New Locally Correctable Codes Based on Projective Reed–Muller Codes

Sian-Jheng Lin, *Member, IEEE*, Yunghsiang S. Han, *Fellow, IEEE* and Nenghai Yu

***Abstract*—Locally decodable codes (LDCs) and locally correctable codes (LCCs) have several important applications, such as private information retrieval, secure multiparty computation, and circuit lower bounds. Three major parameters are considered in LCCs: query complexity, message length, and codeword length. The most familiar LCCs in the regime of low query complexity are the generalized Reed–Muller (GRM) codes. However, it has not previously been determined whether there exist codes that have shorter codeword lengths than GRM codes with the same query complexity and message length. In this paper, we show that projective Reed–Muller (PRM) codes are such LCCs for some parameters. The GRM code is specified by the alphabet size q, the number of variables m, and the degree d, where $d \leq q-2$. When $d = q-2$ and $q-1$ is a power of a prime, we prove that there exists a PRM code with shorter codeword length than a GRM code with the same query complexity and message length. We also present for these PRM codes a perfectly smooth local decoder to recover a symbol in a codeword by accessing no more than q symbols at the coordinates of the codeword.**

I. Introduction

Locally decodable codes (LDCs) [1] are a class of error-correcting codes that allow each message symbol in the codeword to be corrected probabilistically. LDCs access a low number of symbols in the codeword via a randomized algorithm. If local decoding is available for both the message symbols and the parity symbols, the code is called a locally correctable code (LCC) [2]. LDCs and LCCs have several important applications, such as private information retrieval, secure multiparty computation, and circuit lower bounds.

To evaluate LDCs or LCCs, three metrics are considered: the query complexity γ, the message length k, and the codeword length n. The query complexity indicates the number of codeword symbols that need to be accessed to recover a faulty symbol in the codeword. The message length indicates the number of message symbols to be encoded. The codeword length is the number of symbols in the codeword. When the query complexity and the message length are specified, one important research problem is how to construct a code with the shortest codeword length.

To date, a number of LDCs have been proposed for different ratios between the query complexity and the message length. A typical family of LCCs is represented by the generalized Reed–Muller (GRM) codes [3] discovered in the 1960s. GRM codes are obtained by generalizing binary Reed–Muller codes [4], [5] to larger finite fields. GRM codes were the first LCCs/LDCs to be constructed and all later codes of LCCs/LDCs can be seen as generalizations of these. When the coding rate, which is defined as k/n, is greater than $\frac{1}{2}$, GRM codes lose their local decoding ability. In this regime, the LCCs, termed as multiplicity codes [6] and the codes from lifting [7], are available. Furthermore, when the query complexity is very low, matching vector (MV) codes [8], [9] form the known shortest codes in LDCs; however, they are not LCCs. Until now, GRM codes are still the shortest codes among LCCs in the regime of low query complexity. In [2, Sec. 8.3], Yekhanin raised an open question regarding whether there exist codes that are shorter than GRM codes. In this work, we answer a relaxed version of this question by showing that projective Reed–Muller (PRM) codes are LCCs and they are shorter than GRM codes for some parameters.

Lachaud [10] introduced PRM codes by extending Reed-Muller codes to projective spaces. Their dimensions and minimum distances were determined by Sørensen [11]. Since then, the properties of these codes have been intensively studied [12], [13], [14], [15], [16]. However, the local correctability of PRM codes has never been investigated. Apart from the decoding approaches that have been previously proposed [17], [18], we provide a decoder that presents local error-correcting capability of the PRM codes for low query complexities. This regime was previously occupied by GRM codes, and hence the present results are compared with results obtained with these codes. When we align the query complexity and the message length of PRM codes and GRM codes, the proposed PRM codes have better performance on codeword lengths and field sizes.

We can summarize the main contributions of this work as follows.

1) We show that the PRM codes form a class of LCCs.
2) Some proposed PRM codes have shorter code lengths than GRM codes when the query complexity and message length of the codes are set to be the same.
3) A perfectly smooth local decoder for a PRM code is proposed. The decoder can recover a symbol in a codeword by accessing no more than q symbols of the codeword, where q is the size of the finite field.

The rest of this paper is organized as follows. Section II introduces the definitions of LCCs and a number of traditional error-correcting codes. Section III presents the proposed local

This work was supported in part by the Hundred Talents Program of Chinese Academy of Sciences, National Natural Science Foundation of China (Grant No. 61671007), and Start Fund of Dongguan University of Technology (No. KCYXM2017025). Sian-Jheng Lin and Nenghai Yu are with the School of Information Science and Technology, University of Science and Technology of China (USTC), the CAS Key Laboratory of Electro-magnetic Space Information, China. (e-mail: sjlin@ustc.edu.cn, ynh@ustc.edu.cn); Yunghsiang S. Han is with the School of Electrical Engineering & Intelligentization, Dongguan University of Technology, Dongguan, China (e-mail: yunghsiangh@gmail.com)

decoder for PRM codes. Section IV analyzes the local correctabilities of PRM codes and makes comparisons with other codes. Section V concludes the paper.

II. Preliminaries

A. Definitions and Notation

Let $\mathbb{F}_q$ denote a finite field with q elements, where q is a prime power, and let $\mathbb{F}_q^* = \mathbb{F}_q \setminus \{0\}$. For ease of notation, $\mathbf{X}$ can represent $\mathbf{X} = (X_1, \dots X_m)$ or $\mathbf{X} = (X_0, \dots X_m)$ and $[m] = \{1, 2, \dots, m\}$. An m-dimensional affine space over $\mathbb{F}_q$ is defined as

$$\mathbb{A}^m(\mathbb{F}_q) := \{(a_1, \dots, a_m) | a_j \in \mathbb{F}_q, j \in [m]\}.$$

Further, an m-dimensional projective space is defined as

$$\mathbb{P}^m(\mathbb{F}_q) := (\mathbb{A}^{m+1}(\mathbb{F}_q) \setminus \{\mathbf{0}\}) / \sim,$$

where $\mathbf{0}$ is the origin on $\mathbb{A}^m(\mathbb{F}_q)$ and $\sim$ is the equivalence relation defined as follows: given $(a_0, a_1, \dots, a_m)$ and $(b_0, b_1, \dots, b_m)$, if there exists $\lambda \in \mathbb{F}_q^*$ such that $(a_0, a_1, \dots, a_m) = (\lambda b_0, \lambda b_1, \dots, \lambda b_m)$, then this can be written as

$$(a_0, a_1, \dots, a_m) \sim (b_0, b_1, \dots, b_m).$$

For simplicity, $\mathbb{A}^m$ and $\mathbb{P}^m$ are used to denote $\mathbb{A}^m(\mathbb{F}_q)$ and $\mathbb{P}^m(\mathbb{F}_q)$, respectively.

