Accelerated Processing for Maximum Distance Separable Codes using Composite Extension Fields

Diego Ruano*, Daniel E. Lucani† and Olav Geil‡
* IMUVA-Mathematics Research Institute and
Department of Algebra, Analysis, Geometry and Topology, University of Valladolid, Spain
†Department of Engineering and DIGIT, Aarhus University, Denmark
‡Department of Mathematical Sciences, Aalborg University, Denmark
Email: diego.ruano@uva.es, daniel.lucani@eng.au.dk , olav@math.aau.dk
Published in European Wireless 2019; 25th European Wireless Conference. Aarhus, Denmark. (2019)

***Abstract*—This paper describes a new design of Reed-Solomon (RS) codes when using composite extension fields. Our ultimate goal is to provide codes that remain Maximum Distance Separable (MDS), but that can be processed at higher speeds in the encoder and decoder. This is possible by using coefficients in the generator matrix that belong to smaller (and faster) finite fields of the composite extension and limiting the use of the larger (and slower) finite fields to a minimum. We provide formulae and an algorithm to generate such constructions starting from a Vandermonde RS generator matrix and show that even the simplest constructions, e.g., using only processing in two finite fields, can speed up processing by as much as two-fold compared to a Vandermonde RS and Cauchy RS while using the same decoding algorithm, and more than two-fold compared to other RS Cauchy and FFT-based RS.**

***Index Terms*—Reed-Solomon codes, coding theory, composite extension finite fields**

I. Introduction

Reed-Solomon (RS) codes were proposed for the first time in 1960 as a class of error-correcting codes [1]. Since then, RS codes have been used in a large number of applications from data communication to distributed storage both as an error-correcting code or as an erasure-correcting code and relying on several constructions for RS code generator matrices, e.g., Vandermonde, Cauchy. Although popular, these designs have remained relatively unchanged in the last decades and are typically associated with a significant computational costs for encoding (decoding) as the number of symbols (k) increases due to the finite field operation costs and the $O(k^2)$ ($O(k^3)$) scaling. Recent proposals focused on improving the overall scaling of encoders/encoders to $O(k\log(k))$ by using constructions based on the Fast Fourier Transform (FFT) [2], [3], [4]. However, their performance for moderate k, e.g., for communications and storage, is yet to be studied.

Given the growing challenges and upcoming demands in communication and distributed storage systems, the development of efficient constructions of RS codes that can (i) seamlessly replace older constructions without modifying the system's operation, and (ii) deliver higher encoding and decoding speeds, becomes increasingly relevant. In the context of older CPUs, Cauchy matrices were shown to be faster than Vandermonde matrices [5]. However, the use of hardware acceleration, e.g., Single Instruction Multiple Data (SIMD) operations, in Intel [6] and ARM [7] CPUs has made both approaches faster and reduced the gap between them, i.e., making them essentially equivalent in processing performance for the same implementation, e.g., consider Jerasure 2.0's performance described in [8]. Thus, novel solutions that radically redefine the core operations of the finite fields and exploit them are needed to provide the critical speed-ups.

This paper introduces a new design for RS codes with the goal to reduce the computational costs in practical systems. To achieve this, we use the fact that (i) finite fields of the form $\mathbb{F}_{2^s}$ for some s can be computed faster for smaller s [7], and that (ii) it is possible to generate efficient composite extension fields that maintain compatibility in their operations (i.e., product in small field has a direct bit-by-bit mapping to the same product performed in a larger field constructed from that smaller field) [9]. Thus, our design utilizes composite extension finite fields and proposes a deterministic algorithm to maximize the number of columns in the generator matrix that are composed solely by the smallest finite field. It then proceeds to maximize the number of columns with the second smallest finite field (an intermediate finite field), and continues until columns can only be of the largest finite field allowed. We show performance gains for decoding of as much as two-fold in SIMD capable CPUs while using the same decoding algorithm (Gaussian Elimination) as a Vandermonde RS code. Higher gains are expected in devices not capable of SIMD instructions and/or by improving the decoding algorithm as suggested in [9]. We focus on decoding performance as it is more challenging to achieve benefits than for the case of encoding. Encoding performance gains are expected to be the same or higher due to the fact that the encoding of each individual coded fragment does not depend on the encoding of other fragments.

