Guruswami-Sudan Decoding of Elliptic Codes Through Module Basis Reduction

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

***Abstract*— This paper proposes the Guruswami-Sudan (GS) list decoding algorithm for one-point elliptic codes, in which the interpolation is realized by the module basis reduction (BR). Elliptic codes are a kind of algebraic-geometric (AG) codes with a genus of one. Over the same finite field, they have a greater codeword length than Reed-Solomon (RS) codes, capable of correcting more errors. The GS decoding consists of interpolation and root-finding, while the former that determines the interpolation polynomial $\mathcal{Q}(x, y, z)$ dominates the decoding complexity. By defining the Lagrange interpolation function over an elliptic function field, a basis of the interpolation module can be constructed. The desired Gröbner basis that contains $\mathcal{Q}(x, y, z)$ can be determined by reducing the constructed basis. This is namely the BR interpolation and it requires less finite field arithmetic operations than the conventional Kötter's interpolation, facilitating the GS decoding. Re-encoding transform (ReT) is further introduced to facilitate the BR interpolation. This work also shows that both the BR interpolation and its ReT variant will have a lower complexity as the code rate k/n increases, where n and k are the length and dimension of the code, respectively. Our numerical results demonstrate the complexity advantage of the BR interpolation over Kötter's interpolation, and the performance advantage of elliptic codes over RS codes.**

***Index Terms*— Algebraic-geometric codes, basis reduction, elliptic codes, Gröbner basis, interpolation, list decoding.**

I. Introduction

ALGEBRAIC-GEOMETRIC (AG) codes were introduced by Goppa [1]. They are constructed based on algebraic curves over a finite field. Reed-Solomon (RS) codes can be seen as a special class of AG codes that are constructed from a straight line. Hence, the length of an RS code cannot exceed the size of the finite field, limiting the code's minimum Hamming distance and the number of errors that can be corrected by the code. However, there exist other algebraic curves on which the number of rational points[1] can be greater than the size of the field. This gives AG codes the codeword length and distance advantages over RS codes. Therefore, AG codes have the potential to replace RS codes in communication and storage systems. They can also be applied in cryptography and complexity theory. For an AG code, its minimum Hamming distance would be lower bounded by the designed distance d^*, where $d^* = n-k-g+1$, n and k are the length and dimension of the code, respectively, and g is the genus of the curve. Note that for RS codes, $g = 0$ and they become the maximum distance separable (MDS) codes. For AG codes, $g > 0$, they are usually not MDS codes. In the AG family, Hermitian codes have a large codeword length and Hermitian curves also exhibit rich structural symmetries which are beneficial for coding practice. However, their codeword length comes at the cost of a large genus penalty. For elliptic codes, $g = 1$. They are either MDS[2] or almost MDS codes, yielding a good tradeoff between its codeword length and distance property.

Decoding of AG codes can be categorized into the syndrome based approach and the interpolation based approach. The syndrome based decoding algorithms yield a unique decoded message. Syndromes are used to determine the error locations and their magnitudes. Driencourt [2] presented a decoding algorithm for elliptic codes, which is the early syndrome based decoding for the code. It can correct at most $\lfloor \frac{d^*-1}{4} \rfloor$ errors. Following Peterson's algorithm [3] for decoding BCH codes, Justesen *et al.* [4] proposed a decoding algorithm for a class of AG codes based on plane curves. Skorobogatov and Vladut [5] further generalized the algorithm to decode AG codes that are constructed from an arbitrary algebraic curve, presenting an algorithm that can correct at most $\lfloor \frac{d^*-1}{2} \rfloor$ errors. By introducing majority voting for the unknown syndromes, Feng and Rao [6] proposed a decoding algorithm for one-point AG codes that can correct up to $\lfloor \frac{d^*-1}{2} \rfloor$ errors. Duursma [7] further extended the algorithm to decode arbitrary AG codes. The original Berlekamp-Massey (BM) algorithm on univariate linear recurring relation was generalized by Sakata [8] to the multivariate domain for decoding AG codes, which is called the BMS algorithm. Based on the majority voting and the BMS algorithm, Sakata *et al.* [9] presented a more efficient decoding algorithm for AG codes, with a complexity of $O(n^{7/3})$. Its performance in decoding Hermitian codes over wireless channels has been later investigated by Johnston and Carrasco [10], shedding lights on their practical applications. Based on the shift-register

Manuscript received May 28, 2020; revised March 2, 2021; accepted July 13, 2021. Date of publication August 31, 2021; date of current version October 20, 2021. This work was sponsored in part by the National Natural Science Foundation of China (NSFC) under Project 62071498 and Project 61972429 and in part by the Guangdong Major Project of Basic and Applied Basic Research under Project 2019B030302008. An earlier version of this paper was presented in part at the 2020 International Symposium on Information Theory and Its Applications. *(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, and also with Guangdong Province Key Laboratory of Information Security Technology, Guangzhou 510006, China (e-mail: isszhfg@mail.sysu.edu.cn).

Communicated by G. Matthews, Associate Editor for Coding Theory.

Digital Object Identifier 10.1109/TIT.2021.3109447

[1]Rational points are points on algebraic curves with all their coordinates from the finite field.

[2]The condition for an elliptic code being MDS will be explained in Section II.A.

synthesis techniques, Schmidt *et al.* [11] proposed power decoding for low rate RS codes, which can correct more than $\lfloor \frac{d^*-1}{2} \rfloor$ errors. Nielsen and Beelen [12] further generalized it to decode low rate Hermitian codes.

The interpolation based algebraic list decoding has an error-correction capability beyond $\lfloor \frac{d^*-1}{2} \rfloor$. Sudan [13] first proposed the decoding concept for low rate RS codes. Shokrollahi and Wasserman further generalized it for low rate AG codes [14]. By constructing a curve that passes through all interpolation points with a certain multiplicity, Guruswami and Sudan [15] later improved it to decode all rate RS and AG codes, namely, the Guruswami-Sudan (GS) algorithm. It can correct up to $n - \lfloor \sqrt{n(n-d^*)} \rfloor - 1$ errors. The GS algorithm consists of interpolation and root-finding, where the former dominates the decoding complexity. Interpolation constructs a minimum polynomial that has a zero of multiplicity m over a set of interpolation points. This construction can be realized by Kötter's iterative polynomial construction [16]. The decoded message can be obtained by finding roots of the interpolation polynomial. Soft-decision algebraic list decoding of RS codes was later proposed by Kötter and Vardy [17]. To facilitate Kötter's interpolation, re-encoding transform (ReT) has been introduced [18]. By defining the zero basis for each affine point, Høholdt and Nielsen [19] presented a mathematical framework for GS decoding of Hermitian codes, including Kötter's interpolation as the key decoding step. Under Kötter's interpolation paradigm, soft-decision algebraic list decoding of Hermitian codes was later proposed by Chen *et al.* [20]. Recently, GS decoding of elliptic codes using Kötter's interpolation was proposed by the authors [21].

The other interpolation technique is from the perspective of the Gröbner basis of a module [22]. One may form the basis of a module that contains polynomials with the interpolation multiplicity and degree constraints. The basis will be further reduced, yielding the Gröbner basis that contains the desired interpolation polynomial. This is called the basis reduction (BR) interpolation [23]. In comparison with Kötter's interpolation for decoding AG codes, it not only requires less computation, but also eliminates the need of pre-computing the zero basis for each affine point and the corresponding coefficients [24], [25]. Lee and O'Sullivan proposed the GS decoding of Hermitian codes using the BR interpolation [26]. They later generalized the interpolation technique for soft-decision algebraic list decoding of Hermitian codes [27]. Both the Lee-O'Sullivan algorithm [26] and the Mulders-Storjohann (MS) algorithm [28] can be used to reduce the module basis into a Gröbner basis. Other facilitating techniques for the basis reduction process include the Alekhnovich algorithm [29] and the Giorgi-Jeannerod-Villard (GJV) algorithm [30]. Applying the former, Beelen and Brander reduced the complexity in finding the interpolation polynomial for a class of AG codes [31]. Applying the latter, Nielsen and Beelen also presented a fast GS decoding algorithm for Hermitian codes [12].

This paper introduces the GS decoding of one-point elliptic codes using the BR interpolation. In order to construct the basis of a module, the Lagrange interpolation functions over the elliptic function field are introduced. They lead to the formulation of the module basis generators. The constructed basis will be reduced to the desired Gröbner basis by Lee-O'Sullivan's algorithm [26], from which the interpolation polynomial $\mathcal{Q}(x, y, z)$ can be obtained. In order to further reduce the interpolation complexity, the ReT is introduced for the decoding. It transforms the received word into containing at least ε zero symbols, where $\varepsilon = k-1$ or $k-2$. This results in basis generators of the module sharing a common factor which can be removed to reduce the basis reduction complexity. This work shows that the BR interpolation exhibits a complexity of $O(l^3\ m^2 n(n-k))$, where m is the interpolation multiplicity and l is the maximum decoding output list size. The ReT can help reduce it to $O(l^3\ m^2(n-k)^2)$. Hence, both the BR interpolation and its ReT variant will have a lower complexity as the code rate k/n increases. This is in contrast to Kötter's interpolation whose complexity grows with the rate. This work also discusses the impact of using the Alekhnovich algorithm and the GJV algorithm for the basis reduction process. They are capable to reduce the complexity into quasi-linear in n. However, we show that such complexity advantage can only be realized when n is so large that beyond the current practical interest. Finally, our numerical results demonstrate the complexity advantage of the BR interpolation over Kötter's interpolation in decoding elliptic codes, as well as the performance advantage of elliptic codes over RS codes.

II. Background Knowledge

A. Elliptic Codes

Let $\mathbb{F}_q$ 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

$$Y^2Z + \mathfrak{a}_1 XYZ + \mathfrak{a}_3 YZ^2 = X^3 + \mathfrak{a}_2 X^2 Z + \mathfrak{a}_4 XZ^2 + \mathfrak{a}_6 Z^3, \tag{1}$$

where $\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{a}_3, \mathfrak{a}_4, \mathfrak{a}_6 \in \mathbb{F}_q$. On E, there exists the point of infinity $P_\infty = (0, 1, 0)$. With $Z = 1$, curve

$$Y^2 + \mathfrak{a}_1 XY + \mathfrak{a}_3 Y = X^3 + \mathfrak{a}_2 X^2 + \mathfrak{a}_4 X + \mathfrak{a}_6 \tag{2}$$

becomes an affine component, which is called the affine curve. Points on the curve are called affine points. They are denoted as $P_i = (x_i, y_i)$. Let $E(\mathbb{F}_q)$ denote the set of $\mathbb{F}_q$-rational points on E as $E(\mathbb{F}_q) = \{P_i\} \bigcup \{P_\infty\}$. They form an additive Abelian group based on the "chord-and-tangent" rule with P_∞ as the identity element [32]. Let $-P_i$ denote the inverse of P_i, P_i and $-P_i$ are the only affine points on E with the same x-coordinate. For any P_i, there exists a smallest nonnegative integer δ such that $\delta P_i = P_\infty$, where δ is the order of P_i. It divides the order of $E(\mathbb{F}_q)$. Coordinate ring of E is an integral domain, i.e.,

$$\begin{aligned} \mathcal{R} = \ & \mathbb{F}_q[X, Y] / < Y^2 + \mathfrak{a}_1 XY + \mathfrak{a}_3 Y \\ & - X^3 - \mathfrak{a}_2 X^2 - \mathfrak{a}_4 X - \mathfrak{a}_6 > = \mathbb{F}_q[x, y], \end{aligned} \tag{3}$$

where x and y denote the residue classes of X and Y, respectively. Elements of $\mathcal{R}$ are bivariate polynomials with

y-degree less than two. Based on E, its function field $\mathbb{F}_q(E)$ would be the quotient field of $\mathcal{R}$.

Given $h \in \mathbb{F}_q(E)$, its order at a rational point P is $v_P(h)$, where $v_P(\cdot)$ denotes the valuation of h at P [32]. There exists a function Λ which is called the local parameter of P. It enables $v_P(\Lambda) = 1$ and $h = \Lambda^{v_P(h)}h'$, where $v_P(h') = 0$. If $v_P(h) > 0$, h has a zero of order $v_P(h)$ at P. Otherwise, it has a pole of order $-v_P(h)$ at P. For elliptic curves, $-v_{P_\infty}(x) = 2$, $-v_{P_\infty}(y) = 3$ and $-v_{P_\infty}(x^\lambda y^\gamma) = 2\lambda + 3\gamma$.

Definition 1 ([32]): For each point P, we define a formal symbol $[P]$. Let n_P denote an integer that corresponds to P, $D = \sum_{P \in E(\mathbb{F}_q)} n_P[P]$ is a divisor of E. It has a degree of $\deg(D) = \sum_{P \in E(\mathbb{F}_q)} n_P$ and a sum of $\mathrm{sum}(D) = \sum_{P \in E(\mathbb{F}_q)} n_P P$.

