Applications of Coding Theory to Security

Okko Makkonen
May 17, 2024

These lecture notes are written for the course *Applications of Coding Theory to Security* at Aalto University and are based on the textbook [LX04].

1 Channel coding

One of the first uses of coding theory and error-correcting codes was in the transmission of information over an unreliable channel. In this section we give a motivation for some of the concepts in coding theory that will be used throughout the course. The main reference for this section is [LX04, Chapter 2].

1.1 Communication channels

We can think of a communication channel as a randomized transformation of an input $x \in \mathcal{X}$ to an output $y \in \mathcal{Y}$. The channel is described by the input and output *alphabets* $\mathcal{X}$ and $\mathcal{Y}$ as well as the *transition probabilities* $\mathbb{P}[y \mid x]$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ (or the probability density function $p(y \mid x)$). The idea is that the transmitted symbol x is received with some random error as the symbol y.

Communication channels can be for example an antenna transmitting electromagnetic waves to send a message to a receiver, or a physical medium such as a transistor storing information that can be accessed at a later point in time.

Example 1.1. One of the simplest channels is the *binary symmetric channel (BSC)* which has alphabets $\mathcal{X} = \mathcal{Y} = \{0, 1\}$ and transition probabilities

$$\mathbb{P}[y = 0 \mid x = 0] = \mathbb{P}[y = 1 \mid x = 1] = 1 - p$$
$$\mathbb{P}[y = 0 \mid x = 1] = \mathbb{P}[y = 1 \mid x = 0] = p.$$

Here, $p \in [0, 1]$ is known as the *crossover probability*. Sending one bit (= element of $\{0, 1\}$) through this channel results in a probability p of receiving the bit in error. Even in a good channel, the error probability may be too large as we want the error probability to be as close to zero as possible.

The key solution to this problem is to use the channel many times to send just one bit. Let's use the channel 3 times by repeating the desired bit value, *i.e.*, instead of 0 send 000 and instead of 1 send 111. The receiver should do majority voting on the output (choose the bit that appears the most). If they receive 010 decode to 0 and if they receive 110 decode to 1. This method produces the correct result if the channel produces 1 or fewer errors. Therefore, assuming that the errors are independent on the 3 different uses of the channel (the channel is said to be *memoryless*), then the new error probability is

$$P_e = \underbrace{\binom{3}{3}p^3}_{\text{probability of 3 errors}} + \underbrace{\binom{3}{2}p^2(1-p)}_{\text{probability of 2 errors}} = 3p^2 - 2p^3,$$

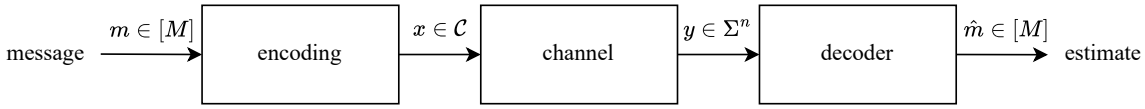

Figure 1: The flow of a message going through a communication channel with encoding and decoding

which is usually much smaller than p. In general, we may repeat the message $2r+1$ times such that as long as $\leqslant r$ errors are introduced, it is still possible to decode the correct answer. Such a coding scheme is known as a repetition code. We are able to send 1 bit by using the channel $n = 2r+1$ times, which means that the communication rate is $1/n$.

By using the channel many times, we have created a new channel on the alphabets $\mathcal{X}^n$ and $\mathcal{Y}^n$ with some new transition probabilities. If n is fixed, then we want to send messages from a proper subset $\mathcal{C} \subset \mathcal{X}^n$. The subset $\mathcal{C}$ is known as a block code of length n.

1.2 Block codes

Let Σ be a finite set of size $q = |\Sigma|$, known as the alphabet. We denote $\Sigma^n = \Sigma \times \cdots \times \Sigma$ to be the set of n-tuples (also known as *words*) with entries in Σ. A nonempty subset $\mathcal{C} \subset \Sigma^n$ is said to be a *block code*[1] of *length* n over the alphabet Σ. The set is also known as the *codebook* and the elements as *codewords*. The *size* of a code $\mathcal{C}$ is $M = |\mathcal{C}|$ and the *rate* of a code is $\mathcal{R} = \log_q(M)/n$. The rate describes the relative size of the code in the ambient space Σ^n. With these notations we say that $\mathcal{C}$ is an $(n, M)_q$-code.

Example 1.2. Above we considered the block code $\mathcal{C} = \{000, 111\} \subseteq \{0,1\}^3$ of length $n = 3$ and size $M = 2$ over an alphabet of size $q = 2$. We say that $\mathcal{C}$ is an $(3,2)_2$-code. The rate of this code is $\mathcal{R} = \log_2(2)/3 = 1/3$, which matches the communication rate discussed above.

When we send information using a block code, we encode each of the M possible messages as one of the codewords and send that through the product channel by using the channel n times. The flow of this process can be seen in Figure 1.