Let

$$\mathcal{H}_d^m = \mathbb{F}_q[X_1, \dots, X_m]_d \cup \{0\},$$

where $\mathbb{F}_q[X_1, \dots, X_m]_d$ is a polynomial ring consisting of the homogeneous polynomials of degree d. For any $F(\mathbf{X}) \in \mathcal{H}_d^m$, it is known that

$$F(\lambda \mathbf{X}) = \lambda^d F(\mathbf{X}) \qquad \forall \lambda \in \mathbb{F}_q^*. \tag{1}$$

Let $F(P)$ be the evaluation of F in some representative $P = [p_1 : p_2 : \cdots : p_m] \in \mathbb{P}^m$. Equation (1) shows that $F(P)$ depends on the choice of the representative of P. To avoid confusion, it is necessary to specify the representative of the elements in $\mathbb{P}^m$. For $P \in \mathbb{P}^m$, we define

$$\mathbf{D}(P) = p_i,$$

where i is the smallest integer such that $p_i \neq 0$. Then the representative of P is defined by

$$\mathbf{N}(P) := (0, \dots, 0, 1, p'_{i+1}, \dots, p'_m),$$

where each $p'_j = p_j / \mathbf{D}(P)$, for $j \geq i+1$. In addition, let

$$\mathbf{N}(\mathbb{P}^m) := \{\mathbf{N}(P) | P \in \mathbb{P}^m\}.$$

For $\mathbf{x}, \mathbf{y} \in \mathbb{F}_q^m$, $\Delta(\mathbf{x}, \mathbf{y})$ denotes the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$, i.e., the number of positions where $\mathbf{x}$ and $\mathbf{y}$ have distinct elements. For $\mathbf{x} \in \mathbb{F}_q^m$, $\mathbf{x}[i]$ denotes the ith symbol of $\mathbf{x}$, and $\mathbf{x}|_S$ denotes $\mathbf{x}$ restricted to symbols indexed by $S \subset [m]$.

B. Locally correctable codes

A class of codes of message length k and codeword length n is called $(\gamma, \delta, \epsilon)$-locally correctable if for each received codeword $\mathbf{y}$ with up to δn errors, each symbol $\mathbf{y}[i]$, $i \in [n]$, can be recovered with probability $1 - \epsilon$ by accessing at most γ symbols chosen by a randomized algorithm. The following is a formal definition of locally correctable codes.

Definition 1. *(Locally correctable code (LCC)) A code $\mathcal{C} \subset \mathbb{F}_q^n$ is $(\gamma, \delta, \epsilon)$-locally correctable if there exists a randomized algorithm (local decoder) $\mathcal{A}$ such that for each pair $(\mathbf{c} \in \mathcal{C}, \mathbf{y} \in \mathbb{F}_q^n)$ and $\Delta(\mathbf{c}, \mathbf{y}) \leq \delta n$,*

$$\Pr[\mathcal{A}(\mathbf{y}, i) = \mathbf{c}[i]] \geq 1 - \epsilon$$

holds for each $i \in [n]$. Furthermore, $\mathcal{A}$ accesses at most γ symbols of $\mathbf{y}$.

In Definition 1, if the local decoding property is available for $i \in [k]$, such codes are called locally decodable codes (LDCs). Clearly, LCCs are LDCs. Since this paper only considers LCCs, details of LDCs are omitted.

A local decoder $\mathcal{A}$ consists of two parts: the randomized query algorithm Q and the deterministic reconstruction algorithm R. A local decoder with query complexity γ is called perfectly smooth if the following requirements are satisfied [19], [20]. First, the deterministic reconstruction algorithm can recover any codeword symbol by accessing at most other γ symbols within the codeword. Second, the randomized query algorithm meets the requirement that, for each symbol, all other symbols have an equal chance of being selected in the set of queries (e.g. the second condition in the following definition). The following is a formal definition.

Definition 2. *(Perfectly smooth decoder) For a $(\gamma, \delta, \epsilon)$-locally correctable code $\mathcal{C} \subset \mathbb{F}_q^n$ with a local decoder, $\mathcal{A}$ consists of a randomized query algorithm Q and a deterministic reconstruction algorithm $R : \mathbb{F}_q^\gamma \times [n] \to \mathbb{F}_q$. For each $c \in \mathcal{C}$ and a point $i \in [n]$, Q reads i and generates a set of queries $Q(i)$ with $|Q(i)| \leq \gamma$. Next, R reads $c|_{Q(i)}$ and i to recover $c[i]$. The local decoder is perfectly smooth if the following conditions hold:*

1) *For each $c \in \mathcal{C}$ and $i \in [n]$,*

$$\Pr[R(c|_{Q(i)}, i) = c[i]] \geq 1 - \epsilon.$$

2) *For each $i \in [n]$, each query in $Q(i)$ is uniformly distributed over $[n]$. That is, for the j-th query $Q(i)[j]$ and $j \in \gamma$,*

$$Pr[Q(i)[j] = k] = 1/(n-1), \qquad \forall k \in [n] \setminus \{i\}.$$

C. Error-correcting codes

A number of error-correcting codes are introduced in this subsection.

1) Reed–Solomon (RS) codes: $(n, d+1)$ RS codes [21] over $\mathbb{F}_q$ treat the message as a single-variate polynomial of degree less than or equal to d, and the codeword is generated by evaluating this polynomial at $n = q$ fixed points. In addition, an extended Reed–Solomon (ERS) code (also called a doubly extended Reed–Solomon code) is constructed by appending an extra symbol to the codeword of a $(q, d+1)$ RS code, where the extra symbol is the coefficient of the polynomial at degree d. The formal definition is as follows.

Definition 3. *The Reed–Solomon code over $\mathbb{F}_q$ of order d and length $n = q$ is defined by*

$$\mathbf{RS}_q(d) = \{(F(\lambda))_{\lambda\in\mathbb{F}_q} | F(X) \in \mathbb{F}_q[X], \deg F \leq d\}.$$

The extended Reed–Solomon code is defined by

$$\begin{aligned}\mathbf{ERS}_q(d) =&\{(F(\lambda_0), \dots, F(\lambda_{q-1}), F(\lambda_\infty))| \\ &F(X) \in \mathbb{F}_q[X], \deg F \leq d\},\end{aligned}$$

where $F(\lambda_\infty)$ denotes the coefficient of X^d.

RS codes are maximum distance separable (MDS) codes, which possess the optimal trade-off between the minimum Hamming distance and the size of redundancy of the code. $(n, d+1)$ RS codes are able to correct up to E errors and S erasures, as long as $2E+S \leq n-d-1$. A number of typical decoders, such as the Berlekamp–Welch algorithm [22], the Berlekamp-Massey algorithm [23], and algorithms based on the fast Fourier transform (FFT) [24], [25], can be used to decode RS codes. In addition, efficient decoders for ERS codes have been presented in [26], [27].