The paper is organized as follows. Section II describes the fundamentals for generation of matrices for RS codes

using two composite extension fields and presents a simple, deterministic algorithm based on the theoretical results. Section III shows the benefits of such a construction using a C++ implementation and measuring decoding speed performance in a real device. Section IV extends our concept to generator matrices using more than two composite extension fields. Section V summarizes our contributions and future work.

II. Generator matrices for Reed-Solomon codes with composite extension fields

Let $\mathbb{F}_q$ be a finite field with $q = 2^s$ elements. A RS code $C_k \subset \mathbb{F}_q^n$, with dimension k, is the vector space generated by the evaluation of the monomials $1, X, \ldots, X^{k-1}$ at the $n = 2^s - 1$ points of $\mathbb{F}_q \setminus \{0\}$. Namely, let α be a primitive element of $\mathbb{F}_q$ and let ev : $\mathbb{F}_q[X] \to \mathbb{F}_q^n$, be given by $\text{ev}(f) = (f(\alpha^0), f(\alpha^1), \ldots, f(\alpha^{q-2}))$. Then, the RS code with dimension k is

$$C_k = \langle \{\text{ev}(X^i) : i = 0, \ldots, k-1\} \rangle$$

Since the evaluation map is injective, an $n \times k$ generator matrix is obtained by considering as columns the evaluation of a monomial at $\mathbb{F}_q \setminus \{0\}$, in this way one obtains the following generator matrix that is a Vandermonde matrix

$$G_k^V = \left(\alpha^{(i-1)(j-1)}\right)_{1 \le i \le n, 1 \le j \le k}.$$

The elements of the matrix G_k^V are in $\mathbb{F}_q$ and only the first column has all elements in $\mathbb{F}_2$. Note that a row in G_k^V is provides the coefficients for generating a coded fragment, while a column is associated to the contribution of an original fragment to the different coded fragments generated. We shall provide a different generator matrix for the RS code C_k such that as many columns as possible have all elements in $\mathbb{F}_2$ (that is, they consist in 0's and 1's). Hence, we will evaluate a different set of polynomials to obtain a different generator matrix for the same code, or equivalently, one may consider elementary transformations in the previous matrix. In particular, we can evaluate k polynomials of degree lower than k such that they are linearly independent. We consider polynomials that evaluate to $\mathbb{F}_2$, i.e., $f \in \mathbb{F}_q[X]$ such that $\text{ev}(f) \in \mathbb{F}_2^n$, which are described by cyclotomic cosets and the trace of a polynomial [10], [11]. For an integer $0 \le a < q-1$, consider the cyclotomic coset modulo $q-1$ defined by

$$I_a = \{a2^i \bmod q - 1 : i = 0, 1, 2, \ldots\}$$

. We consider $q = 2^4$ as a toy example for this section. The different cyclotomic cosets are $I_0 = \{0\}$, $I_1 = \{1, 2, 4, 8\}$, $I_3 = \{3, 6, 12, 9\}$, $I_5 = \{5, 10\}$, $I_7 = \{7, 14, 13, 11\}$. One has that $I_1 = I_2 = I_4 = I_8$ and so on. Theorem 1 provides a basis for the polynomials evaluating to $\mathbb{F}_2$ (see [12, Section 1.3] and [13, Theorem III.8]).

