Chapter 8

Reed-Solomon codes

In the previous chapter we discussed the properties of finite fields, and showed that there exists an essentially unique finite field $\mathbb{F}_q$ with $q = p^m$ elements for any prime p and integer $m \geq 1$.

We now consider (n, k, d) linear codes over a finite field $\mathbb{F}_q$. The most useful such codes are the Reed-Solomon (RS) codes (1960), whose length is limited to $n \leq q$, but which have arbitrary dimension $k, 1 \leq k \leq n$, and which have a minimum distance of $d = n - k + 1$. This meets a certain fundamental bound called the Singleton bound. RS codes are easily decoded, and have a wide range of uses.

We also briefly mention binary Bose-Chaudhuri-Hocquenghem (BCH) codes (1959), which are binary derivatives of RS codes. The parameters (n, k, d) of BCH codes are sometimes slightly better than those of Reed-Muller (RM) codes.

8.1 Linear codes over finite fields

An (n, k) linear code $\mathcal{C}$ over a finite field $\mathbb{F}_q$ is a k-dimensional subspace of the vector space $\mathbb{F}_q^n$ of all n-tuples over $\mathbb{F}_q$. For $q = 2$, this reduces to our previous definition of binary linear codes.

A subset $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a subspace if it is closed under componentwise addition (has the group property) as well as under multiplication by scalars (elements of $\mathbb{F}_q$). A linear code $\mathcal{C}$ thus necessarily contains the all-zero n-tuple $\mathbf{0}$.

Any k linearly independent codewords $(\mathbf{g}_1, \ldots, \mathbf{g}_k)$ generate $\mathcal{C}$, in the sense that

$$\mathcal{C} = \{\sum_{j=1}^{k} a_j \mathbf{g}_j, a_j \in \mathbb{F}_q, 1 \leq j \leq k\}.$$

Thus $\mathcal{C}$ has q^k distinct codewords.

Example 1. The following nine 4-tuples over $\mathbb{F}_3$ form a $(4, 2, 3)$ linear code over the ternary field $\mathbb{F}_3 = \{0, 1, 2\}$, with generators $\mathbf{g}_1 = (1110)$ and $\mathbf{g}_2 = (0121)$:

$$\mathcal{C} = \{0000, 1110, 2220, 0121, 1201, 2011, 0212, 1022, 2102\}.$$

By the group property, $\mathcal{C} = \mathcal{C} - \mathbf{c}$ for any codeword $\mathbf{c} \in \mathcal{C}$, so the set of Hamming distances between any codeword $\mathbf{c} \in \mathcal{C}$ and all other codewords is independent of $\mathbf{c}$. The minimum Hamming distance d between codewords in a linear code $\mathcal{C}$ is thus equal to the minimum Hamming weight of any nonzero codeword (the minimum distance between $\mathbf{0}$ and any other codeword). An (n, k) linear code with minimum Hamming distance d is called an (n, k, d) linear code.

More generally, the group property shows that the number of codewords at Hamming distance w from any codeword $\mathbf{c} \in \mathcal{C}$ is the number N_w of codewords of Hamming weight w in $\mathcal{C}$.

Much of classical algebraic coding theory has been devoted to optimizing the parameters (n, k, d); *i.e.,* maximizing the size q^k of the code for a given length n, minimum distance d and field $\mathbb{F}_q$, or maximizing d for a given n, k and q. The practical motivation for this research has been to maximize the guaranteed error-correction power of the code. Because Hamming distance is a metric satisfying the triangle inequality, a code with Hamming distance d is guaranteed to correct t symbol errors whenever $2t < d$, or in fact to correct t errors and s erasures whenever $2t + s < d$. This elementary metric is not the whole story with regard to performance on an AWGN channel, nor does it take into account decoding complexity; however, it is a good first measure of code power.

8.2 The Singleton bound and MDS codes

The Singleton bound is the most fundamental bound on the parameters (n, k, d) over any field: $k + d \leq n + 1$. A code that meets this bound with equality is called "maximum distance separable" (MDS).

The only binary codes that are MDS are the trivial $(n, n, 1)$ universe codes, the $(n, n-1, 2)$ single-parity-check (SPC) codes, and the $(n, 1, n)$ repetition codes. However, over nonbinary fields, there exist much less trivial MDS codes, notably the Reed-Solomon codes and their close relations.

Let $\mathcal{C} \subseteq \mathcal{A}^n$ be any set of $|\mathcal{C}|$ n-tuples over any symbol alphabet $\mathcal{A}$ of size $q = |\mathcal{A}|$, and let the minimum Hamming distance between codewords in $\mathcal{C}$ be d. Then $\mathcal{C}$ should be able to correct any set of $s = d - 1$ erasures; *i.e.,* after puncturing $s = d - 1$ of the coordinates, the codewords must still all differ in the remaining $n - s = n - d + 1$ coordinates. Therefore there can be at most q^{n-d+1} codewords in $\mathcal{C}$:

Theorem 8.1 (Singleton bound) *A code $\mathcal{C}$ of length n with minimum Hamming distance d over an alphabet of size $q = |\mathcal{A}|$ can have at most $|\mathcal{C}| \leq q^{n-d+1}$ codewords. Equality holds if and only if the codewords run through all q^k possible k-tuples in every set of $k = n-d+1$ coordinates.*

Note that the Singleton bound holds also for nonlinear codes. Any code that meets the Singleton bound with equality is called an MDS code.

Given a (possibly nonlinear) code $\mathcal{C}$ with q^k codewords over an alphabet of size q, a set of k coordinates is called an *information set* of $\mathcal{C}$ if the codewords run through all q^k possible k-tuples in that set of coordinates; *i.e.,* if there is a unique codeword associated with every possible set of symbol values in that set of coordinates. In other words, we may freely specify the symbols in these coordinates, and then the remainder of the codeword is uniquely specified. The case of equality may therefore be restated as follows:

Corollary 8.2 (Information sets of MDS codes) *$\mathcal{C}$ is an MDS code if and only if every subset of $k = n - d + 1$ coordinates is an information set of $\mathcal{C}$.*

If $\mathcal{C}$ is a linear (n, k, d) code over a finite field $\mathbb{F}_q$, then its size is $|\mathcal{C}| = q^k$. Therefore:

Corollary 8.3 (Singleton bound for linear codes) *A linear (n, k, d) code $\mathcal{C}$ over any finite field $\mathbb{F}_q$ has $k \leq n-d+1$. Equality holds if and only if every set of k coordinates is an information set of $\mathcal{C}$.*

Equivalently, $d + k \leq n + 1$, or $d \leq n - k + 1$.