Given the transition probabilities of the communication channel, we may compute the probabilities of the word y being received given that a specific word x was sent as $\mathbb{P}[y \mid x]$. As we only send codewords in $\mathcal{C}$, we are considering the likelihoods $\mathbb{P}[y \mid x]$ for all $x \in \mathcal{C}$. In *maximum likelihood (ML) decoding*, we choose the codeword $x \in \mathcal{C}$ that maximizes this likelihood for the given output y, *i.e.*, the decoder outputs[2]

$$\widehat{x} = \arg\max_{x \in \mathcal{C}} \mathbb{P}[y \mid x].$$

Example 1.3. Consider the (n-fold) binary symmetric channel with crossover probability $p < \frac{1}{2}$. Let $x \in \mathcal{C}$ be a codeword and $y \in \{0,1\}^n$ the received word. Let

$$d = |\{i \in [n] \colon x_i \neq y_i\}|$$

[1]The word block refers to the fact that the codewords all have the same length n. There are also codes of variable length and codes where the different symbols are in different alphabets.

[2]There has to be some rule for breaking a tie in case there are many codewords that maximize the likelihood function, but this is not important for now.

to be the number of coordinates where x and y differ. Then,

$$\mathbb{P}[y \mid x] = p^d(1-p)^{n-d} = \underbrace{(1-p)^n}_{\text{independent of } x \text{ and } y} \left(\frac{p}{1-p}\right)^d$$

as the errors are independent in each of the coordinates. Denote $d = d(x, y)$. Then

$$\widehat{x} = \arg\max_{x \in \mathcal{C}} \mathbb{P}[y \mid x] = \arg\max_{x \in \mathcal{C}} \left(\frac{p}{1-p}\right)^{d(x,y)} = \arg\min_{x \in \mathcal{C}} d(x, y).$$

The above is due to the fact that $\frac{p}{1-p} < 1$, so the quantity is maximized with the smallest exponent.

According to the example above, ML decoding on the binary symmetric channel is equivalent to minimizing the distance given by $d(x, y)$. That is, given the output y, find the codeword $x \in \mathcal{C}$ that is closest to y in the distance given by $d(x, y)$.

1.3 Hamming distance

For $x, y \in \Sigma^n$, the *Hamming distance* between x and y is given by

$$d(x, y) = |\{i \in [n] \colon x_i \neq y_i\}|.$$

The Hamming distance has the following properties:

- $0 \leqslant d(x, y) \leqslant n$ for all $x, y \in \Sigma^n$,
- $d(x, y) = 0$ if and only if $x = y$,
- $d(x, y) = d(y, x)$ for all $x, y \in \Sigma^n$, and
- $d(x, z) \leqslant d(x, y) + d(y, z)$ for all $x, y, z \in \Sigma^n$ (triangle inequality).

We can also write $d(x, y) = d(x_1, y_1) + \cdots + d(x_n, y_n)$, where

$$d(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i \\ 0 & \text{if } x_i = y_i \end{cases}.$$

To prove the triangle inequality it is enough to show it for just $d(x_i, z_i)$. If $d(x_i, y_i) = 0$, then the inequality is clear. On the other hand, if $d(x_i, z_i) = 1$, then $x_i \neq z_i$, so either $x_i \neq y_i$ or $y_i \neq z_i$. Hence, $d(x_i, y_i) + d(y_i, z_i) \geqslant 1 = d(x_i, z_i)$.

In general, we may use *minimum distance decoding* by using the rule

$$\widehat{x} = \arg\min_{x \in \mathcal{C}} d(x, y).$$

This may or may not be equivalent to ML decoding depending on the channel.

It is usually the case that the output y of a channel is close to the input x in the Hamming distance. Therefore, the minimum distance decoding rule makes an error whenever the output y of the channel is closer to another codeword $x' \neq x \in \mathcal{C}$ than to the originally sent codeword x.

Therefore, we want the codewords in $\mathcal{C}$ to be far away from each other so that the error probability of the minimum distance decoding algorithm is low[3].

The *minimum distance* of a block code $\mathcal{C} \subset \Sigma^n$ is defined as

$$d(\mathcal{C}) = \min\{d(x, y) \colon x, y \in \mathcal{C}, x \neq y\}.$$

Further, if $\mathcal{C}$ contains just one element, then we set $d(\mathcal{C}) = n + 1$ (sometimes also $d(\mathcal{C}) = \infty$). If $\mathcal{C} \subset \Sigma^n$ has M elements and minimum distance d, then we say that $\mathcal{C}$ is an (n, M, d)-code.

Lemma 1.4. *If $y \in \Sigma^n$ and $2v < d(\mathcal{C})$, then there is at most one $x \in \mathcal{C}$ such that $d(x, y) \leqslant v$.*

Proof. Assume that there are $x, x' \in \mathcal{C}$ such that $d(x, y) \leqslant v$ and $d(x', y) \leqslant v$. Then, $d(x, x') \leqslant d(x, v) + d(v, x') \leqslant 2v < d(\mathcal{C})$, so $x = x'$. □

If the codeword $x \in \mathcal{C}$ is sent and the word $y \in \Sigma^n$ is received such that $d(x, y) \leqslant v$ with $2v < d(\mathcal{C})$, then $x \in \mathcal{C}$ is the unique codeword closest to y. Thus, the minimum distance decoding algorithm will make the correct decision. In particular, we say that $\mathcal{C}$ is *v-error-correcting*. The value $t = \lfloor (d(\mathcal{C}) - 1)/2 \rfloor$ is said to be the *unique decoding radius*.

A sequence of codes $\mathcal{C}^{(n)} \subset \Sigma^n$ of rate R have q^{Rn} codewords, which grows exponentially in n. Even to describe such a set is impractical without any additional structure. It is typical in coding theory to only consider codes with a vector space structure over the alphabet, so that the code can be described in terms of a basis. To do this we will need to understand finite fields and vector spaces over these fields.

