\epsilon)} \mathcal{L}_j \\
&= \sum_{j \in \bar{\Gamma}_\epsilon} \tilde{z}_j^{(\epsilon)} \prod_{\alpha \in \mathbb{A} \setminus \{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta \in \mathbb{B}_j \setminus \{y_j\}} \frac{y-\beta}{y_j-\beta} \\
&= \mathcal{G}_{\Gamma_\epsilon} \sum_{j \in \bar{\Gamma}_\epsilon} \frac{\tilde{z}_j^{(\epsilon)}}{\mathcal{G}_{\Gamma_\epsilon}(x_j)} \prod_{\alpha \in \mathbb{A}_{\bar{\Gamma}_\epsilon} \setminus \{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta \in \mathbb{B}_j \setminus \{y_j\}} \frac{y-\beta}{y_j-\beta}.
\end{aligned} \tag{56}$$

Therefore,

$$M_t(x,y,z\mathcal{G}_\Gamma) = \mathcal{G}_v^{(u)} \prod_{\epsilon=0}^{u-1} \mathcal{G}_{\Gamma_\epsilon} \Big(z\mathcal{G}_{\Gamma \setminus \Gamma_\epsilon} - \sum_{j \in \bar{\Gamma}_\epsilon} \frac{\tilde{z}_j^{(\epsilon)}}{\mathcal{G}_{\Gamma_\epsilon}(x_j)} \times \prod_{\alpha \in \mathbb{A}_{\bar{\Gamma}_\epsilon} \setminus \{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta \in \mathbb{B}_j \setminus \{y_j\}} \frac{y-\beta}{y_j-\beta} \Big).$$

Since $\mathcal{G} = \prod_{\epsilon=0}^{l} \mathcal{G}_{\Gamma_\epsilon}$ and $\prod_{\epsilon=u}^{l} \mathcal{G}_{\Gamma_\epsilon} \mid \mathcal{G}_v^{(u)}$, $\mathcal{G} \mid M_t(x,y,z\mathcal{G}_\Gamma)$. Therefore, $\mathcal{G} \mid Q(x,y,z\mathcal{G}_\Gamma)$. ∎

In $\mathbf{M}'$, we replace the multiplicities of the transformed re-encoding points by zeroes, resulting in matrix $\tilde{\mathbf{M}}$. Let $\mathcal{I}_{\tilde{\mathbf{M}},l}$ denote the function space obtained by transforming the functions in $\mathcal{I}_{\mathbf{M}',l}$, which is defined as

$$\begin{aligned}
&\Phi : \mathcal{I}_{\mathbf{M}',l} \to \mathcal{I}_{\tilde{\mathbf{M}},l} \\
&Q(x,y,z) \mapsto \tilde{Q}(x,y,z) = \mathcal{G}^{-1}(x) Q(x,y,z\mathcal{G}_\Gamma(x)),
\end{aligned} \tag{57}$$

where Φ is an $\mathbb{F}_q[x]$-module isomorphism between $\mathcal{I}_{\mathbf{M}',l}$ and $\mathcal{I}_{\tilde{\mathbf{M}},l}$.

Theorem 15: $\mathcal{I}_{\tilde{\mathbf{M}},l}$ can be generated as an $\mathbb{F}_q[x]$-module by

$$\begin{aligned}
\tilde{\mathcal{M}} = \{\tilde{M}_t(x,y,z) \mid \tilde{M}_t(x,y,z) = \tilde{\mathcal{G}}_v^{(u)}(x,y) \prod_{\epsilon=0}^{u-1} \Big(z\mathcal{G}_{\Gamma \setminus \Gamma_\epsilon}(x) \\
- \sum_{j \in \bar{\Gamma}_\epsilon} \tilde{z}_j^{(\epsilon)*} \prod_{\alpha \in \mathbb{A}_{\bar{\Gamma}_\epsilon} \setminus \{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta \in \mathbb{B}_j \setminus \{y_j\}} \frac{y-\beta}{y_j-\beta} \Big), \\
t = v + 2u,\ v = 0,1 \text{ and } 0 \leq u \leq l\},
\end{aligned} \tag{58}$$

where

$$\tilde{\mathcal{G}}_v^{(u)}(x,y) = \frac{\mathcal{G}_v^{(u)}(x,y)}{\prod_{\epsilon=u}^{l} \mathcal{G}_{\Gamma_\epsilon}(x)} \tag{59}$$

and

$$\tilde{z}_j^{(\epsilon)*} = \frac{\tilde{z}_j^{(\epsilon)}}{\mathcal{G}_{\Gamma_\epsilon}(x_j)}. \tag{60}$$

Proof: Based on Lemma 14 and eq. (57), the desired result can be directly obtained. ∎