The conditions governing MDS codes are so stringent that the weight distribution of a linear (n, k, d) MDS code over $\mathbb{F}_q$ is completely determined by the parameters n, k, d and q. We will show how to derive N_d and N_{d+1}; the number of codewords N_w of higher weights w may be derived similarly.

Theorem 8.4 (N_d for MDS codes) *An (n, k, d) MDS code over $\mathbb{F}_q$ has $q - 1$ codewords that have weight d in every subset of d coordinates. The total number of codewords of weight d is thus*

$$N_d = \binom{n}{d}(q-1).$$

Proof. Given a subset of $d = n - k + 1$ coordinates, consider the information set consisting of the complementary $n - d = k - 1$ coordinates plus any one of the given coordinates. Fix the symbol values in the $k - 1$ complementary coordinates to 0, and let the symbol values in the remaining information coordinate run through all q symbols in $\mathbb{F}_q$. This must yield q different codewords with all zeroes in the $n - d$ complementary coordinates. One of these codewords must be the all-zero codeword $\mathbf{0}$. The remaining $q - 1$ codewords have weight at most d; but since the minimum nonzero weight is d, all must have weight d. The formula for N_d then follows from the fact that there are $\binom{n}{d}$ different subsets of d coordinates. □

Exercise 1. Show that the number of codewords of weight $d + 1$ in an (n, k, d) linear MDS code is

$$N_{d+1} = \binom{n}{d+1}\left((q^2-1) - \binom{d+1}{d}(q-1)\right),$$

where the first term in parentheses represents the number of codewords with weight $\geq d$ in any subset of $d+1$ coordinates, and the second term represents the number of codewords with weight equal to d. □

Exercise 2. Compute the number of codewords of weights 2 and 3 in an $(n, n - 1, 2)$ SPC code over $\mathbb{F}_2$. Compute the number of codewords of weights 2 and 3 in an $(n, n - 1, 2)$ linear code over $\mathbb{F}_3$. Compute the number of codewords of weights 3 and 4 in a $(4, 2, 3)$ linear code over $\mathbb{F}_3$. □

8.3 Reed-Solomon (RS) Codes

For any code parameters $n, k, d = n - k + 1$ $(1 \leq k \leq n)$ and finite field $\mathbb{F}_q$, there exists a linear (n, k, d) RS code over $\mathbb{F}_q$, as long as $n \leq q + 1$. Thus RS codes form a very large class of MDS codes. In fact, for any n, $1 < k < n - 1$, and q for which an MDS code is known to exist, there also exists an RS code (or doubly, or triply, extended RS code) with the same parameters.

Binary RS codes have length at most $n \leq q + 1 = 3$, and therefore are not very interesting. When we speak of RS codes, we are usually thinking of nonbinary RS codes over $\mathbb{F}_q$, where q may be relatively large.

8.3.1 RS codes as evaluation codes

The most natural definition of RS codes is for length $n = q$. The definition is in terms of a certain evaluation map from the set $(\mathbb{F}_q)^k$ of all k-tuples over $\mathbb{F}_q$ to the set $(\mathbb{F}_q)^n$ of n-tuples over $\mathbb{F}_q$.

Let the k information symbols be denoted by $(f_0, f_1, \ldots, f_{k-1})$, where $f_j \in \mathbb{F}_q, 0 \leq j \leq k-1$. Let $f(z) = f_0 + f_1 z + \cdots + f_{k-1} z^{k-1} \in \mathbb{F}_q[z]$ be the corresponding polynomial in the indeterminate z. (We use z for the indeterminate here to avoid confusion with the indeterminate x used in the polynomials representing the elements of $\mathbb{F}_q$.) Thus $f(z)$ is one of the q^k polynomials over $\mathbb{F}_q$ of degree less than k.

Let $\beta_1, \beta_2, \ldots \beta_q$ be the q different elements of $\mathbb{F}_q$ arranged in some arbitrary order. The most convenient arrangement is $\beta_1 = 0, \beta_2 = 1, \beta_3 = \alpha, \ldots, \beta_j = \alpha^{j-2}, \ldots, \beta_q = \alpha^{q-2} = \alpha^{-1}$, where α is a primitive element of $\mathbb{F}_q$.

The information polynomial $f(z)$ is then mapped into the n-tuple $(f(\beta_1), f(\beta_2), \ldots, f(\beta_q))$ over $\mathbb{F}_q$, whose components $f(\beta_i)$ are equal to the evaluations of the polynomial $f(z)$ at each field element $\beta_i \in \mathbb{F}_q$:

$$f(\beta_i) = \sum_{j=0}^{k-1} f_j \beta_i^j \in \mathbb{F}_q, \quad 1 \leq i \leq q,$$

where we use the convention $0^0 = 1$. Note that it is the polynomial $f(z)$ that varies from codeword to codeword; the "symbol locators" $0, 1, \beta_3 = \alpha, \ldots, \beta_q = \alpha^{-1}$ are fixed. The code generators may thus be taken as the polynomials $g_j(z) = z^j, 0 \leq j \leq k-1$, or as the n-tuples

$$\begin{aligned}
\mathbf{g}_0 &= (1, 1, 1, \ldots, 1) \\
\mathbf{g}_1 &= (0, 1, \alpha, \ldots, \alpha^{-1}) \\
\mathbf{g}_2 &= (0, 1, \alpha^2, \ldots, \alpha^{-2}) \\
&\ldots \\
\mathbf{g}_{k-1} &= (0, 1, \alpha^{k-1}, \ldots, \alpha^{-(k-1)})
\end{aligned}$$

Theorem 8.5 (RS codes) *The q^k q-tuples generated by the mapping $f(z) \mapsto \{f(\beta_i), \beta_i \in \mathbb{F}_q\}$ as the polynomial $f(z)$ ranges over all q^k polynomials over $\mathbb{F}_q$ of degree less than k form a linear $(n = q, k, d = n - k + 1)$ MDS code over $\mathbb{F}_q$.*

Proof. The code is linear because the sum of the codewords corresponding to two polynomials $f_1(z)$ and $f_2(z)$ is the codeword corresponding to the polynomial $f_1(z) + f_2(z)$, and the multiple of the codeword corresponding to $f(z)$ by $\beta \in \mathbb{F}_q$ is the codeword corresponding to the polynomial $\beta f(z)$.