[3]The minimum distance is not the only factor that determines the error-correction performance of a code.

2 Finite fields

We want the alphabet Σ we use to have the structure of a field such as $\mathbb{R}$ or $\mathbb{C}$ to be able to describe codes as vector spaces over the alphabet. Furthermore, in many communication channels the alphabets are finite, so we want a finite set with the structure of a field. The main reference for this section is [LX04, Chapter 3]. See also the lecture notes on finite fields in [Kop23].

2.1 Field theory

A *field* is a commutative ring F with $F^* = F \setminus \{0\}$. This means that the usual rules of arithmetic in fields such as $\mathbb{Q}, \mathbb{R}$ or $\mathbb{C}$ apply. A *finite field* is a field with finitely many elements.

Example 2.1. Consider the ring of integers modulo m, *i.e.*, $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$. If $m = pq$ is a composite number with $p, q > 1$, then $p, q \not\equiv 0 \pmod{m}$, but $pq = m \equiv 0 \pmod{m}$. Therefore, $\mathbb{Z}_m$ is not a field, since it has zero divisors. On the other hand, if p is a prime and $0 < r < p$, then there are integers $a, b \in \mathbb{Z}$ such that

$$ap + br = 1$$

by Bezout's identity, so $br \equiv 1 \pmod{p}$. Thus, r has an inverse element $b = r^{-1}$ in $\mathbb{Z}_p$ if $r \not\equiv 0 \pmod{p}$. The ring $\mathbb{Z}_m$ is a field if and only if m is prime.

The *characteristic* of a field is the smallest positive integer c such that

$$\underbrace{1_F + 1_F + \cdots + 1_F}_{c \text{ terms}} = c \cdot 1_F = 0_F$$

if it exists, otherwise the characteristic is zero. We write, $\mathrm{char}(F) = c$ or $\mathrm{char}(F) = 0$. Recall that the characteristic of a field is either 0 or a prime p. If $\mathrm{char}(F) = 0$, then F has infinitely many elements (it contains a copy of $\mathbb{Q}$), so a finite field has characteristic $p > 0$. The characteristic of the field $\mathbb{Z}_p$ is p.

Let F be a field with characteristic $\mathrm{char}(F) = p$. Then, it is easy to verify that

$$\mathbb{Z}_p \to F, \quad [n] \mapsto n \cdot 1_F$$

is an injective homomorphism. Therefore, F contains a copy of $\mathbb{Z}_p$ as a subfield. If F is a finite field, then F is a vector space of finite dimension n over $\mathbb{Z}_p$. Thus, $|F| = p^n$ for some n and $p = \mathrm{char}(F)$. We can deduce that the finite field of size p is unique up to isomorphism and it is denoted by $\mathbb{F}_p$ or $\mathrm{GF}(p)$ (GF stands for Galois field). The fields $\mathbb{F}_p$ for all primes p, as well as $\mathbb{Q}$, are said to be *prime fields* and every field contains a unique copy of a prime field that is determined by the characteristic.

Example 2.2. The smallest field is the field with 2 elements: $\mathbb{F}_2 = \mathbb{Z}_2 = \{0, 1\}$. This field is known as the *binary field* and it is popular due to its simple structure and ease of computer implementation. The field with three elements is known as the *ternary field*: $\mathbb{F}_3 = \mathbb{Z}_3 = \{0, 1, 2\}$.

There is a finite field of size $4 = 2^2$, but it is not isomorphic $\mathbb{Z}_4$, which is not a field as 4 is composite.

There is no finite field of size $6 = 2 \cdot 3$, since 6 is not a prime power.

Example 2.3. Every finite integral domain F is a field. Consider $x \neq 0 \in F$. Then, $xy = xy'$ if and only if $y = y'$ due to the fact that F is an integral domain ($x(y - y') = 0$). Thus, $\{xy \mid y \in F \setminus \{0\}\}$ contains $|F| - 1$ nonzero elements, so one of them must be 1_F, *i.e.*, every nonzero element has an inverse.

2.2 Polynomial rings

Recall that $F[x]$ denotes the ring of univariate polynomials in x with coefficients in F. A polynomial is said to be *monic* if its leading coefficient is 1. A polynomial f (also denoted with $f(x)$) is *irreducible* if it cannot be expressed as $f = gh$ for some $g, h \in F[x]$ with $\deg(g), \deg(h) < \deg(f)$. If $f_1, \ldots, f_t \in F[x]$ are polynomials, then $I = (f_1, \ldots, f_t) \subseteq F[x]$ is the *ideal* generated by these polynomials.

Let $f, g \in F[x]$ and $g \neq 0$. By the division algorithm there exists unique polynomials $s, r \in F[x]$ such that

$$f = sg + r, \quad \text{and} \quad \deg(r) < \deg(g) \text{ or } r = 0.$$

Example 2.4. Let $g = x - \alpha \in F[x]$ for some $\alpha \in F$. Then

$$f = sg + r$$

where $\deg(r) < \deg(g) = 1$, *i.e.*, r is a constant. By evaluating at $x = \alpha$ we get that $r = f(\alpha)$ as $g(\alpha) = 0$. If α is a root of f, then $f = s \cdot (x - \alpha)$. Therefore, a polynomial of degree d over a field has at most d roots.