2) Generalized Reed–Muller (GRM) codes: GRM codes [3] are a family of linear error-correcting codes obtained by constructing Reed–Muller codes [5] over an arbitrary finite field. For a GRM code $\mathbf{GRM}_d(m, q)$, the message is determined by an m-variate polynomial of degree at most d over $\mathbb{F}_q$, and the codeword is defined by evaluations of the polynomial at points from $\mathbb{A}^m$. The formal definition is as follows.

Definition 4. *The generalized Reed–Muller code over $\mathbb{F}_q$ of order d and length $n = q^m$ is defined by*

$$\begin{aligned}\mathbf{GRM}_q(d, m) =&\{(F(A))_{A\in\mathbb{A}^m}| \\ &F(\mathbf{X}) \in \mathbb{F}_q[X_1, \dots, X_m], \deg F \leq d\},\end{aligned}$$

where $F(\mathbf{X})$ is an m-variate polynomial of degree at most d over $\mathbb{F}_q$.

The code dimensions and minimum distances of GRMs were determined in [28, p. 72]. In particular, $\mathbf{GRM}_q(d, 1)$ are RS codes, and $\mathbf{GRM}_2(1, m)$ are punctured Hadamard codes. Note that any vector A in $\mathbb{A}^m$ represents a corresponding coordinate in a codeword, where the symbol at this coordinate is $F(A)$.

When $d \leq q-2$, GRM codes form a typical family of locally correctable codes of query complexity $d+1$, message length $k = \binom{m+d}{d}$, and code length $n = q^m$. To recover a symbol at coordinate $\mathbf{w} \in \mathbb{A}^m$, the local decoder randomly picks a coordinate $\mathbf{v} \in \mathbb{A}^m \setminus \mathbf{w}$ and randomly selects $d+1$ points falling on

$$L = \{\mathbf{w} + \lambda\mathbf{v} | \lambda \in \mathbb{F}_q^*\}. \tag{2}$$

The local decoder queries those symbols evaluated by the points in L. From (2), the set of symbols evaluated at the points in L is given by

$$\{F(\ell)|\ell \in L\} = \{F(\mathbf{w}+\lambda\mathbf{v})|\lambda \in \mathbb{F}_q^*\} = \{F_{\mathbf{w},\mathbf{v}}(\lambda)|\lambda \in \mathbb{F}_q^*\},$$

where $F_{\mathbf{w},\mathbf{v}}(X) := F(\mathbf{w}+X\mathbf{v})$ is a single-variate polynomial, and $\deg(F_{\mathbf{w},\mathbf{v}}(X)) \leq d$. By appending $F_{\mathbf{w},\mathbf{v}}(0)$ to the set, we obtain

$$\{F_{\mathbf{w},\mathbf{v}}(\lambda)|\lambda \in \mathbb{F}_q^*\} \cup \{F_{\mathbf{w},\mathbf{v}}(0)\} = \{F_{\mathbf{w},\mathbf{v}}(\lambda)|\lambda \in \mathbb{F}_q\}. \tag{3}$$

As $\deg(F_{\mathbf{w},\mathbf{v}}(\lambda)) \leq d$, from Definition 3, (3) forms a codeword of $\mathbf{RS}_q(d)$. If there is no error at the $d+1$ selected symbols, then, by applying the RS decoding algorithm, one can recover the single-variate polynomial $F_{\mathbf{w},\mathbf{v}}(X)$. Then $F_{\mathbf{w},\mathbf{v}}(0) = F(\mathbf{w}+0\cdot\mathbf{v}) = F(\mathbf{w})$ is the recovered symbol.

3) Projective Reed–Muller (PRM) codes: PRM codes [10] are a variant of GRM codes. For a PRM code, $\mathbf{PRM}_q(d, m)$, the message is determined by an $(m+1)$-variate homogeneous polynomial of degree d over $\mathbb{F}_q$, and the codeword is obtained by evaluating the polynomial in an $(m+1)$-dimensional projective space. The PRM codes are defined as follows.

Definition 5 ([11])**.** *The projective Reed–Muller code over $\mathbb{F}_q$ of order d and length $n = (q^{m+1}-1)/(q-1)$ is defined by*

$$\mathbf{PRM}_q(d, m) = \{(F(P))_{P\in\mathbf{N}(\mathbb{P}^m)} | F(\mathbf{X}) \in \mathcal{H}_d^{m+1}\}.$$

The code dimension and the minimum distance of PRM were determined in [11]. Notably, $\mathbf{PRM}_2(1, m)$ are reduced to Hadamard codes. In this case, the proposed local decoder given in the next section is the same as the local decoder for Hadamard codes. In particular, when $m = 1$, we have

$$\begin{aligned}\mathbf{PRM}_q(d, 1) =&\{(F(P))_{P\in\mathbf{N}(\mathbb{P})} | F(\mathbf{X}) \in \mathcal{H}_d^2\} \\ =&\{F(0, 1)\} \cup \{F(1, \lambda)\}_{\lambda\in\mathbb{F}_q},\end{aligned} \tag{4}$$

where $F(X_1, X_2) = \sum_{i=0}^{d} f_{d-i,i} X_1^{d-i} X_2^i$ is a 2-variate homogeneous polynomial of degree d. Let

$$F_1(X) = \sum_{i=0}^{d} f_{d-i,i} X^i.$$

In (4), it can be see that $F(0, 1) = f_{0,d}$ is the coefficient of $F_1(X)$ at degree d, and $F(1, \lambda) = F_1(\lambda)$, for $\lambda \in \mathbb{F}_q$. Thus, by Definition 3, $\mathbf{PRM}_q(d, 1)$ forms an ERS code.

III. Perfectly Smooth Decoder for PRM Codes

In this section, a $(d+1)$-query perfectly smooth decoder for $\mathbf{PRM}_q(d, m)$, $d \leq q-1$, is proposed. The approach is similar to the local decoder for GRM codes. Following Definition 2, the decoder $\mathcal{A}$ is denoted as a pair of algorithms (Q, R). Given a codeword $((F(P))_{P\in\mathbf{N}(\mathbb{P}^m)}) \in \mathbf{PRM}_q(d, m)$ and a point $\mathbf{w} \in \mathbb{P}^m$, the value $F(\mathbf{w})$ can be recovered via the following steps if the selected $d+1$ symbols involving in the decoding procedure have no error. First, the decoder randomly picks a coordinate $\mathbf{v} \in \mathbf{N}(\mathbb{P}^m) \setminus \mathbf{w}$. Then, we consider a line passing through $\mathbf{w}$ and $\mathbf{v}$:

$$L_{\mathbf{w}}(\mathbf{v}) := \{\mathbf{N}(\mathbf{w}+\lambda\mathbf{v})|\lambda \in \mathbb{F}_q\} \cup \{\mathbf{v}\}. \tag{6}$$

Notably, $L_{\mathbf{w}}(\mathbf{v})$ includes $\mathbf{v}$ and $\mathbf{w}$, which are not in the set given in (2) for GRM codes. Let S denote an arbitrary subset of $\mathbb{F}_q^* \cup \{\infty\}$, and $|S| = d+1$. The decoder queries $d+1$ symbols at the corresponding coordinates

$$\{\mathbf{N}(\mathbf{w}+\lambda\mathbf{v})|\lambda \in S\}.$$

Algorithm 1: Randomized query algorithm for PRM codes

Input: $\mathbf{w} \in \mathbb{P}^m$ and d

Output: $\Lambda = (\lambda_i \in \mathbb{F}_q^* \cup \{\infty\})_{i\in[d+1]}$, $L = (L_i \in \mathbf{N}(\mathbb{P}^m))_{i\in[d+1]}$ and $D = (D_i \in \mathbb{F}_q)_{i\in[d+1]}$

1 Let S, $|S| = d+1$, be an arbitrary subset of $\mathbb{F}_q^* \cup \{\infty\}$, and let Λ be constructed by ordering the elements of S in random permutation.