A codeword has a zero symbol in the coordinate corresponding to β_i if and only if $f(\beta_i) = 0$; *i.e.,* if and only if β_i is a root of the equation $f(z) = 0$. By the fundamental theorem of algebra (Theorem 7.9), if $f(z) \neq 0$, then since deg $f(z) \leq k-1$, this equation can have at most $k-1$ roots in $\mathbb{F}_q$. Therefore a nonzero codeword can have at most $k-1$ symbols equal to zero, so its weight is at least $n - k + 1$. Since the code is linear, this implies that its minimum distance is at least $d \geq n - k + 1$. But by the Singleton bound, $d \leq n - k + 1$; thus $d = n - k + 1$. □

Example 2. Consider the quaternary field $\mathbb{F}_4$ as constructed in Chapter 7, using the irreducible polynomial $g(x) = x^2 + x + 1$ over $\mathbb{F}_2$. Let $\beta_1 = 0, \beta_2 = 1, \beta_3 = x, \beta_4 = x^2 = x + 1$. (This is the same ordering of the symbols as in the tables of Chapter 7.) The RS code with $n = q = 4$ and $k = 2$ is then given by the mapping

$$f(z) = f_0 + f_1 z \longrightarrow (f(\beta_1), f(\beta_2), f(\beta_3), f(\beta_4)).$$

More explicitly, the mapping is

$$(f_0, f_1) \longrightarrow ((f_0 + f_1\beta_1), (f_0 + f_1\beta_2), (f_0 + f_1\beta_3), (f_0 + f_1\beta_4)).$$

Since f_0 and f_1 can each take on arbitrary values from $\mathbb{F}_4$, the code has 16 codewords. The two generators of the code correspond to the two polynomials $g_0(z) = 1$ and $g_1(z) = z$, or equivalently the 4-tuples $(1, 1, 1, 1)$ and $(0, 1, x, x^2)$, respectively. The full coding map is

$0,\ 0$	$\longrightarrow$	$0,\ 0,\ 0,\ 0$	$x,\ 0$	$\longrightarrow$	$x,\ x,\ x,\ x$
$0,\ 1$	$\longrightarrow$	$0,\ 1,\ x,\ x^2$	$x,\ 1$	$\longrightarrow$	$x,\ x^2,\ 0,\ 1$
$0,\ x$	$\longrightarrow$	$0,\ x,\ x^2,\ 1$	$x,\ x$	$\longrightarrow$	$x,\ 0,\ 1,\ x^2$
$0,\ x^2$	$\longrightarrow$	$0,\ x^2,\ 1,\ x$	$x,\ x^2$	$\longrightarrow$	$x,\ 1,\ x^2,\ 0$
$1,\ 0$	$\longrightarrow$	$1,\ 1,\ 1,\ 1$	$x^2,\ 0$	$\longrightarrow$	$x^2,\ x^2,\ x^2,\ x^2$
$1,\ 1$	$\longrightarrow$	$1,\ 0,\ x^2,\ x$	$x^2,\ 1$	$\longrightarrow$	$x^2,\ x,\ 1,\ 0$
$1,\ x$	$\longrightarrow$	$1,\ x^2,\ x,\ 0$	$x^2,\ x$	$\longrightarrow$	$x^2,\ 1,\ 0,\ x$
$1,\ x^2$	$\longrightarrow$	$1,\ x,\ 0,\ x^2$	$x^2,\ x^2$	$\longrightarrow$	$x^2,\ 0,\ x,\ 1$

We see by inspection that the minimum nonzero weight of this $(4, 2)$ linear code is $d = 3$. Moreover, there are $N_3 = 12$ codewords of weight 3 and $N_4 = 3$ codewords of weight 4, in accord with the weight formulas of the previous section. □

The RS codes are naturally nested— *i.e.,* the $(n = q, k, d)$ code is a subcode of the $(n = q, k+1, d-1)$ code— since the polynomials of degree less than k are a subset of the polynomials of degree less than $k + 1$. The $(n = q, n, 1)$ RS code is the universe code consisting of all q^q q-tuples over $\mathbb{F}_q$, and the $(n = q, 1, n)$ code is the repetition code over $\mathbb{F}_q$, consisting of all q codewords of the form $(\beta, \beta, \ldots, \beta)$ for $\beta \in \mathbb{F}_q$.

An $(n = q, k, d = n - k + 1)$ RS code may be punctured in any $s \leq n - k = d - 1$ places to yield an $(n = q - s, k, d = n - k + 1 - s)$ MDS code. The minimum distance can be reduced by at most s by such puncturing, so $d \geq n - k + 1 - s$, and the code still has q^k distinct codewords; but then by the Singleton bound, $d \leq n - s - k - 1$. Thus by puncturing an RS code, we can create an $(n, k, d = n - k + 1)$ MDS code for any $n \leq q$ and $k, 1 \leq k \leq n$.

For historical reasons (see following subsections), an RS code of length $n = q$ is called an "extended RS code." Here is how to construct a "doubly extended RS code."

Exercise 3 (doubly extended RS codes). Consider the following map from $(\mathbb{F}_q)^k$ to $(\mathbb{F}_q)^{q+1}$. Let $(f_0, f_1, \ldots, f_{k-1})$ be any k-tuple over $\mathbb{F}_q$, and define the polynomial $f(z) = f_0 + f_1 z + \cdots + f_{k_1} z^{k-1}$ of degree less than k. Map $(f_0, f_1, \ldots, f_{k-1})$ to the $(q + 1)$-tuple $(\{f(\beta_j), \beta_j \in \mathbb{F}_q\}, f_{k-1})$— *i.e.,* to the RS codeword corresponding to $f(z)$, plus an additional component equal to f_{k-1}. Show that the q^k $(q+1)$-tuples generated by this mapping as the polynomial $f(z)$ ranges over all q^k polynomials over $\mathbb{F}_q$ of degree less than k form a linear $(n = q + 1, k, d = n - k + 1)$ MDS code over $\mathbb{F}_q$. [Hint: $f(z)$ has degree less than $k - 1$ if and only if $f_{k-1} = 0$.]

Define a $(4, 2, 3)$ linear code over $\mathbb{F}_3$. Verify that all nonzero words have weight 3. □

8.3.2 Punctured RS codes as transform codes

The $(n = q, k, d = n - k + 1)$ RS code defined above has generators $\mathbf{g}_j$ corresponding to the polynomials $f_j(z) = z^j, 0 \leq j \leq k-1$, evaluated at each of the elements of $\mathbb{F}_q$.

In this section we will consider the punctured $(n = q-1, k, d = n - k + 1)$ RS code obtained by puncturing the above code in the first symbol position, whose locator is $\beta_0 = 0$. We will see that such a code may be regarded as being generated by a transform similar to a Fourier transform. We will also see that it can be characterized as the set of all polynomial multiples of a certain generator polynomial $g(z)$, and that it has a cyclic property.