Due to the division algorithm, all ideals in $F[x]$ are principal (every element in the ideal is a multiple of the nonzero element of lowest degree).

Lemma 2.5. *Let $f \in F[x]$ be an irreducible polynomial. Then, $F[x]/(f)$ is a field of extension degree $\deg(f)$ over F.*

Proof. It is clear that $F[x]/(f)$ is a commutative ring, so we need to first show that it is a field. Assume without loss of generality that f is monic. Let $g \in F[x]$ be a polynomial and consider the ideal $I = (f, g)$ that is generated by f and g. As all ideals are principal, we find that $I = (h$ for some monic $h \in F[x]$. Thus, f is a multiple of h. By the irreducibility of f, we have that either $h = f$ or $h = 1$. In the first case g is a multiple of $h = f$, so $g \equiv 0 \pmod{f}$. In the second case, $1 \in I$, so there are polynomials $p, q \in F[x]$ such that $1 = pf + qg$. Therefore, $qg \equiv 1 \pmod{f}$. Hence, all nonzero elements in $F[x]/(f)$ have an inverse.

Let $n = \deg(f)$. Then, the elements $1, x, \ldots, x^{n-1}$ span $F[x]/(f)$, since any polynomial $g \in F[x]$ is equivalent to a polynomial of degree $< n$ modulo f. Furthermore,

$$p = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1} \equiv 0 \pmod{f} \implies p = 0$$

as the only multiple of f that has degree $< n$ is the zero polynomial. Hence, $1, x, \ldots, x^{n-1}$ are linearly independent. Therefore, $F[x]/(f)$ has extension degree n over F. □

With the above lemma we may construct finite fields by starting with irreducible polynomials over $\mathbb{F}_p$. If $f \in \mathbb{F}_p[x]$ is irreducible of degree n, then $\mathbb{F}_p[x]/(f)$ is a finite field of size p^n. In fact, this is essentially the same way that we constructed the field $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$, since both $p\mathbb{Z}$ and (f) are maximal ideals in the rings $\mathbb{Z}$ and $F[x]$, respectively.

Example 2.6. The polynomial $f = x^2 + x + 1 \in \mathbb{F}_2[x]$ is irreducible, since it has no roots in $\mathbb{F}_2$. Therefore, $\mathbb{F}_2[x]/(f)$ is a finite field of size $2^2 = 4$ and it consists of the elements $0, 1, x, x+1$. The addition in this field is carried out by regular polynomial addition modulo 2, while multiplication is carried out modulo the polynomial f.

Example 2.7. The polynomial $x^2 + 1 \in \mathbb{R}[x]$ is irreducible, but $x^2 + 1 \in \mathbb{F}_2[x]$ is not. You have to be careful when showing if a polynomial is irreducible over a specific field.

2.3 Structure of finite fields

Let F be a finite field with $q = p^n$ elements. Recall that F^* is a group with $q - 1$ elements. Thus, by Lagrange's theorem, $\beta^{q-1} = 1$ for all $\beta \in F^*$. Thus, $\beta^q = \beta$ for all $\beta \in F$.

Lemma 2.8. *Let $F \subset E$ be fields with $|F| = q$. Then,*

$$F = \{\beta \in E \mid \beta^q = \beta\}.$$

Proof. Clearly, F is contained in the set on the right hand side. Furthermore, the polynomial $x^q - x$ can have at most q roots, which gives the equality. □

Lemma 2.9. *Let F be a finite field with $\mathrm{char}(F) = p$. Then, $\sigma\colon F \to F$, $\sigma(x) = x^p$ is an automorphism of F.*

Proof. It is clear that $\sigma(xy) = (xy)^p = x^p y^p = \sigma(x)\sigma(y)$. Furthermore,

$$\sigma(x + y) = (x + y)^p = \sum_{i=0}^{p} \binom{p}{i} x^i y^{p-i} = x^p + y^p = \sigma(x) + \sigma(y,)$$

since $\binom{p}{i} \equiv 0 \pmod{p}$ for $0 < i < p$.

As noted above, $\sigma(x) = x$ for all $x \in \mathbb{F}_p \subseteq F$. Therefore, σ is a linear map over the subfield $\mathbb{F}_p$. Hence, σ is bijective if and only if $\ker(\sigma) = \{0\}$. We see that $\sigma(x) = x^p = 0$ implies that $x = 0$, so σ is an isomorphism. □

The automorphism σ defined above is called the *Frobenius automorphism.*

Theorem 2.10. *There is a unique finite field of size p^n up to isomorphism.*

Proof sketch. The finite field of size $q = p^n$ is the splitting field of the polynomial $x^q - x$, so is unique up to isomorphism. □

As finite fields are unique up to isomorphism, we denote the finite field of size $q = p^n$ for prime p and $n \geqslant 1$ by $\mathbb{F}_q$ (or $\mathrm{GF}(q)$).

Recall that the *order* of a nonzero element $g \in \mathbb{F}_q^*$ is the smallest positive integer m such that $g^m = 1$, denoted by $\mathrm{ord}(g) = m$. The order of an element equals the size of the (multiplicative) group generated by that element.