Definition 2 ([32]): If $h \in \mathbb{F}_q(E)$ and $h \neq 0$, the divisor of h is defined as $\mathrm{div}(h) = \sum_{P \in E(\mathbb{F}_q)} v_P(h)[P]$. $\mathrm{div}(h)$ is also called the principal divisor of E.

For any divisor D, let $\mathfrak{L}(D)$ denote the Riemann-Roch space defined by D. Therefore, $\mathfrak{L}(u[P_\infty]) = \{h \in \mathbb{F}_q(E) | \mathrm{div}(h) + u[P_\infty] \succeq 0\} \bigcup \{0\}$ has a basis consisting of

$$\{\phi_a = x^\lambda y^\gamma \mid 2\lambda + 3\gamma \leq u, \lambda \in \mathbb{N}, \gamma \in \{0, 1\}\} \tag{4}$$

which satisfies $-v_{P_\infty}(\phi_a) < -v_{P_\infty}(\phi_{a+1})$, where "$\succeq$" indicates that the coefficients of $\mathrm{div}(h) + u[P_\infty]$ are nonnegative and $\mathbb{N}$ denotes the set of nonnegative integers. The basis of (4) 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 $-v_{P_\infty}(h) = \max\{-v_{P_\infty}(\phi_a) \mid \zeta_a \neq 0\}$.

Given n distinct affine points $P_0, P_1, \ldots, P_{n-1}$, a divisor $G = \sum_{i=0}^{n-1}[P_i]$ can be formed. Let $f \in \mathfrak{L}(k[P_\infty])$ denote the message polynomial that is

$$f(x, y) = f_0\phi_0 + f_1\phi_1 + \cdots + f_{k-1}\phi_{k-1}, \tag{5}$$

where $f_0, f_1, \ldots, f_{k-1} \in \mathbb{F}_q$ are the message symbols. Based on E, an (n, k) one-point elliptic code can be generated by the following evaluation

$$\begin{aligned}\mathcal{C}_E(G, k[P_\infty]) = &\{(f(P_0), f(P_1), \ldots, f(P_{n-1})), \\ &\forall f \in \mathfrak{L}(k[P_\infty])\},\end{aligned} \tag{6}$$

where codeword $\underline{c} = (c_0, c_1, \ldots, c_{n-1}) = (f(P_0), f(P_1), \ldots, f(P_{n-1})) \in \mathbb{F}_q^n$. Its minimum Hamming distance[3] $d \geq d^* = n - k$.

The above description shows that the number of affine points on the curve defines the length of the code. Based on the Hasse-Weil bound [34], the maximum number of rational points on a curve defined over $\mathbb{F}_q$ is $q + g\lfloor 2\sqrt{q} \rfloor + 1$, where g is the genus of a nonsingular curve of degree $\mathfrak{b}$ and $g = \frac{(\mathfrak{b}-1)(\mathfrak{b}-2)}{2}$. For elliptic curves, $\mathfrak{b} = 3$, $g = 1$ and $|E(\mathbb{F}_q)| \leq q + \lfloor 2\sqrt{q} \rfloor + 1$. It should be pointed that the existence of affine points of order two will make the following interpolation module basis construction cumbersome. Therefore, when designing an elliptic code, the curve coefficients $\mathfrak{a}_1, \mathfrak{a}_2, \mathfrak{a}_3, \mathfrak{a}_4, \mathfrak{a}_6$ should be chosen appropriately, so that the curve does not have affine points of order two, but the number of rational points can still reach the Hasse-Weil bound. Therefore, in this paper, the constructed elliptic codes have length $q + \lfloor 2\sqrt{q} \rfloor$.

[3]Note that an (n, k) elliptic code will be an MDS code if and only if for any $\{P_{i_1}, P_{i_2}, \ldots, P_{i_k}\} \subseteq \{P_0, P_1, \ldots, P_{n-1}\}$, $[P_{i_1}] + [P_{i_2}] + \cdots + [P_{i_k}] - k[P_\infty]$ is not a principal divisor [33].

B. The GS Decoding

Let $\mathcal{R}[z]$ denote the polynomial ring defined 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, it can be seen as a variant of $\underline{c}$, i.e., $\underline{r} = \underline{c} + \underline{e}$, where $\underline{e} \in \mathbb{F}_q^n$ is an error vector.

The GS decoding consists of interpolation and root-finding. First, let us define $\mathbf{P}$ as the set of n interpolation points

$$\mathbf{P} = \{(P_0, r_0), (P_1, r_1), \ldots, (P_{n-1}, r_{n-1})\}. \tag{7}$$

Interpolation: Given a received word $\underline{r}$ and the decoding parameters m and l, construct a nonzero minimum polynomial $\mathcal{Q}(x, y, z) = \sum_{j=0}^{l} \mathcal{Q}_{[j]}(x, y)z^j \in \mathcal{R}[z]_l$, it interpolates all points of $\mathbf{P}$ with a multiplicity of m.

Root-finding: Given the interpolation polynomial $\mathcal{Q}(x, y, z)$, find all z-roots in the form of $f(x, y)$.

A polynomial Q in $\mathcal{R}[z]_l$ can be written as[4]

$$Q = \sum_{a \in \mathbb{N}} \sum_{b \leq l} Q_{ab}\phi_a z^b, \tag{8}$$

where $Q_{ab} \in \mathbb{F}_q$. For each affine point P_i, there exists a basis $\{\psi_{P_i,0}, \psi_{P_i,1}, \ldots, \psi_{P_i,u-1}\}$ of $\mathfrak{L}(u[P_\infty])$ such that $\psi_{P_i,\mu}(x_i, y_i) = 0$ and $v_{P_i}(\psi_{P_i,\mu}) = \mu$. Functions of this basis have an increasing zero order w.r.t. P_i [19]. The complexity of computing the basis is $O(u^2)$. Given a pole basis monomial ϕ_a, we have

$$\phi_a = \sum_{\mu \in \mathbb{N}} \xi_{a,P_i,\mu}\psi_{P_i,\mu}, \tag{9}$$

where $\xi_{a,P_i,\mu} \in \mathbb{F}_q$ is the corresponding coefficient between ϕ_a and $\psi_{P_i,\mu}$ [19], [24]. Since

$$z^b = (z - r_i + r_i)^b = \sum_{\nu \leq b} \binom{b}{\nu} r_i^{b-\nu} (z - r_i)^\nu, \tag{10}$$

together with (9) and (10), Q can be written as

$$\begin{aligned}Q = &\sum_{a \in \mathbb{N}} \sum_{b \leq l} Q_{ab} \left(\sum_{\mu \in \mathbb{N}} \xi_{a,P_i,\mu}\psi_{P_i,\mu} \right) \\ &\cdot \left(\sum_{\nu \leq b} \binom{b}{\nu} r_i^{b-\nu} (z - r_i)^\nu \right) \\ = &\sum_{\mu,\nu \in \mathbb{N}} \left(\sum_{a \in \mathbb{N}} \sum_{b=\nu}^{l} Q_{ab} \binom{b}{\nu} \xi_{a,P_i,\mu} r_i^{b-\nu} \right) \psi_{P_i,\mu} (z - r_i)^\nu.\end{aligned} \tag{11}$$

We denote

$$\mathcal{D}_{\mu\nu}^{(P_i,r_i)}(Q) = \sum_{a \in \mathbb{N}} \sum_{b=\nu}^{l} Q_{ab} \binom{b}{\nu} \xi_{a,P_i,\mu} r_i^{b-\nu} \tag{12}$$

[4]In this paper, we denote the intended interpolation polynomial as $\mathcal{Q}$, while a general polynomial in $\mathcal{R}[z]_l$ is denoted as Q.

as the (μ,ν)-Hasse derivative evaluation of Q at (P_i,r_i). If $Q(P_i,r_i)=0$, Q interpolates (P_i,r_i). Furthermore, if $\mathcal{D}_{\mu\nu}^{(P_i,r_i)}(Q)=0$, $\forall\ \mu+\nu<m$, Q interpolates the point with a zero of multiplicity m. Eq. (12) defines the interpolation constraints for Q. Since there are $\binom{m+1}{2}$ pairs of (μ,ν) that satisfy $\mu+\nu<m$, interpolating all points of $\mathbf{P}$ imposes

$$\mathfrak{C}=n\binom{m+1}{2} \tag{13}$$

constraints on Q.

The $(1,\varpi)$-weighted degree of monomial $\phi_a z^b$ is $\deg_{1,\varpi}\phi_a z^b=-v_{P_\infty}(\phi_a)+\varpi b$, where ϖ is determined by the code dimension k. 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. We claim $\mathrm{ord}(\phi_{a_1}z^{b_1})<\mathrm{ord}(\phi_{a_2}z^{b_2})$, 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 $Q\in\mathcal{R}[z]$, 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 $\mathrm{lod}(Q)=\max\{\mathrm{ord}(\phi_a z^b)\mid Q_{ab}\neq 0\}$. Therefore, 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 decoding an (n,k) elliptic code, $\varpi=-v_{P_\infty}(\phi_{k-1})=k$. Note that when applying the re-encoding transform, if k is odd, $\varpi=1$; otherwise, $\varpi=2$. This will be discussed in Section IV.

Therefore, when decoding an (n,k) elliptic code, the intended interpolation polynomial $\mathcal{Q}$ satisfies the above constraints and being minimum under the above $(1,k)$-revlex order. The message polynomial f can be further decoded by finding z-roots of $\mathcal{Q}$, i.e., $\mathcal{Q}(x,y,f)=0$. This can be realized by the recursive coefficient search algorithm [35], [36]. The following Theorem shows the condition for a successful GS decoding.

Theorem 1 ([15]): Given the interpolation polynomial $\mathcal{Q}$ and a polynomial h in the form of (5), if

$$m(n-|\{i\mid h(P_i)\neq r_i,\ \forall i\}|)>\deg_{1,k}\mathcal{Q}, \tag{14}$$

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

Proof: Substituting h into (11) yields $\mathcal{Q}(x,y,h)=\sum_{\mu,\nu\in\mathbb{N}}\left(\mathcal{D}_{\mu\nu}^{(P_i,r_i)}(\mathcal{Q})\right)\psi_{P_i,\mu}(h-r_i)^\nu$. For point (P_i,r_i) that satisfies $h(P_i)=r_i$, $v_{P_i}(\mathcal{Q}(x,y,h))=\min\{v_{P_i}(\psi_{P_i,\mu})+v_{P_i}((h-r_i)^\nu)\mid\mathcal{D}_{\mu\nu}^{(P_i,r_i)}(\mathcal{Q})\neq 0\}\geq m$. Hence, $\mathcal{Q}(x,y,h)$ has a zero of order at least $m|\{i\mid h(P_i)=r_i,\ \forall i\}|=m(n-|\{i\mid h(P_i)\neq r_i,\ \forall i\}|)$ over the n interpolation points, i.e., $\sum_{i=0}^{n-1}v_{P_i}(\mathcal{Q}(x,y,h))\geq m(n-|\{i\mid h(P_i)\neq r_i,\ \forall i\}|)$. Since h is a polynomial in the form of (5), $\deg_{1,k}h=-v_{P_\infty}(h)\leq k$ and $-v_{P_\infty}(\mathcal{Q}(x,y,h))=\deg_{1,k}\mathcal{Q}(x,y,h)\leq\deg_{1,k}\mathcal{Q}(x,y,z)$. If $\deg_{1,k}\mathcal{Q}<m(n-|\{i\mid h(P_i)\neq r_i,\ \forall i\}|)$, the zero order of $\mathcal{Q}(x,y,h)$ is greater than its pole order. As a result, $\mathcal{Q}(x,y,h)=0$, or alteratively $(z-h)\mid\mathcal{Q}$. ∎

Therefore, if message polynomial f can be decoded, the GS algorithm corrects $|\{i\mid f(P_i)\neq r_i,\ \forall i\}|$ errors and this error-correction capability can be improved by increasing m. Given an $(n,\ k)$ elliptic code, the GS algorithm can correct at most [15]

$$\tau_{\mathrm{GS}}=n-\left\lfloor\sqrt{nk}\right\rfloor-1 \tag{15}$$

errors. Given an interpolation multiplicity m, let l and τ denote the corresponding maximum decoding output list size and the error-correction capability, respectively. Note that $\deg_z\mathcal{Q}\leq l$. The following Theorem characterizes l and τ.

Theorem 2 ([21]): For an (n,k) elliptic code, given an interpolation multiplicity m, the maximum decoding output list size

$$l=\left\lfloor\sqrt{\frac{nm(m+1)}{k}+\frac{1}{4}}-\frac{1}{2}\right\rfloor. \tag{16}$$

If $m(n-\tau)-kl\neq 1$, then the decoding can correct at most

$$\tau=n-\left\lfloor\frac{1}{m}+\frac{lk}{2m}+\frac{(m+1)n}{2(l+1)}\right\rfloor-1 \tag{17}$$

errors. Otherwise, it can correct at most

$$\tau=n-\frac{1+kl}{m} \tag{18}$$

errors.