A handy general lemma in any field $\mathbb{F}$ is as follows:

Lemma 8.6 *Let $\mathbb{F}$ be a field with multiplicative identity 1, and let ω be any nth root of 1 in $\mathbb{F}$ other than 1; i.e., such that $\omega^n = 1, \omega \neq 1$. Then*

$$\sum_{j=0}^{n-1} \omega^j = 0.$$

Proof. Since $\omega^n = 1 = \omega^0$, we have

$$\omega \sum_{j=0}^{n-1} \omega^j = \sum_{j=1}^{n} \omega^j = \sum_{j=0}^{n-1} \omega^j;$$

i.e., the sum $S = \sum_{j=0}^{n-1} \omega^j$ satisfies $\omega S = S$. If $\omega \neq 1$, then this implies $S = 0$. □

If ω is an nth root of unity, then so is ω^i for any integer i, since $(\omega^i)^n = \omega^{in} = (\omega^n)^i = 1$. Therefore we have the immediate corollary:

Corollary 8.7 *Let $\mathbb{F}$ be a field with multiplicative identity 1, let ω be a primitive nth root of 1 in $\mathbb{F}$ (i.e., $\omega^n = 1$ and $\omega^i \neq 1$ for $1 \leq i < n$), and let i be any integer. Then*

$$\sum_{j=0}^{n-1} \omega^{ij} = \begin{cases} n, & \text{if } i = 0 \bmod n; \\ 0, & \text{otherwise.} \end{cases}$$

Proof. If $i = 0 \bmod n$, then $\omega^i = 1$ and the sum is simply $1 + 1 + \cdots + 1$ (n times), an element of $\mathbb{F}$ which we denote by n. If $i \neq 0 \bmod n$ and ω is a primitive nth root of 1, then ω^i is an nth root of 1 other than 1, so Lemma 8.6 applies and the sum is 0. □

Note that if $\mathbb{F}$ is a finite field $\mathbb{F}_q$ and $n = q - 1$, then $n = -1$ in $\mathbb{F}_q$, since p divides q and thus $q = 0$ in $\mathbb{F}_q$.

Using this corollary, we can then define a "finite Fourier transform" (FFT) and "inverse finite Fourier transform" (IFFT) over $\mathbb{F}$ as follows. Let $\mathbf{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{F}^n$ be any n-tuple of elements of $\mathbb{F}$. Then the FFT of $\mathbf{f}$ is defined as the n-tuple $\mathbf{F} = (F_0, F_1, \ldots, F_{n-1}) \in \mathbb{F}^n$, where

$$F_i = \sum_{j=0}^{n-1} f_j \omega^{ij}.$$

The IFFT of $\mathbf{F}$ is defined as $\mathbf{f} = (f_0, f_1, \ldots, f_{n-1}) \in \mathbb{F}^n$, where

$$f_j = n^{-1} \sum_{i=0}^{n-1} F_i \omega^{-ij}.$$

Here n^{-1} denotes the multiplicative inverse of n in $\mathbb{F}_q$, which we assume exists; *i.e.,* $n \neq 0 \in \mathbb{F}_q$. Using Corollary 8.7, we then verify that the IFFT of the FFT of $\mathbf{f}$ is $\mathbf{f}$:

$$n^{-1} \sum_{i=0}^{n-1} \sum_{j'=0}^{n-1} f_{j'} \omega^{ij'} \omega^{-ij} = n^{-1} \sum_{j'=0}^{n-1} f_{j'} \sum_{i=0}^{n-1} \omega^{i(j'-j)} = \sum_{j'=0}^{n-1} f_{j'} \delta_{jj'} = f_j,$$

since Corollary 8.7 shows that $\sum_{i=0}^{n-1} \omega^{i(j'-j)}$ is equal to n when $j = j' \pmod n$ and 0 otherwise. Similarly the FFT of the IFFT of $\mathbf{F}$ is $\mathbf{F}$. In other words, $\mathbf{f} \leftrightarrow \mathbf{F}$ is a transform pair over $\mathbb{F}_q$ analogous to a finite Fourier transform pair over the complex field $\mathbb{C}$.

Now a codeword $\mathbf{F} = (f(1), f(\alpha), \ldots, f(\alpha^{-1}))$ of a punctured $(n = q-1, k, d = n-k+1)$ RS code over $\mathbb{F}_q$ is obtained from an information k-tuple $(f_0, f_1, \ldots, f_{k-1})$ as follows:

$$\mathbf{F} = \sum_{j=0}^{k-1} f_j \alpha^{ij},$$

where α is a primitive nth root of unity in $\mathbb{F}_q$. Thus $\mathbf{F}$ may be regarded as the FFT of the n-tuple $\mathbf{f} = (f_0, f_1, \ldots, f_{k-1}, 0, \ldots, 0)$ whose last $n-k$ elements are zeroes, where the FFT is defined using a primitive nth root of unity $\alpha \in \mathbb{F}_q$.

Using the IFFT and the fact that $n^{-1} = (-1)^{-1} = -1$, the information vector $\mathbf{f}$ is therefore easily recovered from a codeword $\mathbf{F}$ as follows:

$$f_j = -\sum_{i=0}^{n-1} F_i \alpha^{-ij}.$$

Note that the IFFT must equal 0 for the last $n-k$ symbols $f_j, k \le j \le n-1$:

$$\sum_{i=0}^{n-1} F_i \alpha^{-ij} = 0, k \le j \le n-1.$$

8.3.3 RS codes as sets of polynomials

Let us define a polynomial $F(z) = F_0 + F_1 z + \cdots + F_{n-1} z^{n-1}$ of degree $n-1$ or less corresponding to a codeword $\mathbf{F}$. This last equation then implies that that $F(\alpha^{-j}) = 0$ for $k \le j \le n-1$. In other words, $\alpha^j \in \mathbb{F}_q$ is a root of $F(z)$ for $1 \le j \le n-k$.

It follows that the polynomial $F(z)$ has $n-k$ factors of the form $z - \alpha^j, 1 \le j \le n-k$. In other words, $F(z)$ is divisible by the *generator polynomial*

$$g(z) = \prod_{j=1}^{n-k} \left(z - \alpha^j\right).$$

Evidently $\deg g(z) = n-k$.