Note that instead of (59), $\tilde{\mathcal{G}}_v^{(u)}$ can be directly computed based on Theorem 9 for $\tilde{\mathbf{M}}$. Since the common divisor of $M_t(x,y,z\mathcal{G}_\Gamma)$ has been removed, the above generators of $\mathcal{I}_{\tilde{\mathbf{M}},l}$ have a lower x-degree than those of $\mathcal{I}_{\mathbf{M},l}$. This results in a simpler basis reduction computation. Let $\tilde{M}_t = \tilde{M}_t^{(0)} + \tilde{M}_t^{(1)} y + \cdots + \tilde{M}_t^{(2l+1)} y z^l$, the generators of $\mathcal{I}_{\tilde{\mathbf{M}},l}$ can be represented as a matrix $\tilde{\mathbf{V}} \in \mathbb{F}_q[x]^{2(l+1) \times 2(l+1)}$ by letting

$$\tilde{\mathbf{V}}_t = (\tilde{M}_t^{(0)}, \tilde{M}_t^{(1)}, \ldots, \tilde{M}_t^{(2l+1)}). \tag{61}$$

Hence, $\tilde{M}_t = \tilde{\mathbf{V}}_t \cdot (1, y, \ldots, yz^l)^T$ and $\tilde{\mathbf{V}}_{t,s} = \tilde{M}_t^{(s)}$, where $s = 0, 1, \ldots, 2l+1$. Based on the mapping $\Psi_{\underline{w}}$ of (39), $\Psi_{\underline{\tilde{w}}}$ can be similarly defined as: $\mathbb{F}_q[x]^{2(l+1)} \to \mathbb{F}_q[x]^{2(l+1)}$

$$\tilde{\mathbf{V}}_t \mapsto \tilde{\mathbf{V}}_t^* = \tilde{\mathbf{V}}_t \cdot \text{diag}(x^{\tilde{w}_0}, x^{\tilde{w}_1}, \ldots, x^{\tilde{w}_{2l+1}}), \tag{62}$$

where $\underline{\tilde{w}} = (\tilde{w}_0, \tilde{w}_1, \ldots, \tilde{w}_{2l+1})$ and $\tilde{w}_s = \left\lfloor \frac{(k-2\lfloor \frac{k-1}{2} \rfloor)\lfloor \frac{s}{2} \rfloor + 3(s \bmod 2)}{2} \right\rfloor$. Note that with the above transform, polynomials of $\mathcal{I}_{\tilde{\mathbf{M}},l}$ are organized under the $(1, (k-2\lfloor \frac{k-1}{2} \rfloor))$-revlex order [37]. Therefore, $\tilde{\mathbf{V}}$ can be mapped to $\tilde{\mathbf{V}}^*$. Applying the row reduction process described in Section III.C, $\tilde{\mathcal{M}}$ can be reduced into a Gröbner basis $\tilde{\mathcal{M}}'$. It contains the minimum polynomial $\tilde{\mathcal{Q}}$.

Theorem 16: Given $\tilde{\mathcal{Q}}(x,y,z)$ as the minimum polynomial in $\mathcal{I}_{\tilde{\mathbf{M}},l}$, the interpolation polynomial $\mathcal{Q}(x,y,z)$ can be obtained by

$$\mathcal{Q}(x,y,z) = \mathcal{G}(x) \tilde{\mathcal{Q}} \left(x, y, \frac{z - \mathcal{K}_\Gamma(x,y)}{\mathcal{G}_\Gamma(x)} \right). \tag{63}$$

Proof: Based on eqs. (49) and (57), if $\tilde{\mathcal{Q}} \in \mathcal{I}_{\tilde{\mathbf{M}},l}$, it can be written as $\tilde{\mathcal{Q}} = \sum_{t=0}^{2l+1} \mathfrak{h}_t \tilde{M}_t$. Therefore, substituting $\tilde{\mathcal{Q}}$ into eq. (63), it becomes

$$\mathcal{Q} = \sum_{t=0}^{2l+1} \mathfrak{h}_t \left(\mathcal{G}_v^{(u)} \prod_{\epsilon=0}^{u-1} (z - \mathcal{K}_\Gamma - \mathcal{K}_{\underline{\tilde{z}}^{(\epsilon)}}) \right).$$

Note that since $\mathcal{K}_\Gamma(P_j) + \mathcal{K}_{\underline{\tilde{z}}^{(\epsilon)}}(P_j) = z_j^{(\epsilon)}$, $\mathcal{Q} \in \mathcal{I}_{\mathbf{M},l}$.

Given $\tilde{\mathcal{Q}}$ as the minimum polynomial in $\mathcal{I}_{\tilde{\mathbf{M}},l}$, if there exists $\mathcal{Q}' \in \mathcal{I}_{\mathbf{M},l}$ that satisfies $\deg_{1,k}(\mathcal{Q}') < \deg_{1,k}(\mathcal{Q})$, we have $\tilde{\mathcal{Q}}' = \mathcal{G}^{-1}\mathcal{Q}'(x, y, (z + \mathcal{K}_\Gamma)\mathcal{G}_\Gamma)$. Since $\deg_{1,k}(\mathcal{K}_\Gamma) \leq k$, the mapping of $z \mapsto z + \mathcal{K}_\Gamma$ does not change the $(1, k)$-weighted degree of $\mathcal{Q}$ or $\mathcal{Q}'$. Since $\deg_{1,k}(\mathcal{G}_\Gamma) = 2\left\lfloor \frac{k-1}{2} \right\rfloor$,