Proof: Interpolating the n points with a multiplicity of m imposes $\mathfrak{C}$ interpolation constraints. Polynomial Q should contain at least $\mathfrak{C}+1$ coefficients so that the linear system will have a nonzero solution. Therefore,

$$l=\max\left\{l\mid\mathrm{ord}(z^l)\leq\mathfrak{C}\right\}.$$

In GS decoding of an (n,k) elliptic code, monomials are organized under the $(1,k)$-revlex order. Since $\mathrm{ord}(z^l)=\mathrm{ord}(z^{l-1})+|\{\phi_a z^b\mid(l-1)k<\deg_{1,k}\phi_a z^b\leq lk\}|=\mathrm{ord}(z^{l-1})+lk$, $\mathrm{ord}(z^l)=(1+2+\cdots+l)k=\frac{kl(l+1)}{2}$. Substituting it into the above equation leads to eq. (16).

Given $\underline{r}$, let $\tau=|\{i\mid f(P_i)\neq r_i,\forall i\}|$ denote the number of errors. Based on Theorem 1, if $m(n-\tau)>\deg_{1,k}Q$, $Q(x,y,f)=0$. Hence, any monomials $\phi_a z^b$ of Q should satisfy $\deg_{1,k}\phi_a z^b<m(n-\tau)$, and $\deg_{1,k}\phi_a<m(n-\tau)-kb$. Since $b=0,1,\ldots,l$, when $m(n-\tau)-kl\neq 1$, $|\{\phi_a z^b\mid\deg_{1,k}\phi_a z^b<m(n-\tau)\}|=\sum_{b=0}^{l}(m(n-\tau)-kb-1)=m(l+1)(n-\tau)-\frac{kl(l+1)}{2}-l-1$. Therefore, when

$$m(l+1)(n-\tau)-\frac{kl(l+1)}{2}-l-1>\mathfrak{C},$$

the linear system has a nonzero solution. Solving the above inequality leads to (17). Otherwise, when $m(n-\tau)-kl=1$, (18) can be reached. ∎

C. Basis Reduction Algorithms

We solve the interpolation problem using the BR technique, which consists of module basis construction and its reduction. Therefore, some prerequisites on module basis representation and its reduction need to be given. Consider an $\mathbb{F}_q[x]$-module $\mathcal{I}$ of dimension ρ. Any basis of $\mathcal{I}$ can be represented as a matrix in $\mathbb{F}_q[x]$, denoted as Ξ.

Definition 3: Let $\underline{\xi}=(\xi_0(x),\xi_1(x),\ldots,\xi_{\rho-1}(x))$ denote a vector over $\mathbb{F}_q[x]$, and $\underline{w}=(w_0,w_1,\ldots,w_{\rho-1})\in\mathbb{N}^\rho$, the $\underline{w}$-weighted degree of $\underline{\xi}$ is defined as

$$\deg_{\underline{w}}(\underline{\xi})=\max\{\deg_{1,\varpi}\xi_s(x)+w_s,\forall s\}. \tag{19}$$

The leading position of $\underline{\xi}$ is

$$\mathrm{LP}_{\underline{w}}(\underline{\xi})=\max\{s\mid\deg_{1,\varpi}\xi_s(x)+w_s=\deg_{\underline{w}}(\underline{\xi})\} \tag{20}$$

TABLE I
COMPLEXITY FOR COMPUTING A WEAK POPOV FORM OF $\Xi \in \mathbb{F}_q[x]^{\rho\times\rho}$

Algorithm	Complexity in O	Complexity in $O^{\sim}$
Lee-O'Sullivan [26], MS [28]	$\rho^2 \deg \Xi(\rho + \Delta(\Xi))$	$\rho^2 \deg \Xi \Delta(\Xi)$
Alekhnovich [29]	$\rho^\pi(\Upsilon(\Delta(\Xi))\log(\Delta(\Xi)) + \Upsilon(\deg \Xi))$	$\rho^\pi \Delta(\Xi)$
GJV [30]	$\rho^\pi \Upsilon(\deg \Xi)\log(\rho \deg \Xi)\log(\deg \det \Xi)$	$\rho^\pi \deg \Xi$

and the leading term of $\underline{\xi}$ is

$$\mathrm{LT}_{\underline{w}}(\underline{\xi}) = \xi_{\mathrm{LP}_{\underline{w}}(\underline{\xi})}(x). \tag{21}$$

Coefficient of the leading monomial of $\mathrm{LT}_{\underline{w}}(\underline{\xi})$ is called its leading coefficient, denoted as $\mathrm{LC}(\mathrm{LT}_{\underline{w}}(\underline{\xi}))$.

Definition 4: Given a matrix Ξ over $\mathbb{F}_q[x]$, let Ξ_t and $\Xi_{t,s}$ denote its row-t and its entry of row-t column-s, respectively, the $\underline{w}$-weighted degree of Ξ is

$$\deg_{\underline{w}} \Xi = \max\{\deg_{\underline{w}} \Xi_t, \forall t\}. \tag{22}$$

Note that in the un-weighted variants of the above definitions, $\deg_{\underline{w}}(\underline{\xi})$, $\mathrm{LP}_{\underline{w}}(\underline{\xi})$, $\mathrm{LT}_{\underline{w}}(\underline{\xi})$ and $\deg_{\underline{w}} \Xi$ are simplified into $\deg(\underline{\xi})$, $\mathrm{LP}(\underline{\xi})$, $\mathrm{LT}(\underline{\xi})$ and $\deg \Xi$, respectively.

Definition 5: Given a square matrix Ξ over $\mathbb{F}_q[x]$, it is in the weak Popov form if and only if $\mathrm{LP}(\Xi_t) \neq \mathrm{LP}(\Xi_{t'}), \forall t \neq t'$.

Presenting the interpolation module basis as a square matrix in $\mathbb{F}_q[x]$, the Gröbner basis is reached if it is reduced into the weak Popov form [25]. The desired interpolation polynomial $\mathcal{Q}$ is the minimum candidate of the Gröbner basis. There exist several algorithms for reducing matrix Ξ into the weak Popov form including the Lee-O'Sullivan algorithm [26], the MS algorithm [28], the Alekhnovich algorithm [29] and the GJV algorithm [30]. Their complexity mainly depend on $\deg \Xi$ and the orthogonality defect $\Delta(\Xi)$ that is defined as

$$\Delta(\Xi) = \mathrm{rowdeg}\Xi - \deg \det \Xi \leq \rho \deg \Xi,$$

where $\mathrm{rowdeg}\Xi = \sum_{t=0}^{\rho-1} \deg \Xi_t$ and $\det \Xi$ denotes the determinant of Ξ. Let $\Upsilon(u)$ denote the number of finite field arithmetic operations needed in multiplying two polynomials of degree at most u. For finite fields that support the fast Fourier transformation, $\Upsilon(u) = 18u\log u + O(u)$, while in general, $\Upsilon(u) = (18 + 72\log_3 2)u\log u \log\log u + O(u\log u)$ [37]. Table I summarizes the complexity of the existing basis reduction techniques, where $O^{\sim}$ denotes the complexity characterization O without considering the log factors, and π ($\pi \leq 3$) denotes the exponent for multiplication of $\mathbb{F}_q$ matrices. Table I shows the Alekhnovich and the GJV algorithms exhibit an asymptotically lower complexity than the other two algorithms. Note that normally $\rho \ll \Delta(\Xi)$ and $\rho \ll \deg \Xi$. However, they rely on the fast multiplication techniques which contribute to a large constant factor that is not reflected in the characterizations. This research has found out that the Lee-O'Sullivan and the MS algorithms are more efficient for handling codes of practical length. Therefore, we use the Lee-O'Sullivan algorithm for the basis reduction. Its complexity issue will be addressed again in Section V.

III. THE BR INTERPOLATION

The BR interpolation consists of module basis construction and its reduction. The former constructs a basis of $\mathbb{F}_q[x]$-module consisting of polynomials which satisfy the interpolation constraints. Gröbner basis of the module will be obtained by reducing this basis, where its minimum candidate is the desired polynomial $\mathcal{Q}$.

A. Prerequisites

Let $\mathcal{P} = E(\mathbb{F}_q) \setminus \{P_\infty\}$ denote the set of affine points on E. Let $\mathbb{A}$ denote the set of x-coordinates x_i of all elements in $\mathcal{P}$, and $\mathbb{B}_i = \{y \mid y^2 + \mathfrak{a}_1 x_i y + \mathfrak{a}_3 y = x_i^3 + \mathfrak{a}_2 x_i^2 + \mathfrak{a}_4 x_i + \mathfrak{a}_6\}$ as the set of y-coordinates defined by x_i. Note that $|\mathbb{A}| = n/2$. If $P_i = -P_{i'}$, $x_i = x_{i'}$, $\mathbb{B}_i = \mathbb{B}_{i'}$ and $|\mathbb{B}_i| = 2$, $\forall i$.

Theorem 3: Let a divisor $G = \sum_{i=0}^{n-1}[P_i]$, $\mathrm{div}(\mathcal{G}) = G - n[P_\infty]$, where

$$\mathcal{G} = \prod_{\alpha\in\mathbb{A}} (x - \alpha). \tag{23}$$

Proof: For each $\alpha \in \mathbb{A}$, there exist two different affine points P_i and $P_{i'}$ such that $\mathrm{div}(x-\alpha) = [P_i] + [P_{i'}] - 2[P_\infty]$. Therefore, $\mathrm{div}(\prod_{\alpha\in\mathbb{A}}(x-\alpha)) = G - n[P_\infty]$. ∎

The following Lagrange interpolation functions over $\mathbb{F}_q[x]$ are introduced.

Theorem 4: Given $\underline{r} = (r_0, r_1, \ldots, r_{n-1}) \in \mathbb{F}_q^n$, let

$$\mathcal{K}_{\underline{r}} = \sum_{i=0}^{n-1} r_i \mathcal{L}_i. \tag{24}$$

where

$$\mathcal{L}_i = \prod_{\alpha\in\mathbb{A}\setminus\{x_i\}} \frac{x-\alpha}{x_i-\alpha} \prod_{\beta\in\mathbb{B}_i\setminus\{y_i\}} \frac{y-\beta}{y_i-\beta}. \tag{25}$$

Note that $\mathcal{K}_{\underline{r}} \in \mathcal{R}$. It satisfies $\mathcal{K}_{\underline{r}}(P_i) = r_i$, $\mathcal{K}_{\underline{r}}(P_{i'}) = 0$, $\forall i' \neq i$ and $\deg_{1,k} \mathcal{K}_{\underline{r}} \leq n+1$. The complexity of computing $\mathcal{K}_{\underline{r}}$ is $O^{\sim}(n)$.

Proof: Since $|\mathbb{B}_i| = 2$, $\deg_y \mathcal{L}_i < 2$, i.e., $\mathcal{K}_{\underline{r}} \in \mathcal{R}$. Substituting P_i into $\mathcal{L}_i$ yields $\mathcal{L}_i(P_i) = 1$. For $i' \neq i$, $\mathcal{L}_i(P_{i'}) = 0$. Hence, $\mathcal{K}_{\underline{r}}(P_i) = r_i$, and $\mathcal{K}_{\underline{r}}(P_{i'}) = 0$, $\forall i' \neq i$. Since $\deg_{1,k} \mathcal{L}_i = n+1$, $\deg_{1,k} \mathcal{K}_{\underline{r}} \leq n+1$.

Let $P_i = (x_i, y_i)$ and $-P_i = (x_i, y_i^*)$, i.e., $\mathbb{B}_i = \{y_i, y_i^*\}$. $\mathcal{K}_{\underline{r}}$ of (24) can be written as

$$\mathcal{K}_{\underline{r}} = y \sum_{i=0}^{n-1} \frac{r_i}{y_i - y_i^*} \prod_{\alpha\in\mathbb{A}\setminus\{x_i\}} \frac{x-\alpha}{x_i-\alpha} - \sum_{i=0}^{n-1} \frac{r_i y_i^*}{y_i - y_i^*} \prod_{\alpha\in\mathbb{A}\setminus\{x_i\}} \frac{x-\alpha}{x_i-\alpha}.$$

Since $\frac{1}{y_i - y_i^*}$ and $\frac{y_i^*}{y_i - y_i^*}$ are pre-computed, computing $\mathcal{K}_{\underline{r}}$ can be seen as two univariate interpolation problems for $\frac{n}{2}$ points. Using the fast interpolation algorithm of [37], computing $\mathcal{K}_{\underline{r}}$ exhibits a complexity of $O^{\sim}(n)$. ∎

B. Basis Construction

Let $\mathcal{I}_{\mathbf{P}} \subset \mathcal{R}[z]_l$ denote a set of polynomials Q which have a zero of multiplicity m at the interpolation points of $\mathbf{P}$. Note that $\mathcal{I}_{\mathbf{P}}$ is an $\mathcal{R}$-module. Based on Theorems 3 and 4, for an interpolation point (P_i, r_i), $\mathcal{G}(P_i) = 0$ and $r_i - \mathcal{K}_{\underline{r}}(P_i) = 0$. Polynomials $\mathcal{G}$ and $z - \mathcal{K}_{\underline{r}}$ interpolate all points of $\mathbf{P}$.