Theorem 1: A basis for the set of polynomials in $\mathbb{F}_q[X]$ evaluating to $\mathbb{F}_2$ is

$$\bigcup_{I_a} \left\{ f_{I_a, \beta^j} : j \in \{0, \ldots, \#I_a - 1\} \right\},$$

where $\beta = \alpha^{(2^s-1)/(2^{\eta_a}-1)}$, with $\eta_a = \#I_a$, i.e. a primitive element of $\mathbb{F}_{2^{\eta_a}} \subseteq \mathbb{F}_{2^s}$, and $f_{I_a,\beta} = \beta X^a + \beta^2 X^{2a} + \cdots + \beta^{2^{\eta_a - 1}} X^{2^{\eta_a - 1} a}$.

We consider again $q = 2^4$, we provide some of the polynomials given by the previous theorem: one has a cyclotomic coset with one element $I_0 = \{0\}$ and its associated polynomial is $1 = X^0$, that trivially evaluates to $\mathbb{F}_2$. The polynomials, associated to the cyclotomic coset $I_1 = \{1, 2, 4, 8\}$, are $\{f_{I_1,1}, f_{I_1,\alpha}, f_{I_1,\alpha^2}, f_{I_1,\alpha^3}\}$. They are linearly independent and evaluate to $\mathbb{F}_2$:

$$\begin{aligned} f_{I_1,1} &= \text{ev}(X + X^2 + X^4 + X^8) \\ f_{I_1,\alpha} &= \text{ev}(\alpha X + \alpha^2 X^2 + \alpha^4 X^4 + \alpha^8 X^8) \\ f_{I_1,\alpha^2} &= \text{ev}(\alpha^2 X + \alpha^4 X^2 + \alpha^8 X^4 + \alpha X^8) \\ f_{I_1,\alpha^3} &= \text{ev}(\alpha^3 X + \alpha^6 X^2 + \alpha^{12} X^4 + \alpha^9 X^8). \end{aligned}$$

For $I_5 = \{5, 10\}$, we have that $\{f_{I_5,1}, f_{I_5,\alpha^5}\}$ are its associated polynomials that evaluate to $\mathbb{F}_2$, since $\alpha^5 = \alpha^{((16-1)/(4-1))}$. Namely,

$$\begin{aligned} f_{I_5,1} &= X^5 + X^{10} \\ f_{I_5,\alpha^5} &= \alpha^5 X^5 + \alpha^{10} X^{10}. \end{aligned}$$

Continuing this way, we get from I_3 four polynomials of degree 12, as $I_3 = \{3, 6, 9, 12\}$, and from I_7, another four polynomials of degree 14, as $I_7 = \{7, 11, 13, 14\}$, which evaluate to $\mathbb{F}_2$. With these polynomials, we can construct the RS code C_9 with dimension 9 and length 15 over $\mathbb{F}_{16}$. By Theorem 1 we can evaluate $f_{I_1,1}, f_{I_1,\alpha}, f_{I_1,\alpha^2}, f_{I_1,\alpha^3}$, instead of X, X^2, X^4, X^8, to construct the generator matrix since $\{1, 2, 4, 8\} \subset \{0, \ldots, 8\}$. In this way, a generator matrix of the Reed-Solomon code C_9 consists of the columns $\text{ev}(1)$, $\text{ev}(f_{I_1,1})$, $\text{ev}(f_{I_1,\alpha})$, $\text{ev}(f_{I_1,\alpha^2})$, $\text{ev}(f_{I_1,\alpha^3})$, $\text{ev}(X^3)$, $\text{ev}(X^5)$, $\text{ev}(X^6)$ and $\text{ev}(X^7)$. Note that these polynomials will evaluate to codewords that are linearly independent since some of them have different degree and the ones with the same degree are linearly independent from each other by Theorem 1 and linearly independent from the remaining polynomials by the definition of f_{I_a,α^i}. The elements of the first 5 columns of the

generator matrix are in $\mathbb{F}_2$.