2 Choose $\mathbf{v} \in \mathbb{P}^m \setminus \{\mathbf{w}\}$ randomly.

3 **for** $i = 1, 2, \dots, d+1$ **do**

4 $\quad L_i = \begin{cases} \mathbf{v} & \text{if } \lambda_i = \infty \\ \mathbf{N}(\mathbf{w} + \lambda_i \mathbf{v}) & \text{otherwise} \end{cases}$

5 $\quad D_i = \begin{cases} 1 & \text{if } \lambda_i = \infty \\ \mathbf{D}(\mathbf{w} + \lambda_i \mathbf{v}) & \text{otherwise} \end{cases}$

6 **end**

7 **return** Λ, L and D.

Algorithm 2: Deterministic reconstruction algorithm for PRM codes

Input: Λ, $(e_i = F(L_i))_{i\in[d+1]}$, D and $\mathbf{w}$

Output: $F(\mathbf{w})$

1 Find a polynomial $H(X)$, $\deg H \le d$, by an ERS decoder, such that

$$H(\lambda_i) = \mathbf{D}_i^d e_i, \qquad i \in [d+1], \tag{5}$$

where $H(\infty)$ denotes the coefficient of X^d.

2 **return** $H(0)$.

We define $\mathbf{N}(\mathbf{w} + \lambda\mathbf{v}) = \mathbf{v}$ if $\lambda = \infty$. The queried symbols are then denoted as

$$\{e_\lambda = F(\mathbf{N}(\mathbf{w} + \lambda\mathbf{v})) | \lambda \in S\}. \tag{7}$$

Second, the decoder solves a single-variate polynomial $H(X)$, $\deg H \le d$, by an ERS decoder such that

$$H(\lambda) = D_\lambda^d e_\lambda, \qquad \lambda \in S, \tag{8}$$

where

$$D_\lambda = \mathbf{D}(\mathbf{w} + \mathbf{v}\cdot\lambda) \in \mathbb{F}_q. \tag{9}$$

Furthermore, if $\infty \in S$, (8) reduces to

$$H(\infty) = D_\infty^d e_\infty, \tag{10}$$

where the coefficient of X^d is equivalent to $D_\infty^d e_\infty$. After obtaining $H(X)$, the decoder returns $H(0) = F(\mathbf{w})$. The details of the decoding procedure are summarized in Algorithms 1 and 2.

A toy example for $d = 2$, $m = 2$ and $q = 3$ is given to demonstrate the decoding procedure. In this case, the arithmetic in $\mathbb{F}_3 = \{0, 1, 2\}$ is implemented with the modular arithmetic. A codeword of $\mathbf{PRM}_3(2, 2)$ is determined by a homogeneous polynomial

$$\begin{aligned} F(X_1, X_2, X_3) = & f_{200}X_1^2 + f_{020}X_2^2 + f_{002}X_3^2 \\ & + f_{110}X_1X_2 + f_{101}X_1X_3 + f_{011}X_2X_3, \end{aligned}$$

and the codeword is given by

$$\begin{aligned} (F(P) | P \in \{ & (0,0,1), (0,1,0), (0,1,1), (0,1,2), (1,0,0), \\ & (1,0,1), (1,0,2), (1,1,0), (1,1,1), (1,1,2), \\ & (1,2,0), (1,2,1), (1,2,2)\}). \end{aligned}$$

Assuming that we want to recover $F(\mathbf{w})$, $\mathbf{w} = (1, 1, 1) \in \mathbb{P}^2$. In Algorithm 1, the first step generates $S = \mathbb{F}_3^* \cup \{\infty\}$, and a permutation of S is given by $\Lambda = (2, \infty, 1)$. The second step chooses $\mathbf{v} = (1, 0, 2) \in \mathbb{P}^2 \setminus \{\mathbf{w}\}$. Then the third step generates $D = (0, 1, 2)$ and $L = \{L_i\}_{i=1}^3$, where $L_1 = (0, 1, 2)$, $L_2 = (1, 0, 2)$ and $L_3 = (1, 2, 0)$. Then the decoder queries $(e_i = F(L_i))_{i\in[3]}$, and the symbol $F(\mathbf{w})$ can be decoded by Algorithm 2.

Next, we show that the decoder meets the requirements of the perfectly smooth decoder given in Definition 2.

To verify the first requirement, we show that the set given in (7) can be considered as an evaluation of a single-variate polynomial. Thus, $F(\mathbf{w})$ can be recovered via a decoding algorithm of an ERS code, and the first requirement holds. To simplify the derivations, another formulation of (8) is presented as follows. Based on the fact that F is a homogeneous polynomial, we have

$$\begin{aligned} & F(\mathbf{N}(\mathbf{w} + \mathbf{v}\cdot\lambda)) \\ = & D_\lambda^{-d} \times F(D_\lambda \times \mathbf{N}(\mathbf{w} + \mathbf{v}\cdot\lambda)) \\ = & D_\lambda^{-d} \times F(\mathbf{w} + \mathbf{v}\cdot\lambda), \end{aligned} \tag{11}$$

where D_λ is as defined in (9). Thus, (8) can be written as

$$H(\lambda) = F(\mathbf{w} + \mathbf{v}\cdot\lambda), \qquad \lambda \in S. \tag{12}$$

Our goal is then to show that $\{F(\mathbf{w}+\mathbf{v}\cdot\lambda) | \lambda \in \mathbb{F}_q\} \cup \{F(\mathbf{v})\}$ forms an evaluation of a single-variate polynomial.

Lemma 1. *For any $\mathbf{v}, \mathbf{w} \in \mathbf{N}(\mathbb{P}^m)$, $\mathbf{v} \ne \mathbf{w}$, and any $F(\mathbf{X}) \in \mathcal{H}_d^{m+1}$, $d \le q-1$, there exists a single-variate polynomial $H(X)$, $\deg H \le d$, such that*

$$H(\lambda) = F(\mathbf{w} + \mathbf{v}\cdot\lambda), \qquad \lambda \in \mathbb{F}_q, \tag{13}$$

$$H(\infty) = F(\mathbf{v}), \tag{14}$$

where $H(\infty)$ denotes the coefficient of $H(X)$ at degree d.