In fact, the RS code may now be characterized as the set of all polynomials $F(z)$ of degree less than n that are multiples of the degree-$(n-k)$ generator polynomial $g(z)$. We have already observed that $g(z)$ divides every codeword polynomial $F(z)$. But since $F(z) = g(z)h(z)$ has degree less than n if and only if $h(z)$ has degree less than k, there are precisely q^k distinct multiples $g(z)h(z)$ of $g(z)$ such that $\deg F(z) < n$. This accounts for all codewords. Thus the RS code may be characterized as follows:

Theorem 8.8 (RS codes as sets of polynomials) *The $(n = q-1, k, d = n-k+1)$ punctured RS code over $\mathbb{F}_q$ corresponds to the set of q^k polynomials*

$$\{F(z) = g(z)h(z), \deg h(z) < k\},$$

where $g(z) = \prod_{j=1}^{n-k} \left(z - \alpha^j\right)$.

Example 2 (cont.) For the $(4, 2, 3)$ RS code over $\mathbb{F}_4$ of Example 2, if we puncture the first symbol, then we get a $(3, 2, 2)$ RS code with generators $\mathbf{g}_0 = (1, 1, 1)$ and $\mathbf{g}_1 = (1, \alpha, \alpha^2)$ (where α is a primitive field element). This code corresponds to the set of 16 polynomials $\{F(z) = g(z)h(z), \deg h(z) < 2\}$ of degree less than 3 that are multiples of the degree-1 generator polynomial $g(z) = z + \alpha$, as can be verified by inspection. □

8.3.4 Cyclic property of punctured RS codes

Finally, we show that punctured RS codes are cyclic; *i.e.,* the end-around cyclic rotation $\mathbf{F}' = (F_{n-1}, F_0, F_1, \ldots, F_{n-2})$ of any codeword $\mathbf{F} = (F_0, F_1, \ldots, F_{n-1})$ is a codeword. In polynomial terms, this is equivalent to the statement that $F'(z) = zF(z) - F_{n-1}(z^n - 1)$ is a codeword.

We have seen that the polynomial $z^n - 1$ factors completely as follows: $z^n - 1 = \prod_{j=1}^{n} \left(z - \alpha^j\right)$. Consequently $g(z) = \prod_{j=1}^{n-k} \left(z - \alpha^j\right)$ divides $z^n - 1$. Since $g(z)$ also divides $F(z)$, it follows that $g(z)$ divides $F'(z) = zF(z) - F_{n-1}(z^n - 1)$, so $F'(z)$ is a codeword.

Example 2 (cont.) The punctured $(3, 2, 2)$ RS code described above has the cyclic property, as can be verified by inspection. □

Historically, RS codes were introduced by Reed and Solomon (1960) as valuation codes. In the 1960s and 1970s, RS and BCH codes were primarily studied as cyclic codes. The transform approach was popularized by Blahut in the early 1980s. In the past decade, RS codes have again come to be regarded as valuation codes, due in part to the work of Sudan *et al.* on decoding as interpolation. This will be our next topic.

8.4 Introduction to RS decoding

An important property of RS codes is that in practice they can be decoded rather easily, using finite-field arithmetic. For example, decoding of the NASA-standard $(255, 233, 33)$ 16-error-correcting code over $\mathbb{F}_{256}$ is today routinely accomplished at rates of Gb/s.

A series of decoding algorithms bearing names such as Peterson, Berlekamp-Massey, Euclid, and Welch-Berlekamp have been developed over the years for error-correction and erasure-and-error correction. More recently, Sudan and others have devised list and soft-decision error-correction algorithms that can decode significantly beyond the guaranteed error-correction capability of the code. All of these algorithms are polynomial-time, with the most efficient requiring a number of finite-field operations of the order of n^2.

We will present briefly an error-correction algorithm based on viewing the decoding problem as an interpolation problem. Recent decoding algorithms, including the Sudan-type algorithms, involve extensions of the ideas in this algorithm.

We start by introducing a new notation for n-tuples over $\mathbb{F}_q$ for $n \le q$. As before, we index each of the $n \le q$ coordinates by a unique field element $\beta_j \in \mathbb{F}_q, 1 \le j \le n$. An n-tuple $(y_1, y_2, \ldots, y_n)$ can then be represented by a set of pairs $\{(y_j, \beta_j), 1 \le j \le n\}$. Each pair $(y_j, \beta_j) \in (\mathbb{F}_q)^2$ is regarded as a *point* in the two-dimensional space $(\mathbb{F}_q)^2$. An n-tuple is thus specified by a set of n points in $(\mathbb{F}_q)^2$.

A codeword in an $(n = q, k, d = n - k + 1)$ RS code is then an n-tuple specified by a set of $n = q$ points (y_j, β_j) satisfying the equation $y_j = f(\beta_j)$ for a fixed polynomial $f(z) \in \mathbb{F}_q[z]$ of degree less than k. In other words, the n points of the codeword are the n roots $(y, z) \in (\mathbb{F}_q)^2$ of the algebraic equation $y = f(z)$, or $y - f(z) = 0$.

Curves specified by such algebraic equations are called *algebraic curves.* The n points of the curve are said to *lie on* the bivariate polynomial $p(y, z) = y - f(z)$; or, the bivariate polynomial $p(y, z) = y - f(z)$ is said to *pass through* the point $(\alpha, \beta) \in (\mathbb{F}_q)^2$ if $p(\alpha, \beta) = 0$.

The decoding problem can then be expressed as an interpolation problem: given n received points, find the bivariate polynomial of the form $p(y, z) = y - f(z)$ with deg $f(z) < k$ that passes through as many of the received points as possible.

To make further progress, we express the received n-tuple $\mathbf{y}$ as the sum (over $\mathbb{F}_q$) of a codeword $\mathbf{c}$ and an unknown error n-tuple $\mathbf{e}$. The closest codeword $\mathbf{c}$ is the one for which the corresponding error n-tuple $\mathbf{e} = \mathbf{y} - \mathbf{c}$ has least Hamming weight. If the minimum distance of the code is d and $\mathbf{y}$ is within distance $t < d/2$ of a codeword $\mathbf{c}$, then the distance to any other codeword is at least $d - t > d/2$, so the closest codeword is unique.

An *error-locator polynomial* associated with an error n-tuple $\mathbf{e}$ is any polynomial $\Lambda(z)$ such that $\Lambda(\beta_j) = 0$ whenever $e_j \neq 0$. If the error vector has weight t, then clearly the degree-t polynomial

$$\Lambda_0(z) = \prod_{j : e_j \neq 0} (z - \beta_j)$$

whose t roots are the error locators $\{\beta_j : e_j \neq 0\}$ is an error-locator polynomial. Moreover, every error-locator polynomial $\Lambda(z)$ must be a polynomial multiple of $\Lambda_0(z)$.