$$\begin{aligned}
&\deg_{1,k}\left(\mathcal{Q}(x, y, z + \mathcal{K}_\Gamma)\right) \\
&\quad = \deg_{1,k}(\mathcal{G}) + \deg_{1,k}\left(\tilde{\mathcal{Q}}\left(x, y, \frac{z}{\mathcal{G}_\Gamma}\right)\right) \\
&\quad = \deg_{1,k}(\mathcal{G}) + \max_{0\leq s\leq l}\left\{\deg_{1,k}\left(\tilde{\mathcal{Q}}_{[s]}\left(\frac{z}{\mathcal{G}_\Gamma}\right)^s\right)\right\} \\
&\quad = \deg_{1,k}(\mathcal{G}) + \max_{0\leq s\leq l}\left\{\deg_{1,(k-2\left\lfloor\frac{k-1}{2}\right\rfloor)}\left(\tilde{\mathcal{Q}}_{[s]}z^s\right)\right\} \\
&\quad = \deg_{1,k}(\mathcal{G}) + \deg_{1,(k-2\left\lfloor\frac{k-1}{2}\right\rfloor)}\left(\tilde{\mathcal{Q}}\right).
\end{aligned}$$

Therefore, $\deg_{1,(k-2\left\lfloor\frac{k-1}{2}\right\rfloor)}(\tilde{\mathcal{Q}}) = \deg_{1,k}(\mathcal{Q}) - \deg_{1,k}(\mathcal{G})$ and $\deg_{1,(k-2\left\lfloor\frac{k-1}{2}\right\rfloor)}(\tilde{\mathcal{Q}}') = \deg_{1,k}(\mathcal{Q}') - \deg_{1,k}(\mathcal{G})$. Consequently, $\deg_{1,(k-2\left\lfloor\frac{k-1}{2}\right\rfloor)}(\tilde{\mathcal{Q}}') < \deg_{1,(k-2\left\lfloor\frac{k-1}{2}\right\rfloor)}(\tilde{\mathcal{Q}})$, which contradicts $\tilde{\mathcal{Q}}$ being the minimum polynomial. ■

Therefore, the interpolation polynomial $\mathcal{Q}'$ in $\mathcal{I}_{\mathbf{M}',l}$ can be obtained by

$$\mathcal{Q}'(x, y, z) = \mathcal{G}(x)\tilde{\mathcal{Q}}\left(x, y, \frac{z}{\mathcal{G}_\Gamma(x)}\right). \tag{64}$$

If f' is the z-roots of $\mathcal{Q}'$, estimation of the intended message polynomial f can be further obtained by

$$\hat{f}(x, y) = f'(x, y) + \mathcal{K}_\Gamma(x, y). \tag{65}$$

Summarizing this section, the ReT based ASD algorithm that utilizes the BR interpolation can be presented as in Algorithm 2.

Algorithm 2 The ReT Based ASD Algorithm

Input: $\mathbf{\Pi}$ and l;
Output: $\hat{f}$;

1: Compute $\mathbf{M}$ that sustains l;
2: Create balanced lists $\mathcal{S}'_j$ as in (24) and (25);
3: Select the re-encoding points as in (45);
4: Define $\mathcal{K}_\Gamma$ as in (46) and (47);
5: Transform all balanced lists as in (49);
6: Formulate basis of the module isomorphism $\tilde{\mathcal{M}}$ as in (58);
7: Map $\tilde{\mathcal{M}}$ to $\tilde{\mathbf{V}}^*$ as in (61) and (62);
8: Reduce $\tilde{\mathbf{V}}^*$ into $\tilde{\mathbf{V}}^{*\prime}$ using the MS algorithm;
9: Demap $\tilde{\mathbf{V}}^{*\prime}$ to $\tilde{\mathcal{M}}'$ as in (38) and (41);
10: Choose the minimum candidate of $\tilde{\mathcal{M}}'$ as $\tilde{\mathcal{Q}}$;
11: Construct $\mathcal{Q}'$ as in (64);
12: Find the z-roots of $\mathcal{Q}'$ and estimate $\hat{f}$ as in (65).

It should be pointed out that the complexity reducing approach in Section III.D can be similarly applied to this ReT variant. If $\deg_{1,k}\left(\mathcal{K}_{\underline{\tilde{z}}^{(u)}}\right) \leq k$, then $\underline{\tilde{z}}^{(u)}[\Upsilon_1] = \underline{0}$ and $\underline{\tilde{z}}^{(u)}$ would be a codeword. Based on eq. (65), the transformed message candidate would be $\mathcal{K}_{\underline{z}^{(u)}} = \mathcal{K}_{\underline{\tilde{z}}^{(u)}} + \mathcal{K}_\Gamma$, an ML codeword can again be validated. If $\underline{z}^{(u)}$ is an ML codeword, the decoding can be terminated and outputs $\mathcal{K}_{\underline{z}^{(u)}}$ as a message candidate.