To introduce the basis construction, the following Lemma is needed.

Lemma 5: Let $Q = \sum_{j=0}^{\rho} Q_{[j]} z^j \in \mathcal{I}_{\mathbf{P}}$ with $\deg_z Q = \rho < m$, $\mathcal{G}^{m-\rho} | Q_{[\rho]}$.

Proof: Since $Q \in \mathcal{I}_{\mathbf{P}}$, w.r.t. (P_i, r_i), it can be written as $Q = \sum_{\mu+\nu \geq m} Q_{\mu\nu} \Lambda_i^{\mu} (z - r_i)^{\nu}$. Since $\deg_z Q = \rho < m$ and $\nu \leq \rho$, $Q_{[\rho]} = \sum_{\mu \geq m-\rho} Q_{\mu\rho} \Lambda_i^{\mu}$, i.e., $\Lambda_i^{m-\rho} | Q_{[\rho]}$. For an affine point P_i without an order of two, Λ_i can be in the form of $\Lambda_i = x - x_i$. Therefore, $\mathcal{G}^{m-\rho} | Q_{[\rho]}$. ∎

Following a similar manner to [12] and [26], the following module generators are introduced.

Theorem 6: $\mathcal{I}_{\mathbf{P}}$ is generated as an $\mathcal{R}$-module by the following $l+1$ polynomials in $\mathcal{R}[z]_l$

$$\mathcal{H}^{(j)} = \mathcal{G}^{m-j}(z - \mathcal{K}_{\underline{r}})^j, \text{ if } 0 \leq j \leq m, \tag{26}$$

$$\mathcal{H}^{(j)} = z^{j-m}(z - \mathcal{K}_{\underline{r}})^m, \text{ if } m < j \leq l. \tag{27}$$

They are called the module generators.

Proof: Since both $\mathcal{G}$ and $z - \mathcal{K}_{\underline{r}}$ interpolate all points of $\mathbf{P}$, $\mathcal{H}^{(j)}$ has a zero of multiplicity at least m at the points, i.e., $\mathcal{H}^{(j)} \in \mathcal{I}_{\mathbf{P}}$. Note that for polynomial $\mathcal{H}^{(l)}$, $\mathcal{H}^{(l)}_{[l]} = 1$. Given a polynomial $Q \in \mathcal{I}_{\mathbf{P}}$, there exists $h_l = Q_{[l]}$ such that $Q^{(l-1)} = Q - h_l \mathcal{H}^{(l)}$, where $\deg_z Q^{(l-1)} \leq l - 1$. Similarly, eqs. (26) and (27) show that $\mathcal{H}^{(j)}_{[j]} = 1$ for $m \leq j \leq l - 1$. There exist polynomials $h_j \in \mathcal{R}$ such that $Q^{(m-1)} = Q^{(l-1)} - \sum_{j=m}^{l-1} h_j \mathcal{H}^{(j)}$ and $\deg_z Q^{(m-1)} \leq m-1$. Therefore, $Q^{(m-1)} \in \mathcal{I}_{\mathbf{P}}$. Based on Lemma 5, we have $\mathcal{G} | Q^{(m-1)}_{[m-1]}$. Since $\mathcal{H}^{(m-1)}_{[m-1]} = \mathcal{G}$, there exists $h_{m-1} = \frac{Q^{(m-1)}_{[m-1]}}{\mathcal{G}} \in \mathcal{R}$ such that $Q^{(m-2)} = Q^{(m-1)} - h_{m-1} \mathcal{H}^{(m-1)}$ with $\deg_z Q^{(m-2)} \leq m-2$. Following the same manner, when $1 \leq j \leq m-2$, $Q^{(j)}$ can be deduced using $\mathcal{H}^{(j)}$, until $Q^{(0)} = h_0 \mathcal{G}^m$ is reached. ∎

With the above module generators, the module basis can be generated as follows.

Theorem 7: $\mathcal{I}_{\mathbf{P}}$ is generated as an $\mathbb{F}_q[x]$-module by the following basis

$$\mathcal{M}_{\mathbf{P}} = \{M_t \mid M_t = y^{(t \bmod 2)} \mathcal{H}^{(\lfloor \frac{t}{2} \rfloor)}, 0 \leq t \leq 2l+1\}. \tag{28}$$

Proof: Based on Theorem 6, for each $Q \in \mathcal{I}_{\mathbf{P}}$, there exist $h_0, \ldots, h_l \in \mathcal{R}$ such that $Q = \sum_{j=0}^{l} h_j \mathcal{H}^{(j)}$. Since h_j can be written as $h_j = \mathfrak{h}_j^{(0)} + \mathfrak{h}_j^{(1)} y$, where $\mathfrak{h}_j^{(0)}, \mathfrak{h}_j^{(1)} \in \mathbb{F}_q[x]$, $Q = \sum_{j=0}^{l} \sum_{s=0}^{1} \mathfrak{h}_j^{(s)} (y^s \mathcal{H}^{(j)})$. ∎

Therefore, given $\mathbf{P}$, polynomials $\mathcal{G}$ and $\mathcal{K}_{\underline{r}}$ can be defined. They constitute the module generators of (26) and (27), which lead to the module basis construction of (28). The following Theorem characterizes complexity of this basis construction.

Theorem 8: Constructing the basis $\mathcal{M}_{\mathbf{P}}$ of (28) exhibits a complexity of $O^{\sim}(m^3 n)$.

Proof: Given an (n, k) elliptic code, $\mathcal{G}^j$ for $j = 1, 2, \ldots, m$ can be computed offline, and we also know the complexity of computing $\mathcal{K}_{\underline{r}}$ is $O^{\sim}(n)$. Eq. (24) can be written as $\mathcal{K}_{\underline{r}} = \kappa_0(x) + \kappa_1(x) y$, where $\deg_x \kappa_i(x) < \frac{n}{2}$ for $i = 0, 1$. Therefore, computing $\mathcal{K}_{\underline{r}}, \mathcal{K}^2_{\underline{r}}, \ldots, \mathcal{K}^m_{\underline{r}}$ exhibits a complexity of $O^{\sim}(m^2 n)$. Also note that they need to be rationalized into $\mathcal{R}$ of (3). Its complexity would be $O(mn)$. Subsequently, computing the module generators $\mathcal{H}^{(j)}$ has a complexity of $O^{\sim}(m^3 n)$. ∎

C. Basis Reduction

Basis $\mathcal{M}_{\mathbf{P}}$ will be further reduced, yielding the Gröbner basis $\mathcal{M}'_{\mathbf{P}}$ that contains the interpolation polynomial $\mathcal{Q}$.

Since $\mathcal{R}[z]_l$ is a free module over $\mathbb{F}_q[x]$ with a rank of $2(l+1)$, it has a free basis of $\{1, y, z, yz, \ldots, z^l, yz^l\}$. $\mathcal{I}_{\mathbf{P}}$ is a submodule of $\mathcal{R}[z]_l$. For each $Q \in \mathcal{I}_{\mathbf{P}}$, it can be expressed as $Q = Q^{(0)} + Q^{(1)} y + \cdots + Q^{(2l+1)} y z^l$, where $Q^{(0)}, Q^{(1)}, \ldots, Q^{(2l+1)} \in \mathbb{F}_q[x]$. They can also be written as

$$Q = (Q^{(0)}, Q^{(1)}, \ldots, Q^{(2l+1)})(1, y, \ldots, yz^l)^T. \tag{29}$$

Similarly, the basis polynomial can be written as $M_t = (M_t^{(0)}, M_t^{(1)}, \cdots, M_t^{(2l+1)})(1, y, \ldots, yz^l)^T$. The basis $\mathcal{M}_{\mathbf{P}}$ can be represented as a matrix $\mathcal{V}$ in $\mathbb{F}_q[x]$ by letting

$$\mathcal{V}_t = (M_t^{(0)}, M_t^{(1)}, \ldots, M_t^{(2l+1)}), \tag{30}$$

where $\mathcal{V}_{t,s} = M_t^{(s)}$ and $s = 0, 1, \ldots, 2l+1$. Therefore, $M_t = \mathcal{V}_t \cdot (1, y, \ldots, yz^l)^T$. Based on eqs. (26)-(28), it can be seen that $\mathcal{V}$ is a lower triangular matrix. By letting $w_s = k\lfloor \frac{s}{2} \rfloor + 3(s \bmod 2)$, we have $\deg_{\underline{w}} \mathcal{V}_t = \deg_{1,k} M_t$.

The Lee-O'Sullivan basis reduction algorithm will be applied to reduce $\mathcal{M}_{\mathbf{P}}$ into the desired Gröbner basis. For each row $\mathcal{V}_t$, $\mathrm{LP}_{\underline{w}}(\mathcal{V}_t)$ can be determined. Row operations of $\mathcal{V}$ will be performed until $\mathrm{LP}_{\underline{w}}(\mathcal{V}_t) = t$. Since $M_0 = \mathcal{G}^m$ and $M_1 = \mathcal{G}^m y$, $\mathrm{LP}_{\underline{w}}(\mathcal{V}_0) = 0$ and $\mathrm{LP}_{\underline{w}}(\mathcal{V}_1) = 1$. The row operation can start from $\mathcal{V}_2$. In general, if $\mathrm{LP}_{\underline{w}}(\mathcal{V}_t) = t$, $\mathcal{V}_t$ does not need to be modified. Row $\mathcal{V}_{t+1}$ will be further processed. If $\mathrm{LP}_{\underline{w}}(\mathcal{V}_t) = t'$ and $t \neq t'$, we let $u = \deg_{\underline{w}} \mathcal{V}_t - \deg_{\underline{w}} \mathcal{V}_{t'}$ and $v = \mathrm{LC}(\mathrm{LT}_{\underline{w}}(\mathcal{V}_t))\mathrm{LC}(\mathrm{LT}_{\underline{w}}(\mathcal{V}_{t'}))^{-1}$. If $u \geq 0$, $\mathcal{V}_t$ will be updated by

$$\mathcal{V}'_t = \mathcal{V}_t - v x^u \mathcal{V}_{t'}. \tag{31}$$

Otherwise, $\mathcal{V}_{t'}$ and $\mathcal{V}_t$ will be updated by

$$\mathcal{V}'_{t'} = \mathcal{V}_t \tag{32}$$

and

$$\mathcal{V}'_t = x^{-u} \mathcal{V}_t - v \mathcal{V}_{t'}. \tag{33}$$

Note that the update of $\mathcal{V}_t$ only involves the first $t-1$ rows of $\mathcal{V}$, which does not change the leading position of those rows. Finally, the updated matrix $\mathcal{V}'$ satisfies $\mathrm{LP}_{\underline{w}}(\mathcal{V}'_t) = t$, $\forall t$. The updated polynomials can be further obtained by

$$M'_t = \mathcal{V}'_t \cdot (1, y, \ldots, yz^l)^T, \tag{34}$$

which form the Gröbner basis $\mathcal{M}'_{\mathbf{P}}$ of $\mathcal{I}_{\mathbf{P}}$.

The minimum candidate of $\mathcal{M}'_{\mathbf{P}}$ is chosen as the interpolation polynomial $\mathcal{Q}$.

Algorithm 1 The BR Interpolation

Input: $\underline{r}$ and m;
Output: $\mathcal{Q}$;

1: Initialize $\mathcal{M}_{\mathbf{P}}$ as in (26)-(28);
2: Represent $\mathcal{M}_{\mathbf{P}}$ as matrix $\mathcal{V}$ over $\mathbb{F}_q[x]$ as in (30);
3: Reduce $\mathcal{V}$ into a weak Popov form matrix $\mathcal{V}'$ using Lee-O'Sullivan's algorithm;
4: Demap the matrix $\mathcal{V}'$ to $\mathcal{M}'_{\mathbf{P}}$ as in (34);
5: Pick up the minimum candidate from $\mathcal{M}'_{\mathbf{P}}$ as $\mathcal{Q}$.

Summarizing the above description, the BR interpolation algorithm for GS decoding of elliptic codes can be stated as the follows.

The complexity and validity of the above interpolation algorithm is further characterized as the follows.

Theorem 9: Algorithm 1 is correct. Given basis $\mathcal{M}_P$, they can be presented as a matrix $\mathcal{V} \in \mathbb{F}_q[x]^{2(l+1)\times 2(l+1)}$. The complexity of Algorithm 1 is $O(l^3m^2\ n(n-k))$.