$$G_9 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & \alpha^3 & \alpha^5 & \alpha^6 & \alpha^7 \\ 1 & 0 & 1 & 0 & 0 & \alpha^6 & \alpha^{10} & \alpha^{12} & \alpha^{14} \\ 1 & 1 & 0 & 0 & 1 & \alpha^9 & 1 & \alpha^3 & \alpha^6 \\ 1 & 0 & 0 & 1 & 1 & \alpha^{12} & \alpha^5 & \alpha^9 & \alpha^{13} \\ 1 & 0 & 1 & 1 & 0 & 1 & \alpha^{10} & 1 & \alpha^5 \\ 1 & 1 & 1 & 0 & 1 & \alpha^3 & 1 & \alpha^6 & \alpha^{12} \\ 1 & 1 & 0 & 1 & 0 & \alpha^6 & \alpha^5 & \alpha^{12} & \alpha^4 \\ 1 & 0 & 1 & 0 & 1 & \alpha^9 & \alpha^{10} & \alpha^3 & \alpha^{11} \\ 1 & 1 & 0 & 1 & 1 & \alpha^{12} & 1 & \alpha^9 & \alpha^3 \\ 1 & 0 & 1 & 1 & 1 & 1 & \alpha^5 & 1 & \alpha^{10} \\ 1 & 1 & 1 & 1 & 1 & \alpha^3 & \alpha^{10} & \alpha^6 & \alpha^2 \\ 1 & 1 & 1 & 1 & 0 & \alpha^6 & 1 & \alpha^{12} & \alpha^9 \\ 1 & 1 & 1 & 0 & 0 & \alpha^9 & \alpha^5 & \alpha^3 & \alpha \\ 1 & 1 & 0 & 0 & 0 & \alpha^{12} & \alpha^{10} & \alpha^9 & \alpha^8 \end{bmatrix}$$

Let C_{11} be the RS code with dimension 11 and length 15 over $\mathbb{F}_{16}$. The generator matrix of this code can be obtained by considering $\{\text{ev}(X^i) : i = 0, \ldots, 10\}$. As before, by Theorem 1, we can consider $f_{I_1,1}$, $f_{I_1,\alpha}$, f_{I_1,α^2}, f_{I_1,α^3} instead of X, X^2, X^4, X^8 since $\{1, 2, 4, 8\} \subset \{0, \ldots, 10\}$. Furthermore, we can now consider $f_{I_5,1}, f_{I_5,\alpha^5}$ instead of X^5, X^{10}. In this way, by the same reason as before, a generator matrix of the RS code C_{11} is given by $\text{ev}(1)$, $\text{ev}(f_{I_1,1})$, $\text{ev}(f_{I_1,\alpha})$, $\text{ev}(f_{I_1,\alpha^2})$, $\text{ev}(f_{I_1,\alpha^3})$, $f_{I_5,1}$, f_{I_5,α^5}, $\text{ev}(X^3)$, $\text{ev}(X^6)$, $\text{ev}(X^7)$, $\text{ev}(X^9)$. The elements of the first 7 columns of the generator matrix are in $\mathbb{F}_2$.

$$G_{11} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & \alpha^3 & \alpha^6 & \alpha^7 & \alpha^9 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & \alpha^6 & \alpha^{12} & \alpha^{14} & \alpha^3 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & \alpha^9 & \alpha^3 & \alpha^6 & \alpha^{12} \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & \alpha^{12} & \alpha^9 & \alpha^{13} & \alpha^6 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & \alpha^5 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & \alpha^3 & \alpha^6 & \alpha^{12} & \alpha^9 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & \alpha^6 & \alpha^{12} & \alpha^4 & \alpha^3 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & \alpha^9 & \alpha^3 & \alpha^{11} & \alpha^{12} \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & \alpha^{12} & \alpha^9 & \alpha^3 & \alpha^6 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \alpha^{10} & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^9 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & \alpha^6 & \alpha^{12} & \alpha^9 & \alpha^3 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & \alpha^9 & \alpha^3 & \alpha & \alpha^{12} \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & \alpha^{12} & \alpha^9 & \alpha^8 & \alpha^6 \end{bmatrix}$$