Proof. The homogeneous polynomial $F(\mathbf{X})$ is written as

$$F(\mathbf{X}) = \sum_{d_0+\cdots+d_m=d} \gamma_{d_0,\dots,d_m} \prod_{j=0}^{m} X_j^{d_j}. \tag{15}$$

By plugging $(\mathbf{w}X_0 + \mathbf{v}X_1)$ into $F(\mathbf{X})$, we obtain

$$\begin{aligned} & F(\mathbf{w}X_0 + \mathbf{v}X_1) \\ = & F(w_0X_0 + v_0X_1, \dots, w_mX_0 + v_mX_1) \\ = & \sum_{d_0+\cdots+d_m=d} \gamma_{d_0,\dots,d_m} \prod_{j=0}^{m} (w_jX_0 + v_jX_1)^{d_j} \\ = & F_{\mathbf{w},\mathbf{v}}(X_0, X_1). \end{aligned} \tag{16}$$

From (16), it can be observed that $F_{\mathbf{w},\mathbf{v}}(X_0, X_1)$ is also a homogeneous polynomial of degree d in X_0 and X_1. Note that

$$(F_{\mathbf{w},\mathbf{v}}(P))_{P\in\mathbf{N}(\mathbb{P})} \in \mathbf{PRM}_q(d, 1), \tag{17}$$

which is equivalently an ERS code. Next, we show that the set of evaluation points in (17) are $d+1$ symbols in a codeword of the ERS code.

In (17), the set of evaluation points can be written as

$$\mathbf{N}(\mathbb{P}) = \{(1, \lambda) | \lambda \in \mathbb{F}_q\} \cup \{(0, 1)\}. \tag{18}$$

In the following, we show that

$$\begin{aligned} H(X) &= F_{\mathbf{w},\mathbf{v}}(1, X) \\ &= \sum_{d_0+\cdots+d_m=d} \gamma_{d_0,\ldots,d_m} \prod_{j=0}^{m} (w_j + v_j X)^{d_j} \end{aligned} \tag{19}$$

satisfies (13) and (14). It can be seen that $\deg H \leq d$. To verify (14), $(X_0, X_1) = (0, 1)$ is plugged into (16) to obtain

$$\begin{aligned} F(\mathbf{v}) &= F(v_0, \ldots, v_m) \\ &= \sum_{d_0+\cdots+d_m=d} \gamma_{d_0,\ldots,d_m} \prod_{j=0}^{m} v_j^{d_j} = F_{\mathbf{w},\mathbf{v}}(0, 1). \end{aligned} \tag{20}$$

It can be verified that $F(\mathbf{v})$ is equivalent to the coefficient of $H(X)$ at degree d. Hence, (14) holds.

To verify (13), $\lambda \in \mathbb{F}_q$ is plugged into $H(X)$ to give

$$H(\lambda) = F_{\mathbf{w},\mathbf{v}}(1, \lambda) = F(\mathbf{w} + \mathbf{v} \cdot \lambda). \tag{21}$$

This completes the proof. □

Next, the second requirement of the perfectly smooth decoder is considered. First, we show the following result.

Lemma 2. *$L_{\mathbf{w}}(\mathbf{v})$ includes $q+1$ distinct elements of $\mathbf{N}(\mathbb{P}^m)$.*

Proof. This is equivalent to showing the following two statements:

$$\mathbf{v} \notin \{\mathbf{N}(\mathbf{w} + \lambda \mathbf{v}) | \lambda \in \mathbb{F}_q\}, \tag{22}$$

$$\mathbf{N}(\mathbf{w} + \lambda_0 \mathbf{v}) \neq \mathbf{N}(\mathbf{w} + \lambda_1 \mathbf{v}) \quad \forall \lambda_0, \lambda_1 \in \mathbb{F}_q, \lambda_0 \neq \lambda_1. \tag{23}$$

These two statements can be proved by contradiction. To verify (22), assume that there exists $\lambda_0 \in \mathbb{F}_q$ such that

$$\mathbf{v} = \mathbf{N}(\mathbf{w} + \lambda_0 \mathbf{v}). \tag{24}$$

Equation (24) implies that there exists a $\gamma \in \mathbb{F}_q^*$ such that

$$\gamma \mathbf{v} = \mathbf{w} + \lambda_0 \mathbf{v} \Rightarrow \mathbf{w} = (\gamma - \lambda_0)\mathbf{v}. \tag{25}$$

Since $\mathbf{v}$ and $\mathbf{w}$ are in $\mathbf{N}(\mathbb{P}^m)$, we have $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{w} \neq \mathbf{0}$. Hence, (25) can be true only when $\gamma - \lambda_0 = 1$ and $\mathbf{w} = \mathbf{v}$, which contradicts the assumption that $\mathbf{w} \neq \mathbf{v}$. Thus, the assumption is false and (22) is proved.

To verify (23), assume that there exists $\delta_0, \delta_1 \in \mathbb{F}_q$, $\delta_0 \neq \delta_1$, such that

$$\mathbf{N}(\mathbf{w} + \delta_0 \mathbf{v}) = \mathbf{N}(\mathbf{w} + \delta_1 \mathbf{v}). \tag{26}$$

Equation (26) implies that there exists a $\gamma \in \mathbb{F}_q^*$ such that

$$\begin{aligned} &\mathbf{w} + \delta_0 \mathbf{v} = \gamma(\mathbf{w} + \delta_1 \mathbf{v}) \\ &\Rightarrow (1 - \gamma)\mathbf{w} = (\gamma\delta_1 - \delta_0)\mathbf{v}. \end{aligned} \tag{27}$$

Since $\mathbf{v}$ and $\mathbf{w}$ are in $\mathbf{N}(\mathbb{P}^m)$, we have $(1-\gamma) \neq 0$ and

$$\mathbf{w} = (1-\gamma)^{-1}(\gamma\delta_1 - \delta_0)\mathbf{v}.$$

Similar to the argument for (25), (27) is false and hence the assumption is not true. This completes the proof. □

With Lemma 2, the second requirement is shown as follows.