Proof: Based on Theorem 7, $\mathcal{M}_{\mathbf{P}}$ generates the $\mathbb{F}_q[x]$ module $\mathcal{I}_{\mathbf{P}}$. Therefore, $\mathcal{M}'_{\mathbf{P}}$ obtained by $\mathbb{F}_q[x]$ row operations is still a basis of $\mathcal{I}_{\mathbf{P}}$. For $Q \in \mathcal{I}_{\mathbf{P}}$, by (29), let $Q = \underline{Q}(1, y, \ldots, yz^l)^T$, where $\underline{Q} = (Q^{(0)}, Q^{(1)}, \ldots, Q^{(2l+1)})$. Since the updated matrix $\mathcal{V}'$ satisfies $\mathrm{LP}_{\underline{w}}(\mathcal{V}'_t) = t$, $\forall t$, there exists t such that $\mathrm{LP}_{\underline{w}}\underline{Q} = \mathrm{LP}_{\underline{w}}\mathcal{V}'_t$ and $\mathrm{LT}_{\underline{w}}\mathcal{V}'_t \mid \mathrm{LT}_{\underline{w}}\underline{Q}$. Therefore, $\mathcal{M}'_{\mathbf{P}}$ is a Gröbner basis of $\mathcal{I}_{\mathbf{P}}$, and the minimum polynomial of $\mathcal{I}_{\mathbf{P}}$ is also the minimum element of $\mathcal{M}'_{\mathbf{P}}$. Therefore, Algorithm 1 is correct.

Based on Theorem 8, complexity of constructing the basis $\mathcal{M}_{\mathbf{P}}$ is $O^{\sim}(m^3n)$. Since $\deg_{1,k}\mathcal{G} = n$ and $\deg_{1,k}\mathcal{K}_{\underline{r}} \le n+1$, $\deg_{\underline{w}}\mathcal{V} \le \deg_{\underline{w}}\mathcal{V}_{2l+1,2(l-m)+1} \le m(n+1) + k(l-m) + 3$. If $t = 0, 1, \ldots, 2m+1$, $\deg_{\underline{w}}\mathcal{V}_t = mn + 3t - 5\lfloor\frac{t}{2}\rfloor$ and $\deg_{\underline{w}}\mathcal{V}_{t,t} = n(m - \lfloor\frac{t}{2}\rfloor) + w_t$. If $t = 2m+2, 2m+3, \ldots, 2l+1$, $\deg_{\underline{w}}\mathcal{V}_t = m(n+1) + k\lceil\frac{t-2m-1}{2}\rceil + 3(t \bmod 2)$ and $\deg_{\underline{w}}\mathcal{V}_{t,t} = w_t$. Therefore, $\mathrm{rowdeg}\mathcal{V} = \sum_{t=0}^{2l+1}\deg_{\underline{w}}\mathcal{V}_t \approx 2lmn + k(l-m)^2 + 3l$ and $\deg\det\mathcal{V} = \sum_{t=0}^{2l+1}\deg_{\underline{w}}\mathcal{V}_{t,t} \approx m^2n + l^2k + 3l$. Hence, $\Delta(\mathcal{V}) \approx ml(n-k)$. Therefore, based on Table I, reducing $\mathcal{V}$ into a Gröbner basis requires at most

$$(2l+2)^2\deg_{\underline{w}}\mathcal{V}\Delta(\mathcal{V}) \approx 4\ l^3\ m(n-k)(mn+kl-km)$$

multiplications. ∎

Based on Theorem 1, message polynomial f can be further decoded by finding z-roots of $\mathcal{Q}$, i.e., $\mathcal{Q}(x, y, f) = 0$, which can be realized by the recursive coefficient search algorithm [35], [36]. Its complexity is quadratic in n. Although it has the same asymptotic complexity as the interpolation process, in practice, the root-finding complexity will be marginalized by interpolation. Based on the root-finding algorithm for RS codes [38], Alekhnovich provided a faster approach that yields a quasi-linear complexity in n [29]. By using the concept of power series, Beelen and Høholdt further provided the root-finding algorithm for decoding AG codes [39]. Nielsen and Beelen also generalized Alekhnovich's work to decode Hermitian codes [12], which yields a root-finding complexity that is sub-quadratic in n. They can be considered for decoding of elliptic codes. But the same asymptotic complexity (sub-quadratic in n) cannot be straightforwardly reached.

IV. The Re-Encoding Transformed Interpolation

Re-encoding transform helps reduce the BR interpolation complexity by reducing x-degree of the basis entries. The following introduces the transform of interpolation points and the subsequent basis construction. The basis reduction remains unchanged.

A. Transform of the Interpolation Points

First, ε interpolation points in $\mathbf{P}$ will be chosen as the re-encoding points for the transform. Let Γ denote the index set of these re-encoding points, and Γ^c denote the index set of the remaining points. With a received word $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$, the re-encoding polynomial $\mathcal{K}_\Gamma \in \mathfrak{L}(k[P_\infty])$ needs to be constructed for the transform.

Given E, let $\mathcal{P}_\Gamma = \{P_i \mid i \in \Gamma\}$. For the points of $\mathcal{P}_\Gamma$, we further denote $\mathbb{A}_\Gamma = \{x_i \mid i \in \Gamma\}$, $\mathbb{A}_{\Gamma^c} = \mathbb{A} \setminus \mathbb{A}_\Gamma$ and $\mathbb{B}_\Gamma^{(i)} = \{y \mid (x_i, y) \in \mathcal{P}_\Gamma\}$.

The re-encoding polynomial $\mathcal{K}_\Gamma$ can be defined as follows.

Theorem 10: Let

$$\mathcal{K}_\Gamma = \sum_{i\in\Gamma} r_i\mathcal{L}_\Gamma^{(i)}, \tag{35}$$

where

$$\mathcal{L}_\Gamma^{(i)} = \prod_{\alpha\in\mathbb{A}_\Gamma\setminus\{x_i\}}\frac{x-\alpha}{x_i-\alpha}\prod_{\beta\in\mathbb{B}_\Gamma^{(i)}\setminus\{y_i\}}\frac{y-\beta}{y_i-\beta}. \tag{36}$$

It satisfies $\mathcal{K}_\Gamma(P_i) = r_i$, $\forall i \in \Gamma$. Furthermore, let $\sigma = |\{i \mid |\mathbb{B}_\Gamma^{(i)}| = 1, i \in \Gamma\}|$, $\deg_{1,k}\mathcal{K}_\Gamma = \varepsilon + \sigma + 1$.

Proof: Eq. (36) ensures $\mathcal{L}_\Gamma^{(i)}(P_i) = 1$, and $\mathcal{L}_\Gamma^{(i)}(P_{i'}) = 0$, $\forall i' \neq i$ and $i \in \Gamma$. Therefore, $\mathcal{K}_\Gamma(P_i) = r_i$, $\forall i \in \Gamma$. If $|\mathbb{B}_\Gamma^{(i)}| = 1$, $\deg_{1,k}\mathcal{L}_\Gamma^{(i)} = \varepsilon+\sigma-2$. Otherwise, $\deg_{1,k}\mathcal{L}_\Gamma^{(i)} = \varepsilon + \sigma + 1$. Therefore, $\deg_{1,k}\mathcal{K}_\Gamma = \varepsilon + \sigma + 1$. ∎

With the re-encoding polynomial, a new codeword can be generated by

$$\begin{aligned}\underline{c}' &= (\mathcal{K}_\Gamma(P_0), \mathcal{K}_\Gamma(P_1), \ldots, \mathcal{K}_\Gamma(P_{n-1}))\\ &= (c'_0, c'_1, \ldots, c'_{n-1}).\end{aligned} \tag{37}$$

The received word $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$ can therefore be transformed by

$$\underline{r}' = \underline{r} - \underline{c}', \tag{38}$$

where $r'_i = r_i - c'_i$. To simplify the description of the re-encoding transformed interpolation, the following encoding point rearrangement is needed.

Proposition 11: The evaluation encoding order of (6) is rearranged such that the affine points with the same x-coordinate are adjacent to each other.

Therefore, without loss of generality, the first ε interpolation points are chosen as the re-encoding points, i.e., $\Gamma = \{0, 1, \ldots, \varepsilon-1\}$ and $\Gamma^c = \{\varepsilon, \varepsilon+1, \ldots, n-1\}$. Based on (38), the transformed received word becomes

$$\underline{r}' = (0, \ldots, 0, r'_\varepsilon, \ldots, r'_{n-1}). \tag{39}$$

Consequently, the set of interpolation points $\mathbf{P}$ is transformed into

$$\mathbf{P}' = \{(P_0, 0), \ldots, (P_{\varepsilon-1}, 0), (P_\varepsilon, r'_\varepsilon), \ldots, (P_{n-1}, r'_{n-1})\}. \tag{40}$$

This enables the basis polynomials yield a common factor, which can be removed before the basis reduction. Note that $\mathcal{K}_\Gamma \in \mathfrak{L}(k[P_\infty])$, i.e., $\varepsilon+\sigma+1 \leq k$, which limits the number of re-encoding points. In order to maximize the number of points (in $\mathbf{P}'$) with coordinate being zero, σ should be as small as possible.

Corollary 12: Given $\mathcal{C}_E(G, k[P_\infty])$ and $\sigma \leq 1$, if k is odd, $\varepsilon = k-1$; otherwise, $\varepsilon = k-2$.

Proof: Recalling Theorem 10, if ε is even, i.e., $\sigma = 0$, $\deg_{1,k} \mathcal{K}_\Gamma = \varepsilon+1$. Otherwise, $\deg_{1,k} \mathcal{K}_\Gamma = \varepsilon+2$. To construct an (n, k) elliptic code, we need to maintain $\deg_{1,k} \mathcal{K}_\Gamma \leq k$. Hence, if k is odd, $\varepsilon = k-1$. Otherwise, $\varepsilon = k-2$. ∎

B. Basis Construction

Let $\mathcal{I}_{\mathbf{P}'} \subset \mathcal{R}[z]_l$ denote a set of Q which have a zero of multiplicity m at the transformed interpolation points in $\mathbf{P}'$. The following theorem describes the relationship between $\mathcal{I}_{\mathbf{P}'}$ and $\mathcal{I}_{\mathbf{P}}$.

Theorem 13: $\mathcal{Q}$ is the interpolation polynomial in $\mathcal{I}_{\mathbf{P}}$ if and only if $\widetilde{\mathcal{Q}} = \mathcal{Q}(x, y, z + \mathcal{K}_\Gamma)$ is also one in $\mathcal{I}_{\mathbf{P}'}$.

Proof: Based on Theorem 7, if $\mathcal{Q} \in \mathcal{I}_{\mathbf{P}}$, $\mathcal{Q}(x, y, z + \mathcal{K}_\Gamma) = \sum_{j=0}^{l} \sum_{s=0}^{1} \mathfrak{h}_j^{(s)} (y^s \widetilde{\mathcal{H}}^{(j)})$, where $\mathfrak{h}_j^{(s)} \in \mathbb{F}_q[x]$ and $\widetilde{\mathcal{H}}^{(j)} = \mathcal{H}^{(j)}(x, y, z + \mathcal{K}_\Gamma)$. Based on eqs. (26) and (27),

$$\widetilde{\mathcal{H}}^{(j)} = \mathcal{G}^{m-j}(z + \mathcal{K}_\Gamma - \mathcal{K}_{\underline{r}})^j, \text{ if } 0 \leq j \leq m,$$
$$\widetilde{\mathcal{H}}^{(j)} = (z + \mathcal{K}_\Gamma)^{j-m}(z + \mathcal{K}_\Gamma - \mathcal{K}_{\underline{r}})^m, \text{ if } m < j \leq l.$$

Since $\mathcal{K}_{\underline{r}}(P_i) - \mathcal{K}_\Gamma(P_i) = r_i - c'_i = r'_i$, $\widetilde{\mathcal{H}}^{(j)}$ has a zero of multiplicity m at the transformed interpolation points, i.e., $\widetilde{\mathcal{H}}^{(j)} \in \mathcal{I}_{\mathbf{P}'}$. Therefore, $\widetilde{\mathcal{Q}} \in \mathcal{I}_{\mathbf{P}'}$. If $\widetilde{\mathcal{Q}} \in \mathcal{I}_{\mathbf{P}'}$, $\mathcal{Q} \in \mathcal{I}_{\mathbf{P}}$.