Theorem 2.11. *The group of units $\mathbb{F}_q^*$ is a cyclic group.*

Proof. Let $G = \mathbb{F}_q^*$ and $n = |G|$. For each divisor d of n, define

$$G_d = \{g \in G \mid \operatorname{ord}(g) = d\} \subseteq \{x \in G \mid x^d - 1 = 0\}.$$

Recall that the order of an element divides the order of the group by Lagrange's theorem.

If $G_d \neq \varnothing$, let $g \in G_d$. Then,

$$\langle g \rangle = \{x \in G \mid x^d - 1 = 0\}.$$

since $|\langle g \rangle| = d$ and $|\{x \in G \mid x^d - 1 = 0\}| \leqslant d$ as the polynomial $x^d - 1$ can have at most d roots. Therefore, G_d is the set of generators of $\langle g \rangle$ and has size $\varphi(d)$, where φ denotes Euler's totient function.

Finally,

$$n = |G| = \sum_{d|n} |G_d| \leqslant \sum_{d|n} \varphi(d) = n.$$

To achieve equality, we must have that $|G_d| = \varphi(d)$ for all $d \mid n$. In particular, $|G_n| = \varphi(n) > 0$. Thus, G contains an element of order n. □

If $\gamma \in \mathbb{F}_q$ has order $q - 1$, then $F = \{0, 1, \gamma, \gamma^2, \ldots, \gamma^{q-2}\}$. Such a γ is said to be a *primitive element*. As $\mathbb{F}_q$ is cyclic, primitive elements must exist, but are not in general unique.

3 Linear codes

In coding theory we are interested in vector spaces, since a vector space can be described just with the basis, making it much more efficient compared to an arbitrary set with no structure[4]. The main reference for this section is [LX04, Chapter 4].

We write vectors in $\mathbb{F}_q^k$ as row vectors. If G is a $k \times n$ matrix in $\mathbb{F}_q^{k\times n}$ and $x \in \mathbb{F}_q^k$ is a vector, then we write vector-matrix multiplication as $xG \in \mathbb{F}_q^n$. This is a common notation in coding theory as we can see $k \times n$ matrices as linear maps from k dimensions to n dimensions. The ith coordinate of the vector $x \in \mathbb{F}_q^k$ is denoted by x_i for $i \in [k] = \{1, \dots, k\}$.

Example 3.1. Over small finite fields we sometimes write vectors by concatenating the coordinates. In the binary field for example, $000 = (0, 0, 0) \in \mathbb{F}_2^3$ or $111011 = (1, 1, 1, 0, 1, 1) \in \mathbb{F}_2^6$.

3.1 Vector spaces

Let $\mathbb{F}_q$ be a finite field with q elements and let $V \subseteq \mathbb{F}_q^n$ be a k-dimensional vector space over $\mathbb{F}_q$ (here $0 \leqslant k \leqslant n$). This means that there exists a basis $\{v_1, \dots, v_k\}$ of linearly independent vectors that span V. Any vector $x \in V$ can be described uniquely by the coordinates $\alpha_1, \dots, \alpha_k \in \mathbb{F}_q$ as

$$x = \alpha_1 v_1 + \cdots + \alpha_k v_k.$$

Therefore, V contains $|V| = q^k$ vectors corresponding to all elements in $\mathbb{F}_q^k$.

In addition to a basis, we can describe a vector space as the kernel of a linear map, *i.e.*, the set of elements satisfying some linear constraints. Let $\varphi\colon \mathbb{F}_q^n \to \mathbb{F}_q^m$ be a linear map between vector spaces. Then, $V = \ker(\varphi) \subseteq \mathbb{F}_q^n$ is a vector space. In fact, every subspace of $\mathbb{F}_q^n$ can be described in this way. Let $V \subseteq \mathbb{F}_q^n$ be a subspace and V' a complementary subspace such that $V \oplus V' = \mathbb{F}_q^n$. If $\varphi\colon \mathbb{F}_q^n \to \mathbb{F}_q^n$ is the projection onto V', then $\ker(\varphi) = V$.

Let $x, y \in \mathbb{F}_q^n$ be two vectors. Then we can define the *scalar product* (*dot product*) as

$$x \cdot y = xy^T = x_1 y_1 + \cdots + x_n y_n.$$

The vectors x and y are said to be *orthogonal* if $x \cdot y = 0$. For a set $S \subseteq \mathbb{F}_q^n$ we define the *orthogonal complement* as

$$S^\perp = \{x \in \mathbb{F}_q^n \mid x \cdot s = 0 \text{ for all } s \in S\}.$$

A simple computation shows that $S^\perp$ is a subspace. For example, if $S = \varnothing$, then $S^\perp = \mathbb{F}_q^n$, and if $S = \mathbb{F}_q^n$, then $S^\perp = \{0\}$. Furthermore, $S^\perp = \langle S\rangle^\perp$, where $\langle S\rangle$ is the subspace spanned by S. Therefore, it is enough to check orthogonality on a spanning set such as a basis.

Example 3.2. Let $q = 2$ and $S = \{1010, 0101\} \subseteq \mathbb{F}_2^4$. The vectors $x \in S^\perp$ satisfy

$$\begin{aligned} x \cdot (1, 0, 1, 0) = 0 &\iff x_1 + x_3 = 0 \\ x \cdot (0, 1, 0, 1) = 0 &\iff x_2 + x_4 = 0. \end{aligned}$$

Therefore,

$$S^\perp = \{0000, 1010, 0101, 1111\}.$$

[4] In addition to vector spaces over a finite field, we may define codes that are modules over a finite ring.