TABLE I
INTERPOLATION COMPLEXITY OF ASD OF THE $(80, 39)$ ELLIPTIC CODE

l	Kötter (w/o ReT)	Kötter (w. ReT)	BR (w/o ReT)	BR (w. ReT)
4	5.07×10^6	2.03×10^6	2.40×10^6	1.19×10^6
8	6.91×10^7	2.69×10^7	3.80×10^7	1.79×10^7

TABLE II
INTERPOLATION COMPLEXITY OF ASD OF THE $(80, 69)$ ELLIPTIC CODE

l	Kötter (w/o ReT)	Kötter (w. ReT)	BR (w/o ReT)	BR (w. ReT)
4	6.73×10^6	1.42×10^6	8.94×10^5	2.35×10^5
8	1.25×10^8	1.39×10^7	1.72×10^7	2.16×10^6

V. COMPLEXITY ANALYSIS

This section analyzes complexity of the ASD algorithm as well as its ReT variant. As mentioned earlier, the decoding complexity is dominated by the interpolation. In case of the BR interpolation, it consists of the basis construction and its reduction, which will be characterized and validated by numerical results. In our simulations, we measure the complexity as the average number of finite field arithmetic operations in decoding a codeword.

In this paper, the MS algorithm [12] is utilized to reduce a general matrix Ξ into the weak Popov form. Given a matrix $\Xi \in \mathbb{F}_q[x]^{2(l+1)\times 2(l+1)}$, let $\Delta(\Xi)$ denote the orthogonality defect [38] as

$$\Delta(\Xi) = \mathrm{rowdeg}(\Xi) - \deg(\det(\Xi)), \tag{66}$$

where $\mathrm{rowdeg}(\Xi) = \sum_{t=0}^{2l+1}\deg(\Xi_t)$ and $\det(\Xi)$ is the determinant of Ξ. The following Lemma characterizes the complexity of the MS algorithm.

Lemma 17 [12]: Given a matrix $\Xi \in \mathbb{F}_q[x]$ of size $2(l+1) \times 2(l+1)$, the MS algorithm for computing a weak Popov form of Ξ exhibits a complexity of $O((2l+2)^2\deg(\Xi)\Delta(\Xi))$.

We first consider the basis construction complexity. The computation of $\mathcal{H}$ requires at most $O((\nu^{(u)})^3)$ finite field operations, where $\nu^{(u)}$ was defined in eq. (30). Based on Theorem 9, the complexity of computing $\mathcal{G}_v^{(u)}$ is $O(ln)$. Since $\deg_x\left(\mathcal{G}_v^{(u)}\right) < \frac{ln}{2}$ and $\deg_x\left(\mathcal{K}_{\underline{z}^{(u)}}\right) < \frac{n}{2}$, based on Theorem 11, the complexity of the basis construction is $O(l^2n^2)$. Based on Lemma 17, the complexity of the basis reduction will be determined by $\deg(\mathbf{V}^*)$ and $\Delta(\mathbf{V}^*)$. Since $\deg(\mathbf{V}^*) < \frac{ln}{2}$ and $\Delta(\mathbf{V}^*) < 2l^2(n-k)$, it exhibits a complexity of $O(l^5n(n-k))$.

For the ReT based ASD algorithm, transforming the interpolation points exhibits a complexity of $O(n^2)$. Based on eq. (59), the complexity of computing $\tilde{\mathcal{G}}_v^{(u)}$ is $O(ln)$. Based on eq. (58), the complexity of the basis construction can be characterized as $O(l^2(n-k)^2)$. Further based on Lemma 17, since $\deg\left(\tilde{\mathbf{V}}^*\right) < \frac{l(n-k)}{2}$ and $\Delta(\tilde{\mathbf{V}}^*) < 2l^2(n-k)$, it has a complexity of $O(l^5(n-k)^2)$. Therefore, the ReT helps reduce the BR interpolation complexity by a factor of $\frac{k}{n}$.

TABLE III
REDUCED BR INTERPOLATION COMPLEXITY OF ASD OF THE $(80, 39)$ ELLIPTIC CODE

$l = 4$	SNR	5	6	7	8	9	10
ASD	original	2.94×10^6	2.40×10^6	1.89×10^6	1.55×10^6	1.03×10^6	5.31×10^5
	reduced	2.93×10^6	2.39×10^6	1.87×10^6	1.52×10^6	9.29×10^5	3.82×10^5
ASD-ReT	original	1.60×10^6	1.19×10^6	8.99×10^5	7.15×10^5	5.00×10^5	3.24×10^5
	reduced	1.60×10^6	1.19×10^6	8.99×10^5	7.03×10^5	4.31×10^5	2.09×10^5