Since $\deg_{1,k} \mathcal{K}_\Gamma \leq k$, $\deg_{1,k} z = \deg_{1,k}(z + \mathcal{K}_\Gamma) = k$, and $\deg_{1,k} \widetilde{\mathcal{Q}} = \deg_{1,k} \mathcal{Q}$. If $\mathcal{Q}$ is the interpolation polynomial in $\mathcal{I}_{\mathbf{P}}$, $\widetilde{\mathcal{Q}} \in \mathcal{I}_{\mathbf{P}'}$. If there exists $\widetilde{\mathcal{Q}}' \in \mathcal{I}_{\mathbf{P}'}$ which satisfies $\deg_{1,k} \widetilde{\mathcal{Q}}' < \deg_{1,k} \widetilde{\mathcal{Q}}$, $\mathcal{Q}' = \widetilde{\mathcal{Q}}'(x, y, z - \mathcal{K}_\Gamma) \in \mathcal{I}_{\mathbf{P}}$ and it satisfies $\deg_{1,k} \mathcal{Q}' = \deg_{1,k} \widetilde{\mathcal{Q}}'$. This leads to $\deg_{1,k} \mathcal{Q}' < \deg_{1,k} \mathcal{Q}$. It contradicts $\mathcal{Q}$ being the minimum polynomial of $\mathcal{I}_{\mathbf{P}}$. Therefore, $\widetilde{\mathcal{Q}}$ is the minimum interpolation polynomial in $\mathcal{I}_{\mathbf{P}'}$, and vice versa. ∎

Therefore, the interpolation polynomial $\mathcal{Q}$ can be obtained by $\mathcal{Q} = \widetilde{\mathcal{Q}}(x, y, z - \mathcal{K}_\Gamma)$. To describe the basis construction in the case of re-encoding transform, the following notations are needed. Note that polynomial $\mathcal{G}$ of (23) can be factorized into

$$\mathcal{G} = \mathcal{G}_\Gamma \mathcal{G}_{\Gamma^c}, \tag{41}$$

where

$$\mathcal{G}_\Gamma = \prod_{\alpha \in \mathbb{A}_\Gamma} (x - \alpha) \tag{42}$$

and

$$\mathcal{G}_{\Gamma^c} = \prod_{\alpha \in \mathbb{A}_{\Gamma^c}} (x - \alpha). \tag{43}$$

Based on $\underline{r}'$ (eq. (39)) and $\mathbf{P}'$ (eq. (40)), $\mathcal{K}_{\underline{r}'}$ of (24) can be defined as

$$\begin{aligned}\mathcal{K}_{\underline{r}'} &= \sum_{i=0}^{\varepsilon-1} 0 \cdot \mathcal{L}_i + \sum_{i=\varepsilon}^{n-1} r'_i \mathcal{L}_i \\ &= \mathcal{G}_\Gamma \sum_{i=\varepsilon}^{n-1} \frac{r'_i}{\mathcal{G}_\Gamma(x_i)} \prod_{\alpha \in \mathbb{A}_{\Gamma^c} \setminus \{x_i\}} \frac{x - \alpha}{x_i - \alpha} \prod_{\beta \in \mathbb{B}_i \setminus \{y_i\}} \frac{y - \beta}{y_i - \beta}.\end{aligned}$$

Therefore, $\mathcal{G}_\Gamma$ becomes the GCD for both $\mathcal{G}$ and $\mathcal{K}_{\underline{r}'}$. Let us rewrite $\mathcal{K}_{\underline{r}'}$ as

$$\mathcal{K}_{\underline{r}'} = \mathcal{G}_\Gamma \mathcal{K}_{\Gamma^c}, \tag{44}$$

where

$$\mathcal{K}_{\Gamma^c} = \sum_{i=\varepsilon}^{n-1} r_i^* \mathcal{L}_{\Gamma^c}^{(i)}, \tag{45}$$

$$r_i^* = \frac{r'_i}{\mathcal{G}_\Gamma(x_i)}, \tag{46}$$

and

$$\mathcal{L}_{\Gamma^c}^{(i)} = \prod_{\alpha \in \mathbb{A}_{\Gamma^c} \setminus \{x_i\}} \frac{x - \alpha}{x_i - \alpha} \prod_{\beta \in \mathbb{B}_i \setminus \{y_i\}} \frac{y - \beta}{y_i - \beta}. \tag{47}$$

For the transformed interpolation points of $\mathbf{P}'$, its subset $\{(P_i, r'_i) \mid i \in \Gamma^c\}$ can be further transformed into

$$\mathbf{P}^* = \{(P_\varepsilon, r_\varepsilon^*), \ldots, (P_{n-1}, r_{n-1}^*)\}. \tag{48}$$

Note that for (P_i, r'_i) where $i \in \Gamma$, since $r'_i = 0$, $(P_i, r_i^*) = (P_i, r'_i)$. Similarly, let $\mathcal{I}_{\mathbf{P}^*} \subset \mathcal{R}[z]_l$ denote a set of some Q^* which have a zero of multiplicity m at the points of $\mathbf{P}^*$. The following Lemma first reveals the property of the polynomials in $\mathcal{I}_{\mathbf{P}'}$.

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

Proof: Based on Theorem 6, Q can be written as $Q = \sum_{j=0}^{l} h_j \mathcal{H}^{(j)}$, which can be further elaborated as

$$\begin{aligned}Q &= \sum_{j=0}^{m} h_j \mathcal{G}^{m-j} \left(z - \mathcal{K}_{\underline{r}'}\right)^j + \sum_{j=m+1}^{l} h_j z^{j-m} \left(z - \mathcal{K}_{\underline{r}'}\right)^m \\ &= \sum_{j=0}^{m} h_j \mathcal{G}^{m-j} \left(z - \mathcal{G}_\Gamma \mathcal{K}_{\Gamma^c}\right)^j \\ &\quad + \sum_{j=m+1}^{l} h_j z^{j-m} \left(z - \mathcal{G}_\Gamma \mathcal{K}_{\Gamma^c}\right)^m \\ &= \mathcal{G}_\Gamma^m \Bigg(\sum_{j=0}^{m} h_j \mathcal{G}_{\Gamma^c}^{m-j} \left(\frac{z}{\mathcal{G}_\Gamma} - \mathcal{K}_{\Gamma^c}\right)^j \\ &\quad + \sum_{j=m+1}^{l} h_j \mathcal{G}_\Gamma^{j-m} \left(\frac{z}{\mathcal{G}_\Gamma}\right)^{j-m} \left(\frac{z}{\mathcal{G}_\Gamma} - \mathcal{K}_{\Gamma^c}\right)^m \Bigg).\end{aligned}$$

Therefore,

$$\begin{aligned}Q(x, y, z\mathcal{G}_\Gamma) = \mathcal{G}_\Gamma^m \Big(&\sum_{t=0}^{m} h_j \mathcal{G}_{\Gamma^c}^{m-j} \left(z - \mathcal{K}_{\Gamma^c}\right)^j \\ &+ \sum_{j=m+1}^{l} h_j \mathcal{G}_\Gamma^{j-m} z^{j-m} \left(z - \mathcal{K}_{\Gamma^c}\right)^m \Big).\end{aligned} \tag{49}$$

Thus, $\mathcal{G}_\Gamma^m \mid Q(x, y, z\mathcal{G}_\Gamma)$. ∎

Armed with the above lemma, the following bijective mapping between the polynomials of $\mathcal{I}_{\mathbf{P}'}$ and those of $\mathcal{I}_{\mathbf{P}^*}$ can be established

$$\begin{aligned}&\Psi : \mathcal{I}_{\mathbf{P}'} \to \mathcal{I}_{\mathbf{P}^*}\\ &Q(x,y,z) \mapsto Q^*(x,y,z) = \mathcal{G}_\Gamma^{-m} Q(x,y,z\mathcal{G}_\Gamma),\end{aligned} \tag{50}$$

where Ψ is an $\mathbb{F}_q[x]$-module isomorphism between $\mathcal{I}_{\mathbf{P}'}$ and $\mathcal{I}_{\mathbf{P}^*}$. Therefore, $\mathcal{I}_{\mathbf{P}^*}$ can be generated as an $\mathbb{F}_q[x]$-module by the following $2l+2$ polynomials

$$\mathcal{M}_{\mathbf{P}^*} = \{M_t^* \mid M_t^* = y^{(t \bmod 2)} \mathcal{H}_\Gamma^{(\lfloor \frac{t}{2} \rfloor)}, 0 \le t \le 2l+1\}, \tag{51}$$

where

$$\mathcal{H}_\Gamma^{(j)} = \mathcal{G}_{\Gamma^c}^{m-j}(z - \mathcal{K}_{\Gamma^c})^j, \text{ if } 0 \le j \le m, \tag{52}$$

$$\mathcal{H}_\Gamma^{(j)} = (\mathcal{G}_\Gamma z)^{j-m}(z - \mathcal{K}_{\Gamma^c})^m, \text{ if } m < j \le l. \tag{53}$$

Polynomials M_t^* of (51) have a zero of multiplicity m at the points of $\mathbf{P}^*$. Comparing with the polynomials of $\mathcal{M}_{\mathbf{P}}$ (see eqs. (26)-(28)), polynomials of $\mathcal{M}_{\mathbf{P}^*}$ have lower x-degrees. This can reduce the basis reduction complexity. If $Q^* \in \mathcal{I}_{\mathbf{P}^*}$, it can be written as an $\mathbb{F}_q[x]$-linear combination of the above polynomials, i.e.,

$$Q^* = \sum_{t=0}^{2l+1} \mathfrak{h}_t M_t^*, \tag{54}$$

where $\mathfrak{h}_t \in \mathbb{F}_q[x]$. We further prove that constructing $\mathcal{I}_{\mathbf{P}^*}$ will be sufficient to obtain the interpolation polynomial $\mathcal{Q}$.

Lemma 15: If $Q \in \mathcal{I}_{\mathbf{P}'}$, $\deg_{1,k} Q = \deg_{1,\varpi} \mathcal{G}_\Gamma^m + \deg_{1,\varpi} Q^*$, where $\varpi = 1$ if k is odd, otherwise, $\varpi = 2$.

Proof: Based on eq. (50), we know $Q = \mathcal{G}_\Gamma^m Q^*(x, y, \frac{z}{\mathcal{G}_\Gamma})$. Let $Q^* = \sum_{j=0}^{l} Q_{[j]}^* z^j$, where $Q_{[j]}^* \in \mathcal{R}$, it follows that

$$\begin{aligned}\deg_{1,k} Q = &\ \deg_{1,k} \mathcal{G}_\Gamma^m + \deg_{1,k} Q^*\left(x, y, \frac{z}{\mathcal{G}_\Gamma}\right)\\ = &\ \deg_{1,k} \mathcal{G}_\Gamma^m + \deg_{1,k} \sum_{j=0}^{l} Q_{[j]}^* \left(\frac{z}{\mathcal{G}_\Gamma}\right)^j\\ = &\ \deg_{1,k} \mathcal{G}_\Gamma^m + \max_{0\le j\le l}\{\deg_{1,k} Q_{[j]}^*\\ &+ j(k - \deg_{1,k} \mathcal{G}_\Gamma)\}.\end{aligned}$$

Therefore, $\deg_{1,k} Q = \deg_{1,\varpi} \mathcal{G}_\Gamma^m + \deg_{1,\varpi} Q^*$, where $\varpi = k - \deg_{1,k} \mathcal{G}_\Gamma$. Based on Corollary 12 and (42), if k is odd, $\deg_{1,k} \mathcal{G}_\Gamma = k-1$; otherwise, $\deg_{1,k} \mathcal{G}_\Gamma = k-2$. ∎

Since $M_t^* = M_t^{*(0)} + M_t^{*(1)} y + \cdots + M_t^{*(2l+1)} y z^l$, polynomials of $\mathcal{M}_{\mathbf{P}^*}$ can be represented as a matrix $\mathcal{V}^* \in \mathbb{F}_q[x]^{2(l+1)\times 2(l+1)}$ by letting

$$\mathcal{V}_t^* = (M_t^{*(0)}, M_t^{*(1)}, \ldots, M_t^{*(2l+1)}). \tag{55}$$

Hence, $M_t^* = \mathcal{V}_t^* \cdot (1, y, \ldots, yz^l)^T$ and $\mathcal{V}_{t,s}^* = M_t^{*(s)}$. Based on Lemma 15, if k is odd, $\varpi = 1$, $w_s = \lfloor \frac{s}{2} \rfloor + 3(s \bmod 2)$ for $s = 0, 1, \ldots, 2l+1$ and $\deg \mathcal{V}_t^* = \deg_{1,1} M_t^*$. Otherwise, $\varpi = 2$, $w_s = 2\lfloor \frac{s}{2} \rfloor + 3(s \bmod 2)$ and $\deg \mathcal{V}_t^* = \deg_{1,2} M_t^*$. We can further apply the row reduction described in Section III.C to reduce $\mathcal{M}_{\mathbf{P}^*}$ into a Gröbner basis w.r.t. the $(1, \varpi)$-revlex order, denoted as $\mathcal{M}'_{\mathbf{P}^*}$.