The elements of S describe what linear dependencies there are between the symbols of $x \in S^\perp$.

Theorem 3.3. *Let $V \subseteq \mathbb{F}_q^n$ be a subspace. Then,* $\dim(V) + \dim(V^\perp) = n$.

Proof. Let $k = \dim(V)$ and $\{v_1, \ldots, v_k\}$ be a basis of V. Let A be a $k \times n$ matrix whose rows are $v_1, \ldots, v_k$. Then, $x \in V^\perp$ if and only if $v_i x^T = 0$ for all $i = 1, \ldots, k$, *i.e.*, if $Ax^T = 0$. The rank of A is k as $v_1, \ldots, v_k$ are linearly independent, so the right nullspace of A has dimension $n - k$ by the rank-nullity theorem. □

Lemma 3.4. *Let $V \subseteq \mathbb{F}_q^n$ be a subspace. Then, $(V^\perp)^\perp = V$.*

Proof. Let $x \in V$. Then $x \cdot y = 0$ for all $y \in V^\perp$, so $x \in (V^\perp)^\perp$. Therefore, $V \subseteq (V^\perp)^\perp$. Furthermore, $\dim((V^\perp)^\perp) = n - \dim(V^\perp) = \dim(V)$, so the subspaces are equal. □

3.2 Linear block codes

A *linear code* is a subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$ over the field $\mathbb{F}_q$. If $\mathcal{C}$ has dimension k ($0 \leqslant k \leqslant n$), then we say it is an $[n, k]_q$-linear code, or $[n, k]$ code for short. As a block code it has size $M = q^k$ and rate $\mathcal{R} = \log_q(q^k)/n = k/n$. If $\mathcal{C}$ has minimum distance d, then we say it is an $[n, k, d]$-linear code.

Example 3.5. Some (trivial) examples of linear codes are $\{0\} \subseteq \mathbb{F}_q^n$ and $\mathbb{F}_q^n$. These codes are $[n, 0, n+1]$ and $[n, n, 1]$ linear codes. The code whose codewords are of the form $(\lambda, \lambda, \ldots, \lambda) \in \mathbb{F}_q^n$ for $\lambda \in \mathbb{F}_q$ is called a *repetition code*. This code is a $[n, 1, n]$-linear code.

The *dual* of a linear code $\mathcal{C}$ is the orthogonal complement $\mathcal{C}^\perp$. From Theorem 3.3 we get that the dimension of $\mathcal{C}^\perp$ is $n - k$, where k is the dimension of $\mathcal{C}$. Recall, that for vectors over the real numbers $x \cdot x = 0$ if and only if $x = 0$, but this is not the case for vectors over finite fields. Therefore, a code $\mathcal{C}$ is said to be *self-orthogonal* if $\mathcal{C} \subseteq \mathcal{C}^\perp$ and *self-dual* if $\mathcal{C} = \mathcal{C}^\perp$.

Example 3.6. Let $S = \{111\} \subseteq \mathbb{F}_2^3$. Then $\mathcal{C} = S^\perp \subseteq \mathbb{F}_2^3$ is a linear code of length 3 defined as

$$\mathcal{C} = \{(x_1, x_2, x_3) \in \mathbb{F}_2^3 \mid x_1 + x_2 + x_3 = 0\}.$$

The dual code $\mathcal{C}^\perp = \langle S \rangle$ is the repetition code of length 3 over $\mathbb{F}_2$. The code $\mathcal{C}$ is defined by the equation $x_1 + x_2 + x_3 = 0$, which corresponds to the dual codeword $y = 111$ ($x \cdot y = 0$).

3.3 Hamming weight

Recall the definition of Hamming distance as the number of places where two vectors differ. Over a vector space, we see that the Hamming distance is translation invariant and invariant under nonzero scalar multiplication, *i.e.*,

- $d(x, y) = d(x + z, y + z)$ for all $x, y, z \in \mathbb{F}_q^n$,
- $d(x, y) = d(\alpha x, \alpha y)$ for all $x, y \in \mathbb{F}_q^n$ and $\alpha \in \mathbb{F}_q^*$.

Define the *Hamming weight* of a word $x \in \mathbb{F}_q^n$ as $\mathrm{wt}(x) = d(x, 0)$ to be the number of nonzero coordinates in x.

Lemma 3.7. *If $x, y \in \mathbb{F}_q^n$, then $d(x, y) = \mathrm{wt}(x - y)$.*

Proof. As the Hamming distance is translation invariant we get that

$$d(x,y) = d(x-y+y, 0+y) = d(x-y, 0) = \mathrm{wt}(x-y). \qquad \square$$

Define the *minimum weight* of a code $\mathcal{C}$ as the minimum weight of a nonzero vector

$$\mathrm{wt}(\mathcal{C}) = \min\{\mathrm{wt}(x) \mid x \in \mathcal{C}, x \neq 0\}.$$

Lemma 3.8. *Let $\mathcal{C}$ be a linear code. Then $d(\mathcal{C}) = \mathrm{wt}(\mathcal{C})$.*

Proof. By definition of minimum distance, there are vectors $x \neq y \in \mathcal{C}$ such that

$$d(\mathcal{C}) = d(x,y) = \mathrm{wt}(x-y) \geqslant \mathrm{wt}(\mathcal{C})$$

since $x - y \in \mathcal{C}$. On the other hand, there is a vector $z \neq 0 \in \mathcal{C}$ such that

$$\mathrm{wt}(\mathcal{C}) = \mathrm{wt}(z) = d(z,0) \geqslant d(\mathcal{C}).$$

Hence, $\mathrm{wt}(\mathcal{C}) = d(\mathcal{C})$. $\square$

The above lemma states that the minimum distance can be computed simply by computing the minimum weight.