Lemma 3. *For any $\mathbf{w}, \mathbf{p} \in \mathbf{N}(\mathbb{P}^m)$ and $\mathbf{w} \neq \mathbf{p}$, if $L_{\mathbf{w}}(\mathbf{v})$ is constructed by choosing $\mathbf{v} \in \mathbf{N}(\mathbb{P}^m) \setminus \{\mathbf{w}\}$ uniformly, then*

$$\Pr[\mathbf{p} \in L_{\mathbf{w}}(\mathbf{v})] = (q-1)/(q^m - 1).$$

Proof. From (6),

$$\begin{aligned} L_{\mathbf{w}}(\mathbf{v}) \setminus \{\mathbf{w}\} &= \{\mathbf{N}(\mathbf{w} + \lambda\mathbf{v}) | \lambda \in \mathbb{F}_q \setminus \{0\}\} \cup \{\mathbf{v}\} \\ &= \{\mathbf{N}(\mathbf{w} + \lambda\mathbf{v}) | \lambda \in \mathbb{F}_q^*\} \cup \{\mathbf{v}\}. \end{aligned} \tag{28}$$

Thus, when $\mathbf{p} \in L_{\mathbf{w}}(\mathbf{v}) \setminus \{\mathbf{w}\}$, we have

$$\mathbf{p} = \mathbf{v}, \tag{29}$$

or

$$\mathbf{p} \in \{\mathbf{N}(\mathbf{w} + \lambda\mathbf{v}) | \lambda \in \mathbb{F}_q^*\}. \tag{30}$$

Hence, it implies that there exist $\gamma, \lambda_0 \in \mathbb{F}_q^*$ such that

$$\begin{aligned} &\gamma\mathbf{p} = \mathbf{w} + \lambda_0\mathbf{v} \\ \Rightarrow\quad &\mathbf{w} - \gamma\mathbf{p} = -\lambda_0\mathbf{v} \\ \Rightarrow\quad &\mathbf{N}(\mathbf{w} - \gamma\mathbf{p}) = \mathbf{v}. \end{aligned} \tag{31}$$

From (29) and (31), we have

$$\begin{aligned} &\mathbf{v} \in \{\mathbf{N}(\mathbf{w} - \gamma\mathbf{p}) | \gamma \in \mathbb{F}_q^*\} \cup \{\mathbf{p}\} \\ \Rightarrow\ &\mathbf{v} \in L_{\mathbf{w}}(\mathbf{p}) \setminus \{\mathbf{w}\}. \end{aligned} \tag{32}$$

Finally, from (32),

$$\mathbf{p} \in L_{\mathbf{w}}(\mathbf{v}) \setminus \{\mathbf{w}\} \Rightarrow \mathbf{v} \in L_{\mathbf{w}}(\mathbf{p}) \setminus \{\mathbf{w}\}.$$

Similarly, we can show that

$$\mathbf{v} \in L_{\mathbf{w}}(\mathbf{p}) \setminus \{\mathbf{w}\} \Rightarrow \mathbf{p} \in L_{\mathbf{w}}(\mathbf{v}) \setminus \{\mathbf{w}\}.$$

Hence, we have

$$\Pr[\mathbf{p} \in L_{\mathbf{w}}(\mathbf{v}) \setminus \{\mathbf{w}\}] = \Pr[\mathbf{v} \in L_{\mathbf{w}}(\mathbf{p}) \setminus \{\mathbf{w}\}]. \tag{33}$$

From Lemma 2, $|L_{\mathbf{w}}(\mathbf{p}) \setminus \{\mathbf{w}\}| = q$. Since $\mathbf{v}$ is chosen uniformly in $\mathbf{N}(\mathbb{P}^m) \setminus \{\mathbf{w}\}$, we have

$$\Pr[\mathbf{v} \in L_{\mathbf{w}}(\mathbf{p}) \setminus \{\mathbf{w}\}] = q/(n-1) = (q-1)/(q^m-1). \tag{34}$$

From (33) and (34), the proof is completed. □

Lemma 3 indicates that each element of $\mathbf{N}(\mathbb{P}^m) \setminus \{\mathbf{w}\}$ has equal probability to be chosen in $L_{\mathbf{w}}(\mathbf{v})$. In Algorithm 1, the order of queries is randomly permuted, and hence the jth query, $j \in [q]$, is uniformly distributed in $\mathbf{N}(\mathbb{P}^m) \setminus \{\mathbf{w}\}$. Thus, the proposed decoder meets the second requirement for a perfectly smooth decoder.

It is worth mentioning that permutation of queries (Step 1 of Algorithm 1) is necessary. If it is omitted, then the array of queries will be given by

$$(L_i = \mathbf{N}(\mathbf{w} + \omega_i\mathbf{v}))_{i \in [q]}, \tag{35}$$

where $\{\omega_i\}_{i \in [q]}$ denotes the q elements of $\mathbb{F}_q$. We prove that the list in (35) cannot satisfy the second requirement for a perfectly smooth decoder (Definition 2), although this problem has not appeared in GRM codes. To see this, let $V(\mathbf{X}) = \mathbf{N}(\mathbf{w} + \lambda\mathbf{X})$ with domain/codomain $\mathbf{N}(\mathbb{P}^m)$. The notation $\bar{w}$ denotes the smallest integer such that $\mathbf{w}[\bar{w}] \neq 0$, and $\bar{v}$ denotes the smallest integer such that $\mathbf{v}[\bar{v}] \neq 0$. Then, when $\bar{w} < \bar{v}$, there exists $\mathbf{v}' \neq \mathbf{v}$ such that $V(\mathbf{v}) = V(\mathbf{v}')$. That is, $\mathbf{v}' = \mathbf{w} + (1+\lambda)\mathbf{v}$. Since V is not bijective, the image of V is a proper subset of $\mathbf{N}(\mathbb{P}^m)$, and hence the query L_i is not uniformly distributed in $\mathbf{N}(\mathbb{P}^m)$.

IV. Discussion

In this section, the local correctabilities of PRM codes will be proved and PRM codes will be compared with other codes with similar parameters.

A. Local correctabilities of PRM codes

In this subsection, we assume that the received codewords are corrupted, as opposed to the assumption in Section III, where the codewords do not have any error.

Theorem 1. *The PRM code* $\mathbf{PRM}_q(d,m)$, $d \le q-1$, *is* $(d+1, \delta, (d+1)\delta)$*-locally correctable for all* $\delta < 1$.

Proof. The algorithm is the same as for the decoder in Section III, except that corrupted codewords are considered. Given a codeword generated by a polynomial $F(\mathbf{X})$ with δn errors and a point $\mathbf{w} \in \mathbf{N}(\mathbb{P}^m)$, the objective is to recover $F(\mathbf{w})$ by accessing at most $d+1$ symbols of $\mathbf{y}$. First, the decoder calls Algorithm 1 to obtain the list of queries L. After obtaining the symbol values corresponding to L, the decoder calls Algorithm 2 to obtain the result. Since each query is uniformly distributed, the probability that all queries are not corrupted is at least $1-(d+1)\delta$. □

Theorem 2. *The PRM code* $\mathbf{PRM}_q(d,m)$, $d \le \sigma q - 1$, *is* $(q, \delta, 2\delta/(1-\sigma))$*-locally correctable for all* $\delta < 1$.