Analogously, for the RS code C_{13} with dimension 13 and length 15 over $\mathbb{F}_{16}$, we can construct a generator matrix with 11 columns with all elements in $\mathbb{F}_2$. In general, for a RS code C_k with dimension k over $\mathbb{F}_q$, one can obtain a generator matrix with

$$\sum_{I_a \subseteq \{0,\ldots k-1\}} \#I_a$$

columns in $\mathbb{F}_2$. Note that, by Theorem 1, such a number of columns is optimal. For RS codes over $\mathbb{F}_{16}$, Table I shows how many columns can be obtained in $\mathbb{F}_2$.

Dimension k	1	2	3	4	5	6	7	8
#Columns in $\mathbb{F}_2$	1	1	1	1	1	1	1	1
Dimension k	9	10	11	12	13	14	15	
#Columns in $\mathbb{F}_2$	5	5	7	7	11	11	15	

TABLE I: Optimal number of columns in $\mathbb{F}_2$ for various k for RS codes over $\mathbb{F}_{16}$

We have described how to construct a generator matrix by evaluating the polynomials given in Theorem 1. Alternatively, one can perform elementary operations in the Vandermonde matrix G_k^V: let g_i be the column i of the matrix G_k^V, for $i = 1, \ldots, k$. The polynomial $f_{I_a,\beta} = \beta X^a + \beta^2 X^{2a} + \cdots + \beta^{2^{\#I_a-1}} X^{2^{\#I_a-1}a}$ indicates that the column i of the matrix G_k is given by a linear combination of columns of G_k^V:

$$\sum_{j=0}^{\#I_a-1} \beta^{2^j} g_{2^j a+1 \bmod q-1}$$

III. Implementation and experimental results

To test the potential benefits of this approach, we use the *kodo* C++ library [14] in order to implement and measure performance of both RS Vandermonde and RS Cauchy on two different finite fields and our new constructions with two finite fields. We also use the *longhair*[1] and *leopard*[2] libraries to compare our performance against an RS Cauchy optimized implementation (SIMD capable) and an FFT-based RS implementation, respectively. The latter is based on the formulation provided in [4]. For a fair comparison, we consider non-systematic codes. This means that the RS Cauchy decoding tests will be considered using only coded packets, which limits the generation size to 128 packets. The performance evaluation is carried out on an OSX machine (2015) with a 2.2 GHz Intel Core i7 processor with Single Instruction Multiple Data (SIMD) support. Specifically, the CPU provides SSSE3, SSSE4, and AVX2 support, which allows to perform hardware acceleration of finite field operations. We measured the decoding speed of both approaches using the same decoding algorithm in order to compare fairly. However, improved algorithms for our construction can be developed using the intuition proposed in [9] due to the problem's structure.

Figure 1 shows the performance of our approach when using a combination of finite fields $\mathbb{F}_2$ and $\mathbb{F}_{2^4}$ compared to a RS Cauchy code and RS Vandermonde code using $\mathbb{F}_{2^4}$. Unfortunately, *longhair* and *leopard* do not provide support for this finite field. As expected, the performance difference between RS Vandermonde and our proposed code starts when the number of original packets (i.e., dimension) is $k \geq q/2 + 1 = 2^4/2 + 1 = 9$, where q is the size of the larger field. Our results show that gains of up to 37 %