Theorem 16: Given $\mathcal{Q}^*$ as the minimum polynomial in $\mathcal{I}_{\mathbf{P}^*}$, the interpolation polynomial $\mathcal{Q}$ can be obtained by

$$\mathcal{Q} = \mathcal{G}_\Gamma^m \mathcal{Q}^*\left(x, y, \frac{z - \mathcal{K}_\Gamma}{\mathcal{G}_\Gamma}\right). \tag{56}$$

Proof: Based on Theorem 13, if $\widetilde{\mathcal{Q}}$ is the interpolation polynomial in $\mathcal{I}_{\mathbf{P}'}$, $\mathcal{Q} = \widetilde{\mathcal{Q}}(x, y, z - \mathcal{K}_\Gamma)$ will be the interpolation polynomial in $\mathcal{I}_{\mathbf{P}}$. Therefore, we only need to prove that $\widetilde{\mathcal{Q}} = \mathcal{G}_\Gamma^m \mathcal{Q}^*\left(x, y, \frac{z}{\mathcal{G}_\Gamma}\right)$ is the interpolation polynomial in $\mathcal{I}_{\mathbf{P}'}$. The mapping of (50) shows that if $\mathcal{Q}^* \in \mathcal{I}_{\mathbf{P}^*}$, $\widetilde{\mathcal{Q}} \in \mathcal{I}_{\mathbf{P}'}$. If there exists a polynomial $\widetilde{\mathcal{Q}}' \in \mathcal{I}_{\mathbf{P}'}$ that satisfies $\deg_{1,k} \widetilde{\mathcal{Q}}' < \deg_{1,k} \widetilde{\mathcal{Q}}$, $\mathcal{Q}^{*\prime} = \mathcal{G}_\Gamma^{-m} \widetilde{\mathcal{Q}}'(x, y, z\mathcal{G}_\Gamma)$. Based on Lemma 15, $\deg_{1,\varpi} \mathcal{G}_\Gamma^m + \deg_{1,\varpi} \mathcal{Q}^{*\prime} < \deg_{1,\varpi} \mathcal{G}_\Gamma^m + \deg_{1,\varpi} \mathcal{Q}^*$, i.e., $\deg_{1,\varpi} \mathcal{Q}^{*\prime} < \deg_{1,\varpi} \mathcal{Q}^*$, which contradicts $\mathcal{Q}^*$ being the minimum polynomial in $\mathcal{I}_{\mathbf{P}^*}$. ∎

Therefore, after constructing $\mathcal{M}_{\mathbf{P}^*}$, we can first obtain $\mathcal{Q}^*$ by Lee-O'Sullivan's algorithm that was introduced in Section III.C. Based on (50), the interpolation polynomial $\widetilde{\mathcal{Q}}$ can be further obtained by

$$\widetilde{\mathcal{Q}} = \mathcal{G}_\Gamma^m \mathcal{Q}^*\left(x, y, \frac{z}{\mathcal{G}_\Gamma}\right), \tag{57}$$

which interpolates the interpolation points of (40) with a multiplicity of m. The z-roots f' of $\widetilde{\mathcal{Q}}$ can be further determined. Estimation of the intended message polynomial f can be further obtained by

$$\hat{f} = f' + \mathcal{K}_\Gamma. \tag{58}$$

Summarizing the above description, the above re-encoding transformed BR interpolation algorithm is stated as follows.

Algorithm 2 The ReT BR Interpolation

Input: $\underline{r}$ and m;
Output: $\widetilde{\mathcal{Q}}$;

1: Transform the set of interpolation points of $\mathbf{P}$ into $\mathbf{P}'$ as in (37)-(40);
2: Obtain subset $\mathbf{P}^*$ of (48);
3: Initialize $\mathcal{M}_{\mathbf{P}^*}$ as in (51)-(53);
4: Represent $\mathcal{M}_{\mathbf{P}^*}$ as matrix $\mathcal{V}^*$ over $\mathbb{F}_q[x]$ as in (55);
5: Reduce $\mathcal{V}^*$ into a weak Popov form matrix $\mathcal{V}^{*\prime}$ using Lee-O'Sullivan's algorithm;
6: Demap the matrix $\mathcal{V}^{*\prime}$ as $\mathcal{M}'_{\mathbf{P}^*}$ as in (34);
7: Pick up the minimum candidate from $\mathcal{M}'_{\mathbf{P}^*}$ as $\mathcal{Q}^*$;
8: Construct $\widetilde{\mathcal{Q}}$ as in (57).

Based on Table I, the following Theorem shows the correctness and characterizes the complexity of Algorithm 2.

Theorem 17: Algorithm 2 is correct. Given basis $\mathcal{M}_{\mathbf{P}^*}$, it can be presented as a matrix $\mathcal{V}^* \in \mathbb{F}_q[x]^{2(l+1)\times 2(l+1)}$. The complexity of Algorithm 2 is $O(l^3\ m^2(n-k)^2)$.

Proof: Based on eqs. (50) and (51), $\mathcal{M}_{\mathbf{P}^*}$ is a basis of $\mathcal{I}_{\mathbf{P}^*}$. By using Lee-O'Sullivan's algorithm, $\mathcal{M}_{\mathbf{P}^*}$ can be reduced into a Gröbner basis of $\mathcal{I}_{\mathbf{P}^*}$ w.r.t. the $(1, \varpi)$-revlex order. Based on Theorem 16 and (57), $\widetilde{\mathcal{Q}}$ is a minimum polynomial of module $\mathcal{I}_{\mathbf{P}'}$. Therefore, by determining the z-roots f' of $\widetilde{\mathcal{Q}}$, the intended message polynomial f can be further obtained. Therefore, Algorithm 2 is correct.

TABLE II
COMPLEXITY FOR REDUCING $\mathbb{V}$ INTO THE WEAK POPOV FORM

Algorithm	Complexity in O	Complexity in $O^{\sim}$
Lee-O'Sullivan, MS	(60)	$l^3m^2n^2$
Alekhnovich	(61)	$l^{\pi+1}mn$
GJV	(62)	$l^{\pi}mn$

For an (n,k) elliptic code, based on Theorem 4, the re-encoding polynomial $\mathcal{K}_\Gamma$ of (35) can be computed with a complexity of $O^{\sim}(k)$. Obtaining the interpolation points of (48) requires $O^{\sim}(n)$ field operations. Since $\deg_{1,k}\mathcal{K}_{\Gamma^c} \leq n-\varepsilon+1$, computing $\mathcal{K}_{\Gamma^c}$ requires $O^{\sim}(n-\varepsilon)$ field operations. Therefore, computing $\mathcal{H}_\Gamma^{(t)}$ of (52) and (53) requires $O^{\sim}(m^3(n-\varepsilon))$ and $O^{\sim}(m^2(l-m)n)$ field operations, respectively. Hence, complexity of the basis construction is $O^{\sim}(m^3n)$.

Based on (51) and (55), if $0 \leq t \leq 2m+1$, $\deg \mathcal{V}_t^* = m(n-\varepsilon) + \lfloor \frac{t}{2} \rfloor (1-\varpi) + w_t$ and $\deg \mathcal{V}_{t,t}^* = (n-\varepsilon)(m - \lfloor \frac{t}{2} \rfloor) + w_t$. If $2m+1 < t < 2l+2$, $\deg \mathcal{V}_t^* = m(n - 2\varepsilon - \varpi + 1) + \varepsilon \lfloor \frac{t}{2} \rfloor + w_t$ and $\deg \mathcal{V}_{t,t}^* = \varepsilon(\lfloor \frac{t}{2} \rfloor - m) + w_t$. Therefore, $\deg \mathcal{V}^* \leq m(n-\varepsilon+1) + (l-m)(\varepsilon+\varpi) + 3$ and $\Delta(\mathcal{V}^*) < 2ml(n-k)$. Reducing $\mathcal{V}^*$ into a Gröbner basis requires at most $(2l+2)^2 \deg_{\underline{w}} \mathcal{V}^* \Delta(\mathcal{V}^*) \approx 8l^3m(n-k)\left(m(n-k)+k(l-m)\right)$ field operations. Moreover, restoring the interpolation polynomial as in (57) requires $O^{\sim}(lmn)$ field operations. Therefore, the complexity of Algorithm 2 can be characterized as $O(l^3\ m^2(n-k)^2)$. ∎

In comparison with Theorems 8 and 9, it can be seen that the ReT helps reduce both of the basis construction and reduction complexity by a factor of $\frac{k}{n}$. Therefore, this reduction will be more effective for high rate codes.

V. COMPLEXITY COMPARISON

We will further investigate the complexity of the basis reduction process using different approaches including the Lee-O'Sullivan algorithm [26], the MS algorithm [28], the Alekhnovich algorithm [29] and the GJV algorithm [30]. This analysis will reveal that the Lee-O'Sullivan algorithm remains the most efficient for decoding codes of practical length.

Given a matrix $\mathcal{V}$ as in (30), different from Lee-O'Sullivan's algorithm, the following mapping $\Phi_{\underline{w}}$ is needed before the other three algorithms are used to reduce $\mathcal{V}$ into the weak Popov form. Define the mapping $\Phi_{\underline{w}}$ [12]: $\mathbb{F}_q[x]^{2(l+1)} \rightarrow \mathbb{F}_q[x]^{2(l+1)}$

$$\begin{aligned}\mathcal{V}_t \mapsto \mathbb{V}_t =& (x^{w_0}M_t^{(0)}(x), x^{w_1}M_t^{(1)}(x), \ldots, \\ & x^{w_{2l+1}}M_t^{(2l+1)}(x)),\end{aligned} \tag{59}$$

where $\underline{w} = (w_0, w_1, \ldots, w_{2l+1})$ and $w_s = \lfloor \frac{k\lfloor \frac{s}{2} \rfloor + 3(s \bmod 2)}{2} \rfloor$. Therefore, matrix $\mathbb{V}$ will be reduced. We now look into the complexity of the other three basis reduction algorithms under the decoding paradigm without the ReT.

Theorem 18: Given a matrix $\mathbb{V} \in \mathbb{F}_q[x]^{2(l+1)\times 2(l+1)}$, the complexity of reducing $\mathbb{V}$ into the weak Popov form is summarized in Table II.

Proof: Based on (59), we know $\deg \mathbb{V} \leq \deg_x \mathbb{V}_{2l+1,2l-2m} < \frac{mn}{2} + \frac{k(l-m)}{2}$. If $0 \leq t \leq 2m+1$, $\deg \mathbb{V}_t \approx \frac{mn}{2}$ and $\deg \mathbb{V}_{t,t} \approx \frac{n}{2}(m - \frac{t}{2}) + w_t$. If $2m+2 \leq t \leq 2l+1$, $\deg \mathbb{V}_t \approx \frac{mn}{2} + k\frac{t-2m}{4}$ and $\deg \mathbb{V}_{t,t} = w_t$. Therefore, $\mathrm{rowdeg}\mathbb{V} = \sum_{t=0}^{2l+1} \deg \mathbb{V}_t \approx lmn + \frac{k}{2}(l-m)^2$ and $\deg \det \mathbb{V} = \sum_{t=0}^{2l+1} \deg \mathbb{V}_{t,t} \approx \frac{m^2n}{2} + \frac{l^2k}{2}$. Hence, $\Delta(\mathbb{V}) \approx ml(n-k)$. Based on the characterizations of Table I, the complexity characterizations in l, m and n for the Lee-O'Sullivan (and MS) algorithm, the Alekhnovich algorithm and the GJV algorithm are given in (60), (V), and (V), respectively. Omitting the $\log$ factors, their complexity characterizations in $O^{\sim}$ are further summarized in Table II.

$$2l^2mn(lm(n-k)+2l), \tag{60}$$

$$\begin{aligned}&(18+72\log_3 2)(2l)^{\pi}(ml(n-k)\log^2(ml(n-k)) \\ &\cdot \log\log(ml(n-k)) + mn\log(mn)\log\log(mn)),\end{aligned} \tag{61}$$

$$\begin{aligned}&(18+72\log_3 2)(2l)^{\pi}mn\log(mn)\log(lmn) \\ &\cdot \log(m^2n)\log\log(mn).\end{aligned} \tag{62}$$

∎

Since $\pi \leq 3$, Theorem 18 shows that the Alekhnovich and the GJV algorithms have lower asymptotically complexity. However, by carefully investigating the expressions of (60)-(V), we know that for practical l, m and n, the Lee-O'Sullivan and the MS algorithms would yield a lower complexity. Note that the running times of the MS and the Alekhnovich algorithms in Magma have been examined in [31], which also showed the former has a smaller running time when n is less than 3000.

VI. SIMULATION RESULTS

GS decoding of elliptic codes using the proposed BR interpolation have been simulated using C programming language. It is measured over the additive white Gaussian noise (AWGN) channel using BPSK. The decoding performances are presented as a function of the signal-to-noise ratio (SNR) which 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. For the BPSK modulated transmission, codeword symbols are firstly converted into binary representations, and 0 and 1 are mapped to the BPSK symbols of $(1,0)$ and $(-1,0)$, respectively. The decoding complexity is measured as the number of finite field arithmetic operations needed in decoding a codeword. The elliptic codes are compared with similar rate RS codes that are defined over the same finite field.

Figs. 1-3 show the frame error rate (FER) performance of the $(80,27)$, the $(80,39)$ and the $(288,163)$ elliptic codes, respectively. Our results show that with the same interpolation multiplicity m, elliptic codes can outperform their RS counterparts. Note that for all the codes, we choose the minimum m that yields the corresponding error-correction capability. The decoding parameters are defined in Theorem 2. Over the same finite field, elliptic codes are longer. They inherit a greater error-correction capability, yielding a better decoding performance. It is interesting to point out that this also enables elliptic codes achieve a similar performance as RS codes with a *smaller* decoding complexity. E.g., Fig.1 shows that decoding the $(80,27)$ elliptic code with $m=4$ performs similarly as decoding the $(63,21)$ RS code with $m=5$.