If $\Lambda(z)$ is an error-locator polynomial for $\mathbf{e}$ and $f(z)$ is the polynomial that maps to the codeword $\mathbf{c}$, then the bivariate polynomial

$$q(y, z) = \Lambda(z)(y - f(z))$$

evaluates to 0 for every received point (y_j, β_j); *i.e.*, $q(y, z)$ passes through all n received points. The decoding interpolation problem can therefore be expressed as follows: given n received points, find the bivariate polynomial of the form $q(y, z) = \Lambda(z)(y - f(z))$ with deg $f(z) < k$ and deg $\Lambda(z) = t$ as small as possible that passes through all received points.

Define $g(z) = \Lambda(z)f(z)$; then

$$q(y, z) = y\Lambda(z) - g(z).$$

If deg $\Lambda(z) = t$, then deg $g(z) < k + t$. Thus, given t, the coefficients of $\Lambda(z)$ and $g(z)$ are a set of $t + 1 + k + t = k + 2t + 1$ unknowns.

Assuming that the number of errors is not greater than t', and substituting the n received points (y_j, β_j) in the equation $y\Lambda(z) - g(z) = 0$, we obtain a system of n homogeneous linear equations in $k + 2t' + 1$ unknowns. In general, these equations may have no solutions other than the all-zero solution, or may have a linear vector space of solutions. If $k + 2t' + 1 > n$, then there must be a space of solutions of dimension at least $k + 2t' + 1 - n$. In particular, if $2t' + 1 = d = n - k + 1$, then there is at least a one-dimensional space of solutions.

The set of all such solutions may be found by standard techniques for solving systems of linear equations, which in general have complexity of the order of n^3. Because of the special structure of these equations, "fast" techniques with complexity of the order of n^2 may be used. There is active current research on such fast solution techniques.

Any such solution specifies two candidate polynomials $\Lambda(z)$ and $g(z)$. If these are the correct polynomials, then since $g(z) = \Lambda(z)f(z)$, the polynomial $f(z)$ may be found simply by dividing $g(z)$ by $\Lambda(z)$. (Note that the roots of $\Lambda(z)$ need not be found explicitly.) If the candidate polynomials are incorrect, then $g(z)$ may not be divisible by $\Lambda(z)$, in which case this candidate pair may be discarded. If $g(z)$ is divisible by $\Lambda(z)$ and the result $f(z) = g(z)/\Lambda(z)$ is a polynomial of degree less than k such that the codeword corresponding to $f(z)$ is within distance t' of the received n-tuple, then a candidate for the closest codeword has been found.

For standard errors-only error-correction, the distance d is chosen to be odd and t' is chosen such that $2t' + 1 = d$. We then obtain n homogeneous linear equations in $n + 1$ unknowns, ensuring that there exists at least a one-dimensional space of solutions. If the actual number of errors is $t \leq t'$, then there is guaranteed to be one and only one valid candidate solution, so as soon as any valid solution is found, it may be declared to be the closest codeword.

For erasure-and-error-correction, or for decoding of a punctured code, a certain number s of the received symbols may be erased or punctured— *i.e.,* unknown. The above decoding method is easily adapted to this case. We then have a system of $n - s$ homogeneous linear equations in $k + 2t' + 1$ unknowns. If $d - s$ is odd and t' is chosen so that $2t' + s + 1 = d$, then we obtain $n - s$ equations in $k + d - s = n - s + 1$ unknowns, ensuring at least a one-dimensional space of solutions. Moreover, if the actual number of errors is $t \leq t'$, then there is guaranteed to be one and only one valid candidate solution.

Recent work by Sudan *et al.* has explored decoding beyond the guaranteed error-correction capability of the code. The main idea is to replace $\Lambda(z)$ by $\Lambda(y, z)$. Then the polynomial $q(y, z)$ can have more then one factor of type $y - f(z)$, and the list of all such factors is produced at the decoder output. The number of such factors (the list size in Sudan-type decoding) is obviously bounded by $\deg_y q(y, z) = \deg_y \Lambda(y, z) + 1$. It can be shown that under certain conditions, the codeword $\mathbf{c}$ will be on the list, even if the distance between $\mathbf{y}$ and $\mathbf{c}$ exceeds $d/2$. In practice, the probability that the list will contain more than one codeword is usually very small.

Finally, in many cases it may be possible to assign a reliability to each received symbol, where the reliability is an estimate of the log likelihood of the symbol being correct. Or, the symbol receiver may itself put out a list of more than one candidate symbol, each with a different reliability. In such cases the decoder may develop a list of candidate codewords, from which the one with greatest reliability (log likelihood) may then be chosen. Such reliability-based decoding schemes can make up much of the performance difference between hard-decision and maximum-likelihood decoding.

8.5 Applications of RS codes

RS codes are useful for a variety of applications:

(a) For channels that naturally transmit packets or binary m-tuples, RS codes over $\mathbb{F}_{2^m}$ are used for *m-tuple (byte) error correction.*

(b) Over memoryless channels such as the AWGN channel, powerful codes may be constructed by *concatenation* of an "inner code" consisting of $q = 2^m$ codewords or signal points together with an "outer code" over $\mathbb{F}_q$, where $\mathbb{F}_q$ is identified with the inner codewords. Such a concatenated code may be efficiently decoded by maximum-likelihood detection of the inner code followed by algebraic decoding of the RS outer code, preferably using reliability information from the inner decoder in the RS decoding.

(c) RS codes are also useful for *burst error correction*; for this purpose they are often combined with m-tuple (byte) interleavers.

8.5.1 Concatenation using RS Codes

A basic concatenated code involves two codes, called the inner code and outer code, as shown in Figure 1. We see immediately that a concatenation scheme is another example of a layered architecture.

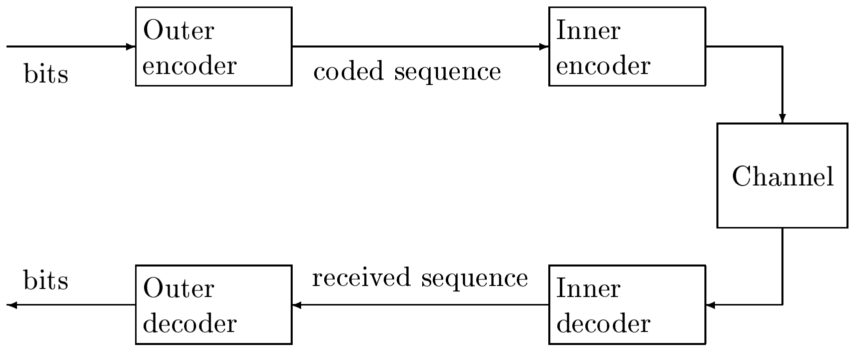

Figure 1. Concatenation of an inner code and an outer code.