Proof. The algorithm is a modification of the decoder in Section III. In this case, the decoder queries all elements corresponding to $L_{\mathbf{w}}(\mathbf{v}) \setminus \mathbf{w}$, and employs an ERS decoding algorithm to decode the symbol. More specifically, assume that there exists a codeword generated by $F(\mathbf{X})$. The decoder receives the codeword $\mathbf{y}$ with δn errors, and, at a point $\mathbf{w} \in \mathbf{N}(\mathbb{P}^m)$, the decoder tries to recover $F(\mathbf{w})$ by accessing at most q symbols of $\mathbf{y}$. The algorithm consists of two steps. In the first step,

$$L_{\mathbf{w}}(\mathbf{v}) := \{\mathbf{N}(\mathbf{w} + \lambda\mathbf{v}) | \lambda \in \mathbb{F}_q^* \cup \{\infty\}\}$$

is constructed by choosing $\mathbf{v} \in \mathbf{N}(\mathbb{P}^m) \setminus \mathbf{w}$ in uniform distribution. Then the codeword symbols indexed by the elements in $L_{\mathbf{w}}(\mathbf{v})$ are queried, and the queried symbols are denoted as

$$\{e_\lambda = F(\mathbf{N}(\mathbf{w} + \lambda\mathbf{v})) | \lambda \in \mathbb{F}_q^* \cup \{\infty\}\}.$$

In the second step, the local decoder tries to find a univariate polynomial $H(X)$ with $\deg H \le d$ such that

$$H(\lambda) = D_\lambda^d e_\lambda, \qquad \lambda \in \mathbb{F}_q^* \cup \{\infty\},$$

can be satisfied for as many λ as possible, where $D_\lambda = \mathbf{D}(\mathbf{w} + \mathbf{v} \cdot \lambda)$. If $H(X)$ can be determined, then the decoder outputs $H(0)$; otherwise it outputs decoding failure. For ERS decoders, it is known that if the number of unsatisfied equations (errors) is less than $\lfloor (1-\sigma)q/2 \rfloor$, the polynomial can be uniquely determined.

As each query set is individual, the probability lower bound of the successful decoding can be evaluated by the Markov inequality. This shows that the probability that $H(X)$ cannot be determined is at most $2\delta/(1-\sigma)$. □

B. Comparison

As shown in Table I, GRM codes and PRM codes are specified by three parameters (q,d,m), where q is the size of the field, d is the degree of the polynomials, and m is the number of variables. From the table, given any GRM code with parameters (q,d,m), where $d = q-2$ and $q-1$ is a prime power, we can construct a PRM code with $(q' = q-1, d, m)$, where $d = q'-1$, but a GRM code with $(q-1,d,m)$ would violate the requirement $d \le q'-2 = q-3$ from LCCs. In this case, it has been proved that the both codes have the same query complexity $d+1$ and message length $\binom{m+d}{d}$. However, the code length of a GRM code is q^m, which is always greater than the code length of a PRM code, given by $((q-1)^{m+1} - 1)/(q-2) < (1+(q-2)^{-1})(q-1)^m = \Theta((q-1)^m)$. For a PRM code, when $q=2$ (and $d=1$), the code is a Hadamard code and the proposed algorithm is actually the same as the well-known local decoder for Hadamard codes. For example, the $(q=9, d=7, m)$ GRM code over $\mathbb{F}_9$ has query complexity 8, message length $\binom{m+8}{8}$, and codeword length 9^m. In contrast, the $(q=8, d=7, m)$ PRM code over $\mathbb{F}_8$ has query complexity 8, message length $\binom{m+8}{8}$, and codeword length $(7^{m+1}-1)/6 = \Theta(7^m)$. In this case, the improving ratio of the codeword length is $\Theta((9/7)^m)$.

The improvements obtained with our proposed codes are even significant when the field size is small. By taking q to be a constant, the message length of both codes is

$$k = \binom{m+d}{d} = \mathcal{O}(m^d),$$

and hence $m = \mathcal{O}(k^{1/d})$. Table I shows that the codeword length of a GRM code is q^m, and that of a PRM code is about $(q-1)^m$. Thus, the improved ratio between the two codes is given by

$$(q/(q-1))^m = \mathcal{O}((q/(q-1))^{k^{1/d}}) = \exp(\mathcal{O}(k^{1/d})),$$

which is between polynomial and exponential.

As discussed previously, we compare PRM codes over a field of size $q-1$ and GRM codes over a field of size q. It might be argued that this comparison is unfair since the field sizes are not the same. However, as stated in [2, Sec. 8.3], the field size is not the major factor considered in the open question. Hence, when we align the query complexity and the message length of PRM codes and GRM codes, the proposed PRM codes have better performance on codeword lengths.

For low query complexities, matching vector (MV) codes [8], [9] have been invented that are shorter than GRM codes as LDCs. However, MV codes are not LCCs, and MV codes will be longer than GRM codes for query complexity $\log^c k$ with some $c > 1$. For high query complexities, GRM codes are not in this regime, since their coding rates cannot exceed $\frac{1}{2}$. In recent years, a number of codes have been proposed in this regime [29], [6], [30]. Thus, our result

TABLE I
PARAMETERS OF THE GENERALIZED REED–MULLER (GRM) CODES AND THE PROJECTIVE REED–MULLER (PRM) CODES

Codes	Restriction	Query complexity	Message length	Code length
GRM	$d \leq q-2$	$d+1$	$\binom{m+d}{d}$	q^m
PRM	$d \leq q-1$	$d+1$	$\binom{m+d}{d}$	$(q^{m+1}-1)/(q-1)$

improves the code lengths for LCCs and LDCs in the role occupied by GRM codes for low/medium query complexities.

V. CONCLUSION

We have shown that PRM codes form a family of LCCs in the regime of low query complexity. When $q = 2$ and $d = 1$, PRM codes are Hadamard codes, and the proposed local decoder is the same as the known decoder for Hadamard codes. Further, given a specified class of GRM codes, for some parameters, we have shown that there exist PRM codes that are shorter than GRM codes with the same query complexity and message length. Considering that GRM codes were the first LCCs/LDCs to be constructed, we conclude that the proposed local decoding algorithm shows that PRM codes break the oldest bound on the codeword length of LCCs/LDCs.