TABLE IV
REDUCED BR INTERPOLATION COMPLEXITY OF ASD OF THE $(80, 69)$ ELLIPTIC CODE

$l = 4$	SNR	5	6	7	8	9	10
ASD	original	1.30×10^6	8.94×10^5	5.17×10^5	2.88×10^5	2.23×10^5	2.08×10^5
	reduced	1.29×10^6	8.48×10^5	3.99×10^5	1.28×10^5	4.67×10^4	2.68×10^4
ASD-ReT	original	3.30×10^5	2.35×10^5	1.67×10^5	1.35×10^5	1.27×10^5	1.25×10^5
	reduced	3.29×10^5	2.23×10^5	1.53×10^5	1.23×10^5	1.18×10^5	1.16×10^5

The above analysis shows complexity of the BR interpolation reduces as the code rate increases. The ReT contributes to the complexity reduction by a factor of $\frac{k}{n}$, also the code rate, implying high rate codes can benefit more from the transform. Tables I-II show the numerical results of the interpolation in using different algorithms for decoding the $(80, 39)$ and the $(80, 69)$ elliptic codes, respectively. They are constructed based on $E : y^2 + y = x^3$, which is defined over $\mathbb{F}_{64}$. It should be pointed out that the computation required by the ReT itself is also counted. Our numerical results validate the above complexity characterizations. Moreover, comparing with Kötter's interpolation, the BR interpolation shows a smaller complexity. Note that without the ReT, Kötter's interpolation exhibits a complexity of $O(l^5k^2)$, while the BR interpolation exhibits a complexity of $O(l^5n(n-k))$. In the asymptotic manner, both k and $n-k$ can be replaced by n. This will result in the two interpolation approaches exhibiting a complexity of $O(l^5n^2)$. That says both Kötter's interpolation and the BR interpolation have the same asymptotic complexity. However, when resuming Kötter's interpolation complexity to $O(l^5k^2)$, it increases with the code rate, which is opposite to the BR interpolation. By pairing Tables I and II, we can observe that the higher rate code exhibits a lower BR complexity. The situation reverses for Kötter's interpolation.

Tables III and IV further show the complexity of the ASD and the ReT based ASD algorithms, both of which are assisted by the complexity reduction approach of Section VI. It shows that the BR interpolation complexity can be reduced by assessing degree of the module seeds $\mathcal{K}_{\underline{\tilde{z}}^{(u)}}$. This will be more obvious when the signal-to-noise ratio (SNR) increases, where the SNR is defined as E_b/N_0, where E_b and N_0 are the transmitted energy per information bit and the noise power density, respectively. Our simulation shows that at the high SNR region, more decoding events will exhibit $\deg_{1,k}\left(\mathcal{K}_{\underline{\tilde{z}}^{(u)}}\right) \leq k$ and $\underline{\tilde{z}}^{(u)}$ can be validated by the ML criterion.

VI. DECODING PERFORMANCE

This section presents the ASD performance of elliptic codes. The decoding frame error rate (FER) is obtained over the AWGN channel using BPSK modulation. The FER performance is presented as a function of the SNR.

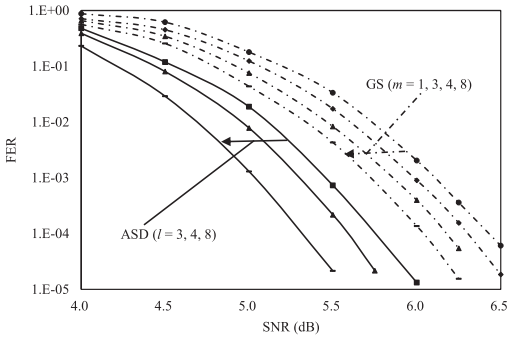

Fig. 1. Performance of the $(80, 39)$ elliptic code.

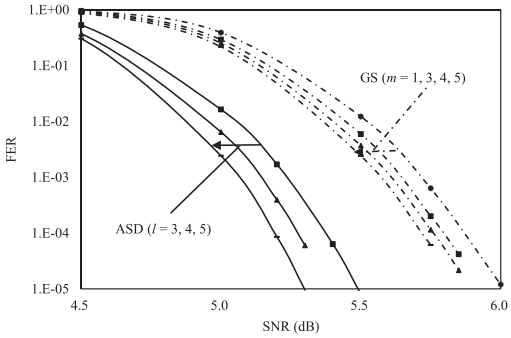

Fig. 2. Performance of the $(288, 163)$ elliptic code.

Figs. 1 and 2 show the performance of the GS and ASD algorithms in decoding the $(80, 39)$ and $(288, 163)$ elliptic codes,[2] respectively. The GS algorithm is parameterized by the interpolation multiplicity m. Our simulation results show that the ASD algorithm can substantially outperform the GS algorithm, due to its feature of using soft received information. For example, the ASD with $l = 3$ outperforms the GS with $m = 3$ by 0.5 dB at the FER of 1.0×10^{-4}. It should be pointed out that this performance advantage is realized with a smaller decoding computational cost. The interpolation cost $\mathfrak{C}_M$ (defined in eq. (15)) for the ASD is about 370. For the GS algorithm, it would be 480. For the ASD algorithm, the

[2]The $(288, 163)$ elliptic code is constructed based on $E : y^2+y = x^3+\mathfrak{a}_6$, which is defined over $\mathbb{F}_{256}$, where $\mathfrak{a}_6$ satisfies $\mathfrak{a}_6^{2^0} + \mathfrak{a}_6^{2^1} + \cdots + \mathfrak{a}_6^{2^7} = 1$.