[1]Longhair: https://github.com/catid/longhair

[2]Leopard: https://github.com/catid/leopard

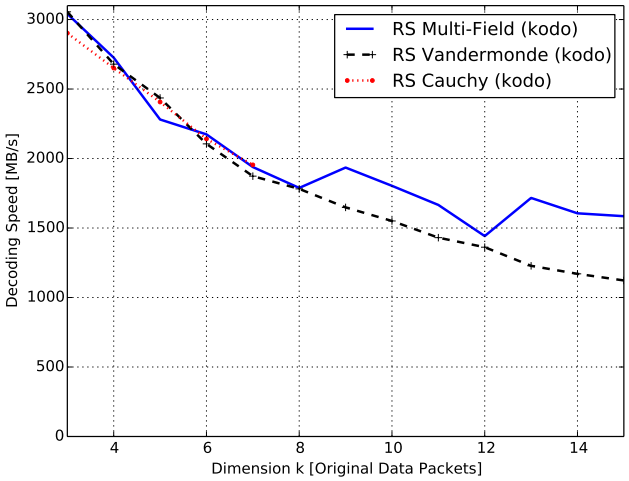

Fig. 1: Decoding speeds of RS Vandermonde ($\mathbb{F}_{2^4}$), RS Cauchy ($\mathbb{F}_{2^4}$) and RS Multi-Field using $\mathbb{F}_2$ and $\mathbb{F}_{2^4}$

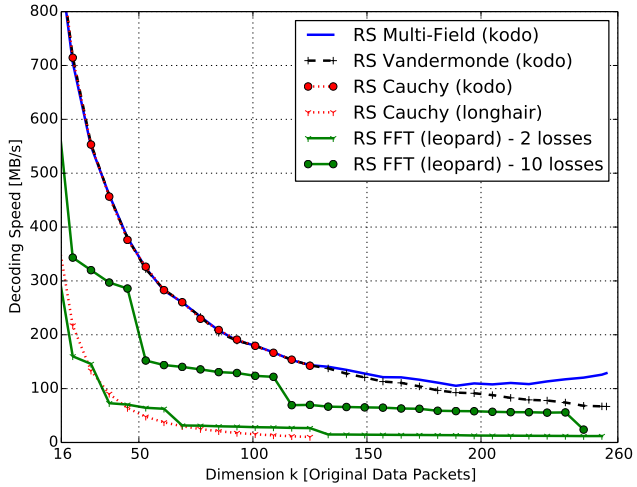

Fig. 2: Decoding speeds of Various RS schemes using $\mathbb{F}_{2^8}$ and RS Multi-Field using $\mathbb{F}_2$ and $\mathbb{F}_{2^8}$

are attainable by simply changing the coding coefficients as described in Section II. Note that the RS Cauchy performs similarly to RS Vandermonde due to the use of similar SIMD instructions for the product operations. Figure 1 also shows that the decoding speed for a generation size (dimension) of $k \geq 12$ is kept above 1440 MB/s with our approach, while the standard RS approach continues to decrease in decoding speed for the same region. This is related to the increased number of 1 and 0 coefficients introduced with our scheme.

Figure 2 shows the performance of our approach when using a combination of finite fields $\mathbb{F}_2$ and $\mathbb{F}_{2^8}$, similar to Figure 1. In this case, a dimension $k \geq q/2 + 1 = 256/2 + 1 = 129$ is necessary to see differences between our scheme and RS. In this case, we compare to RS Vandermonde and Cauchy implementations from *kodo*, but also *longhair*'s Cauchy implementation and *leopard*'s FFT-based RS decoder. Our results show not only that the decoding speed of our approach can be almost twice as high than RS Cauchy or Vandermonde from *kodo* and larger than the other schemes, but also that the decoding speed increases when $k > 200$. The latter means that the decoding speed is predictably never worse than 100 MB/s for this particular device using our approach, which provides some system design advantages at the time of planning and allocating resources for decoding purposes, which may rely on worst case performance of the module. The results for the RS Vandermonde and Cauchy and our scheme are not dependent on the loss patterns. However, Figure 2 shows that *leopard*'s decoder depends heavily on the generated redundancy and losses, e.g., there is a performance gap of five times between the performance for two losses and ten losses, with the former being slower. This makes the performance of the FFT-based RS less predictable than the other schemes, with worst case scenarios being an order of magnitude slower than other schemes. The fact that the performance is lower when fewer losses occur, may limit its use in some applications, e.g., distributed storage, where the system is likely to repair damaged fragments after a few losses.