3.4 Generator matrix and parity-check matrix

Let $\mathcal{C}$ be an $[n,k,d]$ code over $\mathbb{F}_q$. A $k \times n$ matrix G is said to be a *generator matrix* of $\mathcal{C}$ if the rows of G form a basis of $\mathcal{C}$. Similarly, a $(n-k) \times n$ matrix H is said to be a *parity-check matrix* of $\mathcal{C}$ if it is the generator matrix of the dual $\mathcal{C}^\perp$.

By definition of the dual, every row of G is orthogonal to every row of H, *i.e.*, $GH^T = 0$.

Lemma 3.9. *A vector $x \in \mathbb{F}_q^n$ is contained in a code $\mathcal{C}$ if and only if it can be expressed as $x = mG$ for some $m \in \mathbb{F}_q^k$. On the other hand, $x \in \mathcal{C}$ if and only if $xH^T = 0$.*

Proof. A vector x is in the code if it can be expressed as a linear combination of the rows of G, *i.e.*, if

$$x = m_1 g^1 + \cdots + m_k g^k = mG,$$

where $m = (m_1, \ldots, m_k)$ and $g^1, \ldots, g^k$ are the rows of G.

Let $h^1, \ldots, h^{n-k}$ be the rows of H, *i.e.*, a basis of $\mathcal{C}^\perp$. Then, $x \in \mathcal{C} = (\mathcal{C}^\perp)^\perp$ if and only if $x \cdot h^i = 0$ for all $i = 1, \ldots, n-k$, *i.e.*, if $xH^T = 0$. $\square$

Let G be a generator matrix for linear code $\mathcal{C}$ and let M be a $k \times k$ invertible matrix over $\mathbb{F}_q$. Then, MG is also a generator matrix of $\mathcal{C}$. In fact, all generator matrices of $\mathcal{C}$ are of this form.

Let $g_1, \ldots, g_n$ be the columns of G. A set $\mathcal{I} = \{i_1, \ldots, i_k\} \subseteq [n]$ is an *information set* of $\mathcal{C}$ if the columns $g_{i_1}, \ldots, g_{i_k}$ form a basis of $\mathbb{F}_q^k$. Another way to say this is that the submatrix $G_\mathcal{I}$ consisting of the columns indexed by $\mathcal{I}$ is invertible. A codeword $x \in \mathcal{C}$ is uniquely determined by its projection on an information set, since $x_\mathcal{I} = mG_\mathcal{I}$ and

$$x = mG = mG_\mathcal{I}(G_\mathcal{I})^{-1}G = x_\mathcal{I}(G_\mathcal{I})^{-1}G.$$

Theorem 3.10. *Let $\mathcal{C}$ be a linear code with parity-check matrix H. Then*

- *$d(\mathcal{C}) \geqslant d$ if and only if any $d-1$ columns of H are linearly independent, and*
- *$d(\mathcal{C}) \leqslant d$ if and only if there is a set of d columns of H that are linearly dependent.*

Proof. Let h_i denote the ith column of H. The columns $h_{i_1}, \ldots, h_{i_e}$ are linearly dependent if and only if there exists coefficients $x_{i_1}, \ldots, x_{i_e}$ such that

$$0 = x_{i_1}h_{i_1} + \cdots + x_{i_e}h_{i_e} = x_1h_1 + \cdots + x_nh_n = (xH^T)^T,$$

where $x_i = 0$ if $i \notin \{i_1, \ldots, i_e\}$. This is equivalent to saying that there is a nonzero codeword $x \in \mathcal{C}$ that has only zero coordinates outside of $\{i_1, \ldots, i_e\}$. Such an $x \neq 0$ will have weight $\mathrm{wt}(x) \leqslant e$. The result follows from this. □

A generator matrix G is said to be in *standard form* (or *systematic form*) if $G = (I_k \mid X)$ for some $k \times (n-k)$ matrix X. Here, I_k is the $k \times k$ identity matrix. Similarly, a parity-check matrix H is said to be in standard form if $H = (Y \mid I_{n-k})$ for some $(n-k) \times k$ matrix Y. If $G = (I_k \mid X)$ is in standard form, then $H = (-X^T \mid I_{n-k})$ is a parity-check matrix of the same code. This follows from the fact that

$$GH^T = I_k \cdot (-X) + X \cdot I_{n-k} = 0$$

and G and H have full rank.

Let $m \in \mathbb{F}_q^k$ denote a message. Then we can encode m to a codeword in $\mathcal{C}$ by the mapping $m \mapsto mG$. If G is in standard form, then the first k coordinates of $x = mG$ correspond directly to the message m.