The inner code is typically a binary code, either block or convolutional, used over a binary channel, such as a discrete-time AWGN channel with a 2-PAM signal set. The inner decoder is typically a maximum-likelihood decoder, such as a Viterbi decoder, which uses soft (reliability-weighted) outputs from the channel.

The outer code is typically a RS code over a finite field $\mathbb{F}_{2^m}$ of characteristic 2. Each field element may thus be represented by a binary m-tuple. The field $\mathbb{F}_{256}$ is commonly used, in which case field elements are represented by 8-bit bytes. An RS codeword of length n symbols may therefore be converted into an RS-coded sequence of nm bits, which is the input sequence to the inner encoder.

The input sequence to the inner encoder is often permuted by an interleaver to ensure that symbols in a given RS codeword are transmitted and decoded independently. A simple row-column block interleaver works as follows, assuming that the inner code is a binary linear block code of dimension $k = mN$. Each RS codeword, consisting of n m-bit symbols, forms a horizontal row in a rectangular $N \times n$ array of m-bit symbols, as shown below. Each column of the array, consisting of N symbols, is taken as the input symbol sequence of length $k = mN$ bits for one inner code block.

symbol 1 in RS cw 1	symbol 2 in RS cw 1	...	symbol n in RS cw 1
symbol 1 in RS cw 2	symbol 2 in RS cw 2	...	symbol n in RS cw 2
...	...		...
symbol 1 in RS cw N	symbol 2 in RS cw N	...	symbol n in RS cw N

The input sequence is read out column by column, with the column length N chosen large enough so that symbols in the same row are effectively independent. Such an interleaver also protects against error bursts of length mN bits or less.

The output of the inner decoder is correspondingly viewed as a sequence of m-bit symbols. Typically the inner decoder puts out a sequence of hard decisions on bits, which are converted to hard decisions on symbols. As we have shown, the inner decoder output may preferably also include reliability information indicating how confident it is in its hard decisions. However, for simplicity we will omit this refinement. The performance of the inner code may then be characterized by the probability $P_s(E)$ that an m-bit output symbol is in error.

The outer RS decoder takes this sequence of symbol hard decisions and performs error-correction, using an algebraic decoding algorithm. If the RS code has minimum distance d, then the decoder can decode any pattern of t errors correctly provided that $2t < d$; *i.e.,* provided that $t \leq t_{\max} = \lfloor (d-1)/2 \rfloor$. Most RS error-correctors give up if there is no codeword within Hamming distance $t_{\max}$ of the received sequence. With such a bounded-distance decoder, therefore, the probability per codeword of not decoding correctly is

$$\Pr(\text{ndc}) = \Pr\{t_{\max} + 1 \text{ or more symbol errors}\}.$$

It is difficult to obtain good analytical estimates of the probability of not decoding correctly when $p = P_s(E)$ is relatively large, which is the operating region for concatenated codes (but see Exercise 4 below). Empirically, one observes that $\Pr(\text{ndc}) \approx 1$ when $p \approx t_{\max}/n$. As p decreases, there is a threshold region, typically $p \approx 10^{-2}$–10^{-3}, where $\Pr(\text{ndc})$ suddenly decreases to very low values, *e.g.,* $\Pr(\text{ndc}) \approx 10^{-12}$.

The objective of the inner code is therefore to achieve a very moderate symbol probability of error such as $P_s(E) \approx 10^{-2}$–10^{-3} at as low an E_b/N_0 as possible (on an AWGN channel). For this purpose maximum-likelihood decoding— specifically, the VA— is a good choice. The code should have a trellis with as many states as the VA can reasonably decode, and should be the best code with that number of states, which almost always implies a convolutional code. A 64-state convolutional code was standardized in the 1970's, and around 1990 NASA built a 2^{14}-state Viterbi decoder.

The objective of the outer code is then to drive $\Pr(\text{ndc})$ down to the target error rate with as little redundancy as possible. RS codes and decoders are ideal for this purpose. A $(255, 223, 33)$ RS code over $\mathbb{F}_{256}$ was standardized in the 1970's, and is still in common use, nowadays at rates up to Gb/s. Concatenated codes like this can attain low error rates within about 2–3 dB of the Shannon limit on a power-limited AWGN channel.

Exercise 4. Show that if symbol errors occur independently with probability $p = P_s(E)$, then the probability of the RS decoder not decoding correctly is

$$\Pr(\text{ndc}) = \sum_{t=t_{\max}+1}^{n} \binom{n}{t} p^t (1-p)^{n-t}.$$

Using the Chernoff bound, prove the exponential upper bound

$$\Pr(\text{ndc}) \le e^{-nD(\tau||p)},$$

where $\tau = (t_{\max} + 1)/n$ and $D(\tau||p)$ is the divergence (relative entropy)

$$D(\tau||p) = \tau \ln \frac{\tau}{p} + (1-\tau) \ln \frac{1-\tau}{1-p}$$

between two Bernoulli distributions with parameters τ and p, respectively. □

8.6 Binary BCH codes

In a finite field $\mathbb{F}_{2^m}$ of characteristic 2, there always exists a prime subfield $\mathbb{F}_2$ consisting of the two elements $\mathbb{F}_2 = \{0, 1\}$. Correspondingly, in an $(n, k, d = n - k + 1)$ RS code over $\mathbb{F}_{2^m}$, there always exists a subset of codewords whose components are entirely in the subfield $\mathbb{F}_2$. This "subfield subcode" is called a binary Bose-Chaudhuri-Hocquenghem (BCH) code. (Historically, binary BCH codes were developed independently of RS codes, but at about the same time, namely 1959-60.)

Example 2 (cont.). The binary BCH code derived from the $(4, 2, 3)$ RS code over $\mathbb{F}_4$ of Example 2 is the binary code consisting of the two codewords 0000 and 1111; *i.e.,* the $(4, 1, 4)$ repetition code over $\mathbb{F}_2$.

A binary BCH code is obviously linear, since the sum of any two binary RS codewords is another binary RS codeword. Its parameters (n', k', d') are related to the parameters (n, k, d) of the RS code from which it is derived as follows. Its length n' is evidently the same as the length n of the RS code. Its minimum Hamming distance d' must be at least as great as the minimum distance d of the RS code, because any two words must differ in at least d places. Usually $d' = d$. Finally, by the Singleton bound, $k' \le n' - d' + 1 \le n - d + 1 = k$. Usually k' is considerably less than k, and the binary BCH code falls considerably short of being MDS.