As an important remark, these results are for a computer that exploits hardware acceleration (SIMD instructions). As measured in [7], the gap between a hardware-accelerated implementation and an implementation without it is an order of magnitude. This means that the gains are expected to be much larger in devices without these capabilities, e.g., lower end smart phones, IoT devices, as the cost of processing large fields is significantly higher without SIMD functionalities with respect to processing smaller finite fields.

IV. Generator Matrices with Columns in an Intermediate Field

We considered in Section II generator matrices for RS codes with as many columns with all elements in $\mathbb{F}_2$ as possible. Although we did not consider it in our experimental results yet, one may modify the matrices obtained in Section II to construct generator matrices where some of the columns, that do not have all elements in $\mathbb{F}_2$, have all their elements in an intermediate field, $\mathbb{F}_{2^r}$, with $r \mid s$, since $\mathbb{F}_2 \subset \mathbb{F}_{2^r} \subset \mathbb{F}_{2^s}$. This modification will allow the encoders and decoders to operate in an intermediate field that is faster than the largest field, thus increasing further the encoding/decoding speeds in the system. We proceed as in Section II, considering 2^r-cyclotomic cosets modulo $2^s - 1$:

$$I_a^{2^r} = \{a2^{ri} \bmod 2^s - 1 : i = 0, 1, 2, \ldots\}$$

and rewrite Theorem 1 as follows.

Theorem 2: A basis for the set of polynomials in $\mathbb{F}_q[X]$ evaluating to $\mathbb{F}_{2^r}$, with $r \mid s$, is

$$\bigcup_{I_a^{2^r}} \left\{ f_{I_a^{2^r},\beta^j} : j \in \{0, \ldots, \#I_a^{2^r} - 1\} \right\},$$

Dim. k	1	2	3	4	5	6	7	8
#Cols. in $\mathbb{F}_2^n$	1	1	1	1	1	1	1	1
#Cols. in $\mathbb{F}_4^n \setminus \mathbb{F}_2^n$	0	0	0	0	2	2	2	2
Dim. k	9	10	11	12	13	14	15	
#Cols. in $\mathbb{F}_2^n$	5	5	7	7	11	11	15	
#Cols. in $\mathbb{F}_4^n \setminus \mathbb{F}_2^n$	0	2	2	2	0	2	0	

TABLE II: Optimal number of columns in $\mathbb{F}_2$ and $\mathbb{F}_4$ for various k for RS codes over $\mathbb{F}_{16}$

where $\beta = \alpha^{(2^s-1)/(2^{\eta_a}-1)}$, with $\eta_a = \#I_a^{2^r}$, i.e. a primitive element of $\mathbb{F}_{2^{\eta_a}} \subseteq \mathbb{F}_{2^s}$, and $f_{I_a^{2^r},\beta} = \beta X^a + \beta^{2^r} X^{2^r a} + \cdots + \beta^{2^{r\eta_a - 1}} X^{2^{r\eta_a-1} a}$.

For example, for $q = 2^4$, we have that $\mathbb{F}_2 \subset \mathbb{F}_4 \subset \mathbb{F}_{16}$. The different 4-cyclotomic cosets are $I_0^4 = \{0\}$, $I_1^4 = \{1, 4\}$, $I_2^4 = \{2, 8\}$, $I_3^4 = \{3, 12\}$, $I_5^4 = \{5, 10\}$, $I_6^4 = \{6, 9\}$, $I_7^4 = \{7, 13\}$, $I_{11}^4 = \{11, 14\}$. Hence, one can obtain generator matrices for RS codes with the following number of columns in $\mathbb{F}_2^n$ and $\mathbb{F}_4^n \setminus \mathbb{F}_2^n$, as in Table II.