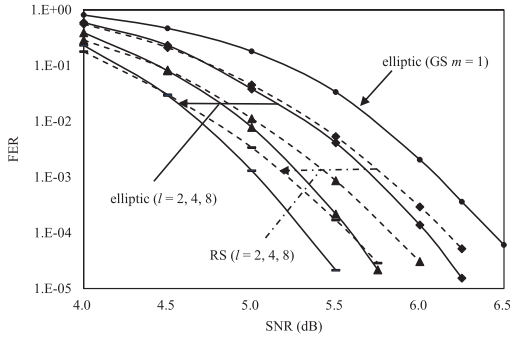

Fig. 3. Performance of the $(80, 39)$ elliptic code and the $(63, 31)$ RS code.

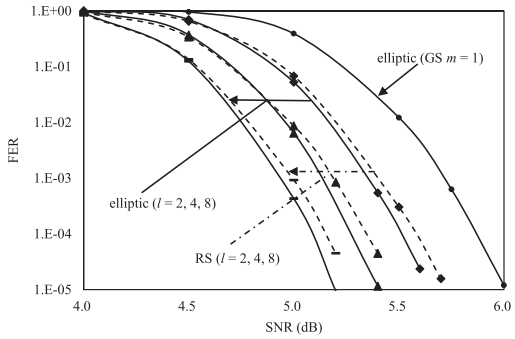

Fig. 4. Performance of the $(288, 163)$ elliptic code and the $(255, 144)$ RS code.

BR interpolation complexity is 9.58×10^5, while for the GS algorithm, it would be 1.43×10^6.

The elliptic codes are further compared with similar rate RS codes that are defined over the same finite field. Figs. 3 and 4 compare two pairs of similar rate elliptic codes and RS codes. Note that over the same finite field, elliptic codes have longer codeword length and inherit a greater error-correction capability, yielding a better decoding performance. It can been seen that with the same decoding OLS l, the elliptic codes can outperform the similar rate RS codes. Fig. 3 shows that decoding the $(80, 39)$ elliptic code with $l = 4$ performs similarly as decoding the $(63, 31)$ RS code with $l = 8$. However, Table IV of [35] shows that the decoding interpolation complexity is 3.01×10^7 for RS code. Table I shows the interpolation will be simpler for the elliptic code. If using Kötter's interpolation, the complexity will be 5.07×10^6 for the elliptic code, and 3.50×10^8 for the RS code. Fig. 4 compares decoding performance of the ASD of the $(288, 163)$ elliptic code and the $(255, 144)$ RS code. A similar phenomenon can also be observed. Therefore, empowered by a greater codeword length, elliptic codes can substantially outperform similar rate RS codes, but with less decoding computational cost.

VII. Conclusion

This paper has proposed the algebraic soft decoding algorithm of elliptic codes using module basis reduction interpolation, as well as its re-encoding transform variant. Based on multiplicity matrix $\mathbf{M}$, an interpolation ideal $\mathcal{I}_{\mathbf{M}}$ has been defined. With the decoding output list size l, the module $\mathcal{I}_{\mathbf{M},l}$ can be further defined. By characterizing the zero basis of each affine point, the basis for a sequence of $\mathbb{F}_q[x]$-submodules of $\mathcal{R}$ have been proposed. Based on the Lagrange interpolation functions, generators for $\mathcal{I}_{\mathbf{M},l}$ have been formulated, forming a module basis. This basis can be further reduced by the Mulders-Storjohann algorithm, resulting in the desired Gröbner basis that contains the interpolation polynomial $\mathcal{Q}$. The re-encoding transform has also been introduced to reduce the degree of the module basis entries, which exhibits a lower computational complexity. By assessing the degree of the Lagrange interpolation polynomials, the basis reduction interpolation complexity can be further reduced. The complexity of these proposed algorithms have been analyzed, which yields a complexity of $O(l^5 n(n-k))$ and $O(l^5(n-k)^2)$ in the basis reduction interpolation in the cases of without the re-encoding transform and with the re-encoding transform, respectively. Such characterizations have shown that both the basis reduction interpolation and the re-encoding transform are more effective in yielding a low complexity for high rate code. This has also been verified by our numerical results. Finally, our simulation results have demonstrated algebraic soft decoding's performance advantage over the Guruswami-Sudan algorithm, as well as elliptic codes' performance advantage over similar rate Reed-Solomon codes.