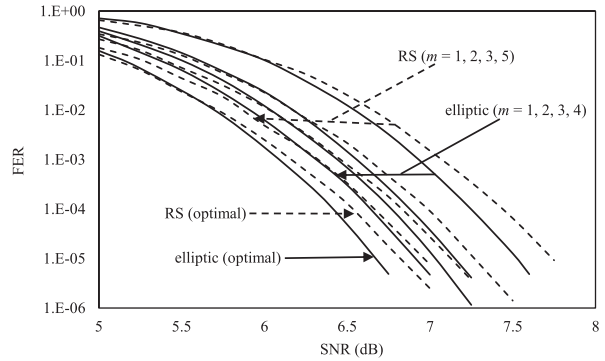

Fig. 1. Performance of the (80, 27) elliptic code and the (63, 21) RS code.

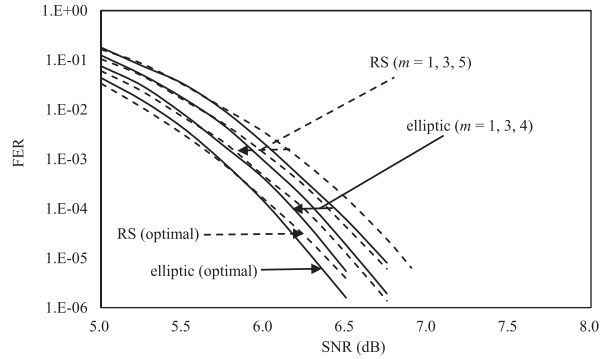

Fig. 2. Performance of the (80, 39) elliptic code and the (63, 31) RS code.

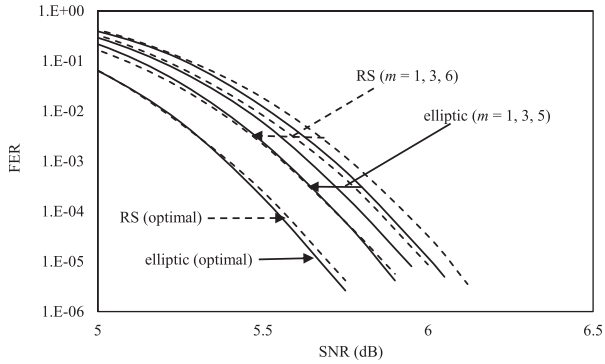

Fig. 3. Performance of the (288, 163) elliptic code and the (255, 144) RS code.

The BR interpolation complexity is 1.16×10^7 for the elliptic code, and 1.20×10^7 for the RS code. If using Kötter's interpolation, the complexity will be 1.65×10^7 for the elliptic code, and 2.72×10^7 for the RS code. A similar phenomenon can also be observed in Figs. 2 and 3. They demonstrate that elliptic codes can be considered to replace RS codes for better performance. Note that the optimal GS decoding performances are obtained by assuming the algorithm can decode at most τ_{GS} errors, where τ_{GS} was defined by eq. (15) for elliptic codes. For RS codes, $\tau_{\mathrm{GS}} = n - \left\lfloor \sqrt{n(k-1)} \right\rfloor - 1$.

The proof of Theorem 9 shows that the basis reduction dominates the interpolation complexity, and its complexity reduces as the code rate increases. Tables III-IV show the

TABLE III
INTERPOLATION COMPLEXITY IN DECODING THE $(80, 27)$ ELLIPTIC CODE

(m, l, τ)		$(2, 3, 29)$	$(4, 7, 31)$	$(7, 12, 32)$
Kötter		7.93×10^5	1.65×10^7	2.14×10^8
BR	Construction	1.46×10^4	4.85×10^4	1.78×10^5
	Reduction	4.48×10^5	1.16×10^7	1.91×10^8

TABLE IV
INTERPOLATION COMPLEXITY IN DECODING THE $(80, 39)$ ELLIPTIC CODE

(m, l, τ)		$(2, 3, 20)$	$(4, 5, 22)$	$(8, 11, 23)$
Kötter		6.78×10^5	8.05×10^6	2.15×10^8
BR	Construction	1.46×10^4	4.85×10^4	2.50×10^5
	Reduction	2.80×10^5	4.06×10^6	1.36×10^8

TABLE V
RET INTERPOLATION COMPLEXITY IN DECODING THE $(80, 27)$ ELLIPTIC CODE

(m, l, τ)		$(2, 3, 29)$	$(4, 7, 31)$	$(7, 12, 32)$
Kötter		1.05×10^6	1.17×10^7	1.44×10^8
BR	Construction	3.08×10^4	8.95×10^4	8.78×10^5
	Reduction	3.21×10^5	9.95×10^6	1.33×10^8

TABLE VI
RET INTERPOLATION COMPLEXITY IN DECODING THE $(80, 39)$ ELLIPTIC CODE

(m, l, τ)		$(2, 3, 20)$	$(4, 5, 22)$	$(8, 11, 23)$
Kötter		8.19×10^5	3.87×10^6	9.32×10^7
BR	Construction	4.44×10^4	6.43×10^4	2.46×10^5
	Reduction	1.50×10^5	2.11×10^6	7.09×10^7

numerical results of the interpolation complexity in decoding the $(80, 27)$ and the $(80, 39)$ elliptic codes, respectively. Our numerical results validate the characterization of Theorem 9. With the same decoding parameters, e.g., $m = 2$ and $l = 3$, the BR interpolation exhibits a smaller complexity for the $(80, 39)$ elliptic code. Note that the complexity of Kötter's interpolation is $O(lm^4\ n^2)$, which exhibits the same asymptotic behavior as the BR interpolation. The ratio of the complexity of the two interpolation approaches is $\frac{l^3 m^2 n(n-k)}{lm^4\ n^2} = (\frac{l}{m})^2(1 - \frac{k}{n})$. Since $m \leq l$, their comparison depends on the interplay between $\frac{l}{m}$ and $\frac{k}{n}$. Our numerical results show in practice, the BR interpolation has a smaller complexity than Kötter's interpolation, with more significant reduction realized for the higher rate codes.

Tables V-VI show the numerical results of the BR interpolation assisted by the ReT. Pairing Tables III and V, IV and VI, the complexity reduction factor of $\frac{k}{n}$ can be observed. Note that with the ReT, Kötter's interpolation exhibits a complexity of $O(lm^4\ n(n-k))$. Therefore, complexity of the BR interpolation and Kötter's interpolation still exhibit a ratio of $\left(\frac{l}{m}\right)^2 (1 - \frac{k}{n})$. Our numerical results also show that for the

elliptic codes, the BR interpolation has a complexity advantage over Kötter's interpolation by at most an order of magnitude.

VII. Conclusion

This paper has proposed the GS algorithm for decoding elliptic codes using the module basis reduction interpolation. The Lagrange interpolation function over the elliptic function fields has been proposed for constructing the module. A basis of the module containing polynomials that satisfy all the interpolation and degree constrains has also been defined. This basis can be further reduced by the Lee-O'Sullivan algorithm, resulting in the desired Gröbner basis that contains the interpolation polynomial $\mathcal{Q}$. The re-encoding transform has been further introduced to reduce the degree of module basis entries, yielding a basis isomorphism. Its reduction exhibits a smaller complexity. This research has further shown the BR interpolation yields a complexity of $O(l^3\ m^2 n(n-k))$. Assisted by the re-encoding transform, this complexity can be reduced to $O(l^3\ m^2(n-k)^2)$. They both show the interpolation technique will have a smaller complexity for high rate codes. Furthermore, we have analysed the complexity of using the Alekhnovich and the GJV algorithms for the basis reduction process. They show a complexity of $O^{\sim}(l^{\pi+1}mn)$ and $O^{\sim}(l^{\pi}mn)$, respectively. We have also demonstrated that for codes of practical length, the Lee-O'Sullivan and the MS algorithms will be more efficient. Finally, the BR interpolation's complexity advantage over Kötter's interpolation has also been demonstrated. Our simulation results have also demonstrated that elliptic codes can outperform the similar rate RS codes defined over the same finite field.

References

[1] V. Goppa, "Codes associated with divisors," *Problemy Peredachi Informatsii*, vol. 13, no. 1, pp. 33–39, 1977.

[2] Y. Driencourt, "Some properties of elliptic codes over a field of characteristic 2," in *Proc. 3rd Int. Conf. (AAECC)*. Berlin, Germany: Springer, 1985, pp. 185–193.

[3] W. W. Peterson, "Encoding and error-correction procedures for Bose-Chaudhuri codes," *IRE Trans. Inf. Theory*, vol. 6, pp. 459–470, Mar. 1960.

[4] J. Justesen, K. J. Larsen, H. E. Jensen, A. Havemose, and T. Hoholdt, "Construction and decoding of a class of algebraic geometry codes," *IEEE Trans. Inf. Theory*, vol. 35, no. 4, pp. 811–821, Jul. 1989.

[5] A. Skorobogatov and S. Vladut, "On the decoding of algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 36, no. 5, pp. 1051–1060, Sep. 1990.

[6] G. Feng and T. Rao, "Decoding algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inf. Theory*, vol. 39, no. 1, pp. 37–45, Jan. 1993.

[7] I. M. Duursma, "Majority coset decoding," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 1067–1070, May 1993.

[8] S. Sakata, "Extension of the Berlekamp-Massey algorithm to N dimensions," *Inf. Comput.*, vol. 84, no. 2, pp. 207–239, Feb. 1990.

[9] 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.

[10] M. Johnston and R. Carrasco, "Construction and performance of algebraic–geometric codes over AWGN and fading channels," *IEE Proc.-Commun.*, vol. 152, no. 5, pp. 713–722, Oct. 2005.

[11] G. Schmidt, V. Sidorenko, and M. Bossert, "Decoding Reed–Solomon codes beyond half the minimum distance using shift-register synthesis," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2006, pp. 459–463.

[12] 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.

[13] M. Sudan, "Decoding of Reed–Solomon codes beyond the error-correction bound," *J. Complex.*, vol. 13, no. 1, pp. 180–193, Mar. 1997.

[14] M. A. Shokrollahi and H. Wasserman, "List decoding of algebraic-geometric codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 432–437, Mar. 1999.

[15] 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.

[16] R. Kötter, "On algebraic decoding of algebraic-geometric and cyclic codes," Ph.D. dissertation, Dept. Elect. Eng., Linköping Univ., Linköping, Sweden, 1996.

[17] 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.

[18] R. Koetter and A. Vardy, "A complexity reducing transformation in algebraic list decoding of Reed–Solomon codes," in *Proc. IEEE Inf. Theory Workshop*, Paris, France, Mar./Apr. 2003, pp. 10–13.

[19] T. Høholdt and R. Nielsen, "Decoding Hermitian codes with Sudan's algorithm," in *Proc. AAECC* (Lecture Notes in Computer Science), vol. 1719. Berlin, Germany: Springer-Verlag, 1999, pp. 260–269.

[20] 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.

[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] H. O'Keeffe and P. Fitzpatrick, "Gröbner basis solutions of constrained interpolation problems," *Linear Algebra Appl.*, vols. 351–352, pp. 533–551, Aug. 2002.

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

[24] 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.

[25] J. S. R. Nielsen, "List decoding of algebraic codes," Ph.D. dissertation, Dept. Appl. Math. Comput. Sci., Tech. Univ. Denmark, Lyngby, Denmark, 2013.

[26] 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.

[27] 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.

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

[29] 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.

[30] 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.

[31] 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.

[32] L. Washington, *Elliptic Curves: Number Theory and Cryptography*. Boca Raton, FL, USA: CRC Press, 2008.

[33] C. Munuera, "On MDS elliptic codes," *Discrete Math.*, vol. 117, nos. 1–3, pp. 279–286, 1993.

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

[35] 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.

[36] 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.*, Glasgow, U.K., Jun. 2007, pp. 851–856.

[37] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[38] R. M. Roth and G. Ruckenstein, "Efficient decoding of Reed–Solomon codes beyond half the minimum distance," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 246–257, Jan. 2000.

[39] P. Beelen and T. Høholdt, "The decoding of algebraic geometry codes," *Adv. Algebraic Geometry Codes*, vol. 5, pp. 49–98, Mar. 2008.

Yunqi Wan (Member, IEEE) 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. He is currently pursuing the Ph.D. degree in electronics and information technology with Sun Yat-sen University, Guangzhou, China. 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 to 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 became an Associate Professor and a Professor of Sun Yat-sen University, respectively. 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 Conference Committee and External Nomination Committee, a Committee Member of the CIE Information Theory Society, and the Chair of the IEEE Information Theory Society Guangzhou Chapter. He has been involved in organizing several international conferences, including the 2018 IEEE Information Theory Workshop (ITW) at Guangzhou, for which he was the General Co-Chair. He is an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. He likes reading and photography.

Fangguo Zhang was born in 1972. He received the Ph.D. degree from the School of Communication Engineering, Xidian University, in 2001. He is currently a Professor with the School of Computer Science and Engineering, Sun Yat-sen University, China. He is the Co-Director of 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.