V. Conclusion

This paper presented a novel construction for RS codes that is compatible with multiple composite extension fields, e.g., [9], and that has a shown potential to reduce practical computation costs in real devices. Our measurement results showed that up to twice the speed in decoding is possible in hardware-accelerated devices. This is possible even limiting our approach to the use of two finite fields, which can be deployed in current systems by simply changing the coding coefficients of the coding matrix and without modifications of the underlying finite field constructions. Further gains are expected when using our constructions for multiple composite fields and for computationally limited devices without hardware-acceleration capabilities, i.e., devices that require significantly more effort to compute larger finite fields. Future work will consider an extensive measurement campaign to understand the benefits to different classes of commercial devices. In particular, we expect lower end devices without hardware optimization (e.g., SIMD instructions) to benefit further from our new RS constructions. Future work will consider the design of systematic RS codes using composite extension fields, which is of interest in distributed storage applications.

Acknowledgment

This research is supported in part by the Spanish Ministry of Economy/FEDER, grant No. MTM2015-65764-C3-2-P and No. RYC-2016-20208 (AEI/FSE/UE), the SCALE-IoT project granted by the Danish Council for Independent Research (Grant No. DFF 7026-00042B), the Danish Council for Independent Research grant No. DFF-4002-00367, the AUFF Starting Grant Project AUFF-2017-FLS-7-1, and Aarhus Universitys DIGIT Centre.

References

[1] 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.

[2] 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 Trans. Info. Theory*, vol. 62, no. 10, pp. 5343–5358, 2016.

[3] I. Reed, R. Scholtz, T.-K. Truong, and L. Welch, "The fast decoding of Reed-Solomon codes using Fermat theoretic transforms and continued fractions," *IEEE Trans. Info. Theory*, vol. 24, no. 1, pp. 100–106, Jan. 1978.

[4] 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 Trans.on Info. Theory*, vol. 62, no. 11, pp. 6284–6299, Nov 2016.

[5] J. S. Plank, J. Luo, C. D. Schuman, L. Xu, and Z. Wilcox-O'Hearn, "A performance evaluation and examination of open-source erasure coding libraries for storage," in *FAST-2009: 7th Usenix Conf. on File and Storage Tech.*, February 2009.

[6] J. S. Plank, K. M. Greenan, and E. L. Miller, "Screaming fast Galois Field arithmetic using Intel SIMD instructions," in *FAST-2013: 11th Usenix Conf. on File and Storage Techn.*, San Jose, February 2013.

[7] C. W. Sorensen, A. Paramanathan, J. A. Cabrera, M. V. Pedersen, D. E. Lucani, and F. H. P. Fitzek, "Leaner and meaner: Network coding in simd enabled commercial devices," in *IEEE Wireless Comm. and Networking Conf.*, April 2016, pp. 1–6.

[8] D. Burihabwa, P. Felber, H. Mercier, and V. Schiavoni, *A Performance Evaluation of Erasure Coding Libraries for Cloud-Based Data Stores*. Berlin, Heidelberg: Springer, 2016, pp. 160–173.

[9] J. Heide and D. Lucani, "Composite extension finite fields for low overhead network coding: Telescopic codes," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 4505–4510.

[10] P. Delsarte, "On subfield subcodes of modified Reed-Solomon codes," *IEEE Trans. on Info. Theory*, vol. IT-21, no. 5, pp. 575–576, 1975.

[11] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes. I.* North-Holland Publishing Co., Amsterdam-New York-Oxford, 1977, north-Holland Mathematical Library, Vol. 16.

[12] L. Rédei, *Lacunary polynomials over finite fields*. North-Holland, 1973.

[13] F. Hernando, M. E. O'Sullivan, E. Popovici, and S. Srivastava, "Subfield-subcodes of generalized toric codes," in *IEEE Int. Symposium on Info. Theory*, June 2010, pp. 1125–1129.

[14] M. V. Pedersen, J. Heide, and F. H. P. Fitzek, *Kodo: An Open and Research Oriented Network Coding Library*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 145–152.