Appendix

Recalling eq. (29), $\mathrm{v}_{P_{\alpha_0}^{(u)}}(y - \mathcal{H}(x)) \geq \mu_\alpha^{(u)}, \ \forall \alpha \in \mathbb{A}$. Based on eq. (8), for each P_j,

$$y = \sum_{b \in \mathbb{N}} \xi_{2,P_j,b} \psi_{P_j,b} \tag{67}$$

and

$$\mathcal{H}(x) = \sum_{b \in \mathbb{N}} (\zeta_0 \xi_{0,P_j,b} + \sum_{i=1}^{\nu^{(u)}-1} \zeta_i \xi_{2i-1,P_j,b}) \psi_{P_j,b}. \tag{68}$$

Therefore, for $P_{\alpha_0}^{(u)}$, if $\zeta_0 \xi_{0,P_{\alpha_0}^{(u)},b} + \sum_{i=1}^{\nu^{(u)}-1} \zeta_i \xi_{2i-1,P_{\alpha_0}^{(u)},b} = \xi_{2,P_{\alpha_0}^{(u)},b}$ with $0 \leq b < \mu_\alpha^{(u)}$, $y - \mathcal{H}(x)$ satisfies the required conditions. Consequently, the $\mathcal{H}(x)$ can be determined by solving the linear system

$$\underline{\zeta} \mathfrak{S} = \underline{\xi}, \tag{69}$$

where $\mathfrak{S}$ is a square matrix of size $\nu^{(u)}$, $\underline{\zeta} = (\zeta_0, \zeta_1, \ldots, \zeta_{\nu^{(u)}-1})$, and $\underline{\xi} = (\xi_{2,P_{\alpha_0}^{(u)},0}, \xi_{2,P_{\alpha_0}^{(u)},1} \ldots, \xi_{2,P_{\alpha_0}^{(u)},\mu_\alpha^{(u)}-1})$. The zero basis functions of each affine point can be generated based on Theorem 3 of [21]. The corresponding coefficients $\xi_{a,P_j,b}$ can be further determined. Therefore, $\mathcal{H}(x)$ is obtained.

References

[1] V. D. Goppa, "Codes on algebraic curves," *Soviet Math. Dokl.*, vol. 24, no. 1, pp. 170–172, Jul. 1981.
[2] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.
[3] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equation for decoding Goppa codes," *Inf. Control*, vol. 27, no. 1, pp. 87–99, 1975.
[4] L. R. Welch and E. R. Berlekamp, "Error correction for algebraic block codes," U.S. Patent 4 633 470, Dec. 30, 1986.
[5] M. Sudan, "Decoding of Reed–Solomon codes beyond the error-correction bound," *J. Complex.*, vol. 13, no. 1, pp. 180–193, Mar. 1997.
[6] 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.

[7] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.
[8] X.-W. Wu and P. H. 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.
[9] L. Chen, R. A. Carrasco, M. Johnston, and E. G. Chester, "Efficient factorisation algorithm for list decoding algebraic-geometric and Reed–Solomon codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Glasgow, U.K., Jun. 2007, pp. 851–856.
[10] R. Kötter, "On algebraic decoding of algebraic-geometric and cyclic codes," Ph.D. dissertation, Dept. Electr. Eng., Linköping Univ., Linköping, Sweden, 1996.
[11] K. Lee and M. E. O'Sullivan, "An interpolation algorithm using Gröbner bases for soft-decision decoding of Reed–Solomon codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Seattle, WA, USA, Jul. 2006, pp. 2032–2036.
[12] T. Mulders and A. Storjohann, "On lattice reduction for polynomial matrices," *J Symbolic Comput.*, vol. 35, no. 4, pp. 377–401, Apr. 2003.
[13] 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, Jul. 2005.
[14] P. Giorgi, C.-P. Jeannerod, and G. Villard, "On the complexity of polynomial matrix computations," in *Proc. Int. Symp. Symbolic Algebr. Comput. (ISSAC)*, 2003, pp. 135–142.
[15] R. Kötter and A. Vardy, "A complexity reducing transformation in algebraic list decoding of Reed–Solomon codes," in *Proc. IEEE Inform. Theory Workshop (ITW)*, Paris, France, Mar. 2003, pp. 10–13.
[16] J. Xing, L. Chen, and M. Bossert, "Progressive algebraic soft-decision decoding of Reed–Solomon codes using module minimization," *IEEE Trans. Commun.*, vol. 67, no. 11, pp. 7379–7391, Nov. 2019.
[17] S. Sakata, "Extension of the Berlekamp-Massey algorithm to N dimensions," *Inf. Comput.*, vol. 84, no. 2, pp. 207–239, Feb. 1990.
[18] G. Feng and T. Rao, "Decoding algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inf. Theory*, vol. 39, no. 1, pp. 37–46, Jan. 1993.
[19] S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Hoholdt, "Fast decoding of algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, pp. 1672–1677, Nov. 1995.
[20] T. Høholdt and R. Nielsen, "Decoding Hermitian codes with Sudan's algorithm," in *AAECC* (Lecture Notes in Computer Science), vol. 1719. Berlin, Germany: Springer-Verlag, 1999, pp. 260–269.
[21] Y. Wan, L. Chen, and F. Zhang, "Design of Guruswami–Sudan list decoding for elliptic codes," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Visby, Sweden, Aug. 2019, pp. 1–5.
[22] K. Lee and M. E. O'Sullivan, "List decoding of Hermitian codes using Gröbner bases," *J. Symbolic Comput.*, vol. 44, no. 12, pp. 1662–1675, 2009.
[23] 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.
[24] P. Beelen and K. Brander, "Efficient list decoding of a class of algebraic-geometry codes," *Adv. Math. Commun.*, vol. 4, no. 4, pp. 485–518, 2010.
[25] J. S. R. Nielsen and P. Beelen, "Sub-quadratic decoding of one-point Hermitian codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3225–3240, Jun. 2015.
[26] Y. Wan, L. Chen, and F. Zhang, "Algebraic list decoding of elliptic codes through module basis reduction," in *Proc. Int. Symp. Inf. Theory App. (ISITA)*, Kapolei, HI, USA, Oct. 2020, pp. 185–189.
[27] L. Chen, R. Carrasco, and M. Johnston, "Soft-decision list decoding of Hermitian codes," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2169–2176, Aug. 2009.
[28] K. Lee and M. E. O'Sullivan, "Algebraic soft-decision decoding of Hermitian codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2587–2600, Jun. 2010.
[29] 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, Apr. 2018.
[30] L. Washington, *Elliptic Curves: Number Theory Cryptography*. Boca Raton, FL, USA: CRC Press, 2008.
[31] C. Munuera, "On MDS elliptic codes," *Discrete Math.*, vol. 117, nos. 1–3, pp. 279–286, 1993.
[32] H. Stichtenoth, *Algebraic Function Fields and Codes*, vol. 254. Berlin, Germany: Springer-Verlag, 2009.
[33] Y. Wan, L. Chen, and F. Zhang, "Algebraic soft decoding of elliptic codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Melbourne, FL, Australia, Jul. 2021, pp. 521–526.
[34] V. Neiger, "Bases of relations in one or several variables: Fast algorithms and applications," Ph.D. dissertation, Laboratoire de l'Informatique du Parallélisme, École Normale Supérieure de Lyon-Univ. Waterloo, Lyon, France, 2016.
[35] J. Xing, L. Chen, and M. Bossert, "Module minimisation based low-complexity soft decoding of Reed–Solomon codes," *IET Commun.*, vol. 13, no. 20, pp. 3489–3499, Dec. 2019.
[36] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 320–327, Mar. 1994.
[37] 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.
[38] J. S. R. Nielsen and A. Zeh, "Multi-trial Guruswami–Sudan decoding for generalised Reed–Solomon codes," *Designs, Codes Cryptogr.*, vol. 73, no. 2, pp. 507–527, Nov. 2014.