REFERENCES

[1] J. Katz and L. Trevisan, "On the efficiency of local decoding procedures for error-correcting codes," in *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 2000, pp. 80–86.
[2] S. Yekhanin, "Locally decodable codes," *Foundations and Trends in Theoretical Computer Science*, vol. 6, no. 3, pp. 139–255, 2012.
[3] T. Kasami, S. Lin, and W. Peterson, "New generalizations of the Reed-Muller codes–I: Primitive codes," *IEEE Transactions on Information Theory*, vol. 14, no. 2, pp. 189–199, Mar. 1968.
[4] D. E. Muller, "Application of boolean algebra to switching circuit design and to error detection," *Transactions of the I.R.E. Professional Group on Electronic Computers*, vol. EC-3, no. 3, pp. 6–12, Sept. 1954.
[5] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *Transactions of the IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 38–49, Sept. 1954.
[6] S. Kopparty, S. Saraf, and S. Yekhanin, "High-rate codes with sublinear-time decoding," *J. ACM*, vol. 61, no. 5, pp. 28:1–28:20, Sept. 2014.
[7] A. Guo, "High-rate locally correctable codes via lifting," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 6672–6682, Dec. 2016.
[8] S. Yekhanin, "Towards 3-query locally decodable codes of subexponential length," *J. ACM*, vol. 55, no. 1, pp. 1:1–1:16, Feb. 2008.
[9] Z. Dvir, P. Gopalan, and S. Yekhanin, "Matching vector codes," in *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, Oct. 2010, pp. 705–714.
[10] G. Lachaud, "Projective Reed-Muller codes," in *2nd International Colloquium on Coding Theory and Applications*, Cachan-Paris, France, 1986, pp. 125–129.
[11] A. B. Sørensen, "Projective Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 37, no. 6, pp. 1567–1576, Nov. 1991.
[12] T. P. Berger and L. de Maximy, "Cyclic projective Reed-Muller codes," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 77–81.
[13] P. Ding and J. D. Key, "Subcodes of the projective generalized Reed-Muller codes spanned by minimum-weight vectors," *Des. Codes Cryptography*, vol. 26, no. 1-3, pp. 197–211, Jun. 2002.
[14] T. P. Berger, "Automorphism groups of homogeneous and projective Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 48, no. 5, pp. 1035–1045, May 2002.
[15] S. Ballet and R. Rolland, "On low weight codewords of generalized affine and projective Reed–Muller codes," *Designs, Codes and Cryptography*, vol. 73, no. 2, pp. 271–297, 2014.
[16] C. Carvalho and V. G. Neumann, "The next-to-minimal weights of binary projective Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6300–6303, Nov. 2016.
[17] N. Nakashima and H. Matsui, "A decoding algorithm for projective Reed-Muller codes of 2-dimensional projective space with DFT," in *2014 International Symposium on Information Theory and its Applications*, Melbourne, VIC, Australia, Oct. 2014, pp. 358–362.
[18] ——, "Decoding of projective Reed-Muller codes by dividing a projective space into affine spaces," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E99.A, no. 3, pp. 733–741, 2016.
[19] L. Trevisan, "Some applications of coding theory in computational complexity," *Quaderni di Matematica*, vol. 13, pp. 347–424, 2004.
[20] B. Hemenway, R. Ostrovsky, and M. Wootters, "Local correctability of expander codes," in *Proc. of the 40th International Conference on Automata, Languages, and Programming*, vol. Part I, Berlin, Germany, July 2013, pp. 540–551.
[21] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.
[22] L. R. Welch and E. R. Berlekamp, "Error correction for algebraic block codes," US Patent 4 633 470, Dec. 1986.
[23] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969.
[24] S. J. Lin, T. Y. Al-Naffouri, Y. S. Han, and W. H. Chung, "Novel polynomial basis with fast Fourier transform and its application to Reed-Solomon erasure codes," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6284–6299, Nov. 2016.
[25] S. J. Lin, T. Y. Al-Naffouri, and Y. S. Han, "FFT algorithm for binary extension finite fields and its application to Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5343–5358, Oct. 2016.
[26] J. O. Jensen, "On decoding doubly extended Reed-Solomon codes," in *Proceedings of 1995 IEEE International Symposium on Information Theory*, Whistler, BC, Canada, Sep. 1995, pp. 280–280.
[27] L. L. Joiner and J. J. Komo, "Time domain decoding of extended reed-solomon codes," in *Proceedings of SOUTHEASTCON '96*, Tampa, FL, USA, April 1996, pp. 238–241.
[28] I. Blake and R. C. Mullin, *Finite Fields and Coding Theory*. Cambridge, Massachusetts, USA: Academic Press, 1975.
[29] A. Guo, S. Kopparty, and M. Sudan, "New affine-invariant codes from lifting," in *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, Berkeley, California, USA, 2013, pp. 529–540.
[30] S. Kopparty, O. Meir, N. Ron-Zewi, and S. Saraf, "High-rate locally-correctable and locally-testable codes with sub-polynomial query complexity," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, Cambridge, MA, USA, 2016, pp. 202–215.

Sian-Jheng Lin (M'16) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from National Chiao Tung University, Hsinchu, Taiwan, in 2004, 2006, and 2010, respectively. From 2010 to 2014, he was a postdoc with the Research Center for Information Technology Innovation, Academia Sinica. From 2014 to 2016, He was a postdoc with the Electrial Engineering Department at King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. He was a part-time lecturer at Yuanpei University from 2007 to 2008, and at Hsuan Chuang University From 2008 to 2010. He is currently a researcher with the School of Information Science and Technology at University of Science and Technology of China (USTC), Hefei, China. In recent years, his research focus on the algorithms of MDS codes and its applications to storage systems.

Yunghsiang S. Han (S'90-M'93-SM'08-F'11) was born in Taipei, Taiwan, 1962. He received B.Sc. and M.Sc. degrees in electrical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1984 and 1986, respectively, and a Ph.D. degree from the School of Computer and Information Science, Syracuse University, Syracuse, NY, in 1993. He was from 1986 to 1988 a lecturer at Ming-Hsin Engineering College, Hsinchu, Taiwan. He was a teaching assistant from 1989 to 1992, and a research associate in the School of Computer and Information Science, Syracuse University from 1992 to 1993. He was, from 1993 to 1997, an Associate Professor in the Department of Electronic Engineering at Hua Fan College of Humanities and Technology, Taipei Hsien, Taiwan. He was with the Department of Computer Science and Information Engineering at National Chi Nan University, Nantou, Taiwan from 1997 to 2004. He was promoted to Professor in 1998. He was a visiting scholar in the Department of Electrical Engineering at University of Hawaii at Manoa, HI from June to October 2001, the SUPRIA visiting research scholar in the Department of Electrical Engineering and Computer Science and CASE center at Syracuse University, NY from September 2002 to January 2004 and July 2012 to June 2013, and the visiting scholar in the Department of Electrical and Computer Engineering at University of Texas at Austin, TX from August 2008 to June 2009. He was with the Graduate Institute of Communication Engineering at National Taipei University, Taipei, Taiwan from August 2004 to July 2010. From August 2010, he is with the Department of Electrical Engineering at National Taiwan University of Science and Technology as Chair Professor. He is also a Chair Professor at National Taipei University from February 2015. His research interests are in error-control coding, wireless networks, and security.

Dr. Han was a winner of the 1994 Syracuse University Doctoral Prize and a Fellow of IEEE. One of his papers won the prestigious 2013 ACM CCS Test-of-Time Award in cybersecurity.

Nenghai Yu received his B.S. degree in 1987 from Nanjing University of Posts and Telecommunications, M.E. degree in 1992 from Tsinghua University and Ph.D. degree in 2004 from University of Science and Technology of China, where he is currently a professor. His research interests include multimedia security, multimedia information retrieval, video processing, information hiding and security, privacy and reliability in cloud computing.