Determining k' for a binary BCH code derived from a given (n, k, d) RS code over F_{2^m} is an exercise in cyclotomics. Let us associate a polynomial $F(z) \in \mathbb{F}_{2^m}[z]$ with each codeword $\mathbf{F} = (F_0, F_1, \ldots, F_{n-1})$ of the RS code as in Section 8.3.3. We showed in Lemma 7.19 that $F(z)$ is actually a binary polynomial with all coefficients in the binary subfield $\{0, 1\}$ if and only if the roots of $F(z)$ are cyclotomic cosets; *i.e.,* if $\beta \in \mathbb{F}_{2^m}$ is a root of $F(z)$, then so are $\beta^2, \beta^4, \ldots$.

It was further shown in Section 8.3.3 that the punctured $(n = q - 1, k, d = n - k + 1)$ RS code over $\mathbb{F}_q$ may be characterized as the set of q^k polynomials $F(z)$ of degree less than n that are multiples of the generator polynomial $g(z) = \prod_{j=1}^{n-k}(z - \alpha^j)$, whose roots are the first $n - k$ powers $\{\alpha, \alpha^2, \ldots, \alpha^{n-k}\}$ of a primitive element $\alpha \in \mathbb{F}_q$. Since $n - k = d - 1$, the subset of binary polynomials in this set can thus be characterized as follows:

Theorem 8.9 (BCH codes) *Given a field $\mathbb{F}_q$ of characteristic 2, the $(n = q-1, k', d)$ BCH code over $\mathbb{F}_2$ corresponds to the set of $2^{k'}$ binary polynomials*

$$\{F(z) = g(z)h(z), \deg h(z) < k'\},$$

where $g(z)$ is the product of the distinct polynomials in the set of cyclotomic polynomials of the elements $\{\alpha, \alpha^2, \ldots, \alpha^{d-1}\}$ of $\mathbb{F}_q$, and $k' = n - \deg g(z)$.

Example 3. Let us find the parameters (k', d) of the BCH codes of length $n = 15$. The cyclotomic polynomials of the elements of $\mathbb{F}_{16}$ are the binary irreducible polynomials whose degrees divide 4, namely (taking α to be a root of $x^4 + x + 1$):

polynomial	roots
x	0
$x+1$	1
x^2+x+1	α^5, α^{10}
x^4+x+1	$\alpha, \alpha^2, \alpha^4, \alpha^8$
$x^4+x^3+x^2+x+1$	$\alpha^3, \alpha^6, \alpha^{12}, \alpha^9$
x^4+x^3+1	$\alpha^7, \alpha^{14}, \alpha^{13}, \alpha^{11}$

For the first BCH code, we take $g(x) = x^4 + x + 1$, which has α and α^2 as roots; therefore this code is the subfield subcode of the $(15, 13, 3)$ RS code over $\mathbb{F}_{16}$. It has $d = 3$ and $k' = n - \deg g(x) = 15 - 4 = 11$; *i.e.,* it is a $(15, 11, 3)$ code.

For the second BCH code, we take $g(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)$, which has $\{\alpha, \alpha^2, \alpha^3, \alpha^4\}$ as roots; therefore this code is the subfield subcode of the $(15, 11, 5)$ RS code over $\mathbb{F}_{16}$. It has $d = 5$ and $k' = n - \deg g(x) = 7$; *i.e.,* it is a $(15, 7, 5)$ code.

For the third BCH code, we take $g(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1)(x^2 + x + 1)$, which has $\{\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}$ as roots; therefore this code is the subfield subcode of the $(15, 9, 7)$ RS code over $\mathbb{F}_{16}$. It has $d = 7$ and $k' = n - \deg g(x) = 5$; *i.e.,* $(n, k, d) = (15, 5, 7)$.

Finally, we find a $(15, 1, 15)$ BCH code with a degree-14 generator polynomial $g(x)$ which has $\{\alpha, \alpha^2, \ldots, \alpha^{14}\}$ as roots; *i.e.,* the binary repetition code of length 15. □

Table I below gives the parameters $(n = q, k', d+1)$ for all binary BCH codes of lengths $n = 2^m$ with even minimum distances $d+1$. By puncturing one symbol, one may derive $(n = q-1, k', d)$ binary BCH codes with odd minimum distances as above; these are the binary BCH codes that are more usually tabulated. The former codes may be obtained from the latter by adding an overall parity check as in Chapter 6, Exercise 1.

$n = q$	k'	$d+1$
2	1	2
4	3	2
4	1	4
8	7	2
8	4	4
8	1	8
16	15	2
16	11	4
16	7	6
16	5	8
16	1	16
32	31	2
32	26	4
32	21	6
32	16	8
32	11	12
32	6	16
32	1	32
64	63	2
64	57	4
64	51	6
64	45	8
64	39	10
64	36	12
64	30	14
64	24	16
64	18	22
64	16	24
64	10	28
64	7	32
64	1	64

Table I. Binary BCH codes of lengths $n = 2^m$ for $m \leq 6$.

We see that binary BCH codes with $n = q$ include codes equivalent to SPC codes, extended Hamming codes, biorthogonal codes, and repetition codes.

In comparison to binary RM codes, there exist more binary BCH codes; in particular, codes with distances other than 2^{m-r}. Also, for $n \geq 64$, we begin to see codes with slightly better k for a given n and d; *e.g.,* the $(64, 45, 8)$ binary BCH code has three more information bits than the corresponding RM code, and the $(64, 24, 16)$ code has two more information bits.

In principle, a binary BCH code may be decoded by any decoding algorithm for the RS code from which it is derived. Candidate decoded RS codewords that are not binary may be discarded. In practice, certain simplifications are possible in the binary case.

For these reasons, particularly the availability of algebraic decoding algorithms, binary BCH codes have historically received more attention than binary RM codes.

However, for trellis-based maximum-likelihood (Viterbi) decoding, which we will discuss shortly, RM codes usually have a better performance-complexity tradeoff. For example, the $(64, 45, 8)$ binary BCH code has eight times the trellis complexity of the $(64, 42, 8)$ RM code, and the $(64, 24, 16)$ binary BCH code has four times the trellis complexity of the $(64, 22, 16)$ RM code. The corresponding increase in complexity of trellis-based decoding algorithms will usually not justify the slight increase in number of information bits.