Yunqi Wan received the B.Sc. degree in mathematics and applied mathematics and the M.Sc. degree in probability and statistics from Northwest Normal University, Lanzhou, China, in 2011 and 2017, respectively, and the Ph.D. degree in electronics and information technology from Sun Yat-sen University, Guangzhou, China, in 2021. His research interests include channel coding and its applications.

Li Chen (Senior Member, IEEE) received the B.Sc. degree in applied physics from Jinan University, China, in 2003, and the M.Sc. degree in communications and signal processing and the Ph.D. degree in communications engineering from Newcastle University, U.K., in 2004 and 2008, respectively. From 2007 to 2010, he was a Research Associate with Newcastle University. In 2010, he returned China as a Lecturer of the School of Information Science and Technology, Sun Yat-sen University, Guangzhou. From 2011 to 2012, he was a Visiting Researcher with the Institute of Network Coding, The Chinese University of Hong Kong. From 2011 and 2016, he was an Associate Professor and a Professor of the university. Since 2013, he has been the Associate Head of the Department of Electronic and Communication Engineering (ECE). From July 2015 to October 2015, he was a Visitor of the Institute of Communications Engineering, Ulm University, Germany. From October 2015 to June 2016, he was a Visiting Associate Professor with the Department of Electrical Engineering, University of Notre Dame, USA. From 2017 to 2020, he was the Deputy Dean of the School of Electronics and Communication Engineering. His research interests include information theory, error-correction codes, and data communications. He is a Senior Member of the Chinese Institute of Electronics (CIE). He is a member of the IEEE Information Theory Society Board of Governors and its External Nomination Committee and the Chair of its Conference Committee. He is also the Chair of the IEEE Information Theory Society Guangzhou Chapter. He is an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. He has been organizing several international conferences and workshops, including the 2018 IEEE Information Theory Workshop (ITW) at Guangzhou and the 2022 IEEE East Asian School of Information Theory (EASIT) at Shenzhen, for which he is the General Co-Chair. He is also the TPC Co-Chair of the 2022 IEEE/CIC International Conference on Communications in China (ICCC) at Foshan. He likes reading and photography.

Fangguo Zhang received the Ph.D. degree from the School of Communication Engineering, Xidian University, in 2001. He is currently a Professor at the School of Computer Science and Engineering, Sun Yat-sen University, China. He is the Co-Director of the Guangdong Key Laboratory of Information Security Technology. His research mainly focuses on cryptography and its applications. His specific interests are elliptic curve cryptography, secure obfuscation, blockchain, anonymity, and privacy.