3.5 Decoding of linear codes

During transmission, a codeword $x \in \mathcal{C}$ gets transformed to another word $y = x + e \in \mathbb{F}_q^n$. As we wanted to find the closest codeword $\widehat{x} \in \mathcal{C}$ to the received vector y, we want to find the vector e with the smallest weight, such that $y - e \in \mathcal{C}$. Recall that $y - e \in \mathcal{C}$ if and only if the cosets of y and e match, *i.e.*, $y + \mathcal{C} = e + \mathcal{C}$. A codeword of minimal weight in a coset is called the *coset leader*. Therefore, minimum distance decoding can be performed by finding the coset leader e of the coset of y and setting $\widehat{x} = y - e$.

Let H be the parity-check matrix of a linear code $\mathcal{C}$. The *syndrome* of a word y (with respect to the parity-check matrix H) is

$$S(y) = yH^T.$$

Recall that $y \in \mathcal{C}$ if and only if $S(y) = 0$. Notice that S is linear, so $S(x) = S(y)$ if and only if $x - y \in \mathcal{C}$, *i.e.*, x and y are in the same coset. Therefore, the syndrome can be used to determine the coset of a word. By precomputing all coset leaders and their syndromes, minimum distance decoding can be performed by a simple lookup table. This is known as *syndrome decoding*.

Another decoding method is a probabilistic method called *information set decoding (ISD)*, which involves randomly choosing an information set $\mathcal{I}$ of the code and computing $\widehat{x} = y_{\mathcal{I}}(G_{\mathcal{I}})^{-1}G$. If there are no errors within $\mathcal{I}$, then this produces the correct result. Otherwise, we will not get a codeword that is close to y, so we can try again.

Instead of decoding to the unique closest codeword within the radius $t = \lfloor (d-1)/2 \rfloor$ (if such exists), it may be beneficial to go beyond this radius and finding all codewords. This is known as *list-decoding* and has been shown to produce good performance even when unique decoding is desired.

In general, decoding a randomly chosen linear code is computationally hard. To have efficient algorithms, we need more structure from the codes.

3.6 Erasure coding

Another type of channel is the erasure channel, where instead of errors, we may loose symbols of the transmitted word in known positions. The *binary erasure channel (BEC)* has alphabets $\mathcal{X} = \{0, 1\}$ and $\mathcal{Y} = \{0, 1, ?\}$ and can be described by the transition probabilities

$$\begin{aligned} \mathbb{P}[y = 0 \mid x = 0] &= \mathbb{P}[y = 1 \mid x = 1] = 1 - \varepsilon \\ \mathbb{P}[y = ? \mid x = 1] &= \mathbb{P}[y = ? \mid x = 0] = \varepsilon, \end{aligned}$$

where $\varepsilon \in [0, 1]$. The symbol ? denotes that the sent symbol was *erased*. We can generalize the erasure channel to any alphabet such that the symbol is erased with probability ε and kept the same with probability $1 - \varepsilon$.

Let $\mathcal{J} = \{i \in [n] \mid y_i \neq ?\}$ be the set of coordinates that are not erased in the output of the (n-fold) erasure channel. If $\mathcal{J}$ contains an information set of the code $\mathcal{C}$ that was used, then the sent codeword can be uniquely decoded.

Recall that the codewords in the dual code $\mathcal{C}^\perp$ define linear conditions on the symbols of the codewords. For example, if $0111 \in \mathcal{C}^\perp$, then $x_2 + x_3 + x_4 = 0$ for all codewords $x = (x_1, x_2, x_3, x_4) \in \mathcal{C}$. If the symbols x_3 and x_4 are known, then x_2 can be solved.

Example 3.11. Let $\mathcal{C} = S^\perp \subseteq \mathbb{F}_2^3$, where $S = \{111\}$. If the received word is $y = 1?0$, then $x_1 = 1$ and $x_3 = 0$. The symbols have to satisfy $x_1 + x_2 + x_3 = 0$, so $x_2 = 1$. The sent codeword was $x = 110$.

Example 3.12. A parity-check condition such as $000100001010 \in \mathbb{F}_2^{12}$ is said to have low density, since it has a small number of 1's compared to 0's. By combining multiple such low density parity checks, we can get a code with good erasure correction capabilities. We can correct the erasures in the symbols iteratively by starting with a parity check condition where we know all but one of the required symbols. Such a code construction is known as a *low-density parity-check (LDPC) code*[5].

[5]These codes were discovered in the 1960's, but they were not seen as useful due to the large block lengths required. They were subsequently rediscovered later in the 1990's and have found many applications in wireless communications such as WiFi and 5G.

4 Bounds on the sizes of codes

In coding theory we want to find codes with a high rate and good error correction properties. As the minimum distance of a code affects the error correction, we want to bound the maximal size of a (possibly nonlinear) code with a given minimum distance. The main reference for this section is [LX04, Chapter 5].

4.1 Lower bounds

Consider an alphabet Σ of size q and words in Σ^n. Let $r \geqslant 0$ be an integer and $u \in \Sigma^n$ a word. Then the *Hamming ball* of radius r and center point u is

$$\{v \in \Sigma^n \mid d(u,v) \leqslant r\}.$$

Lemma 4.1. *The Hamming ball contains $V_q^n(r)$ elements, where*

$$V_q^n(r) = \begin{cases} \sum_{i=0}^{r} \binom{n}{i}(q-1)^i & \text{if } 0 \leqslant r \leqslant n \\ q^n & \text{if } r \geqslant n \end{cases}.$$

Proof. If $r \geqslant n$, then the Hamming ball contains all elements in the space Σ^n, so $V_q^n(r) = q^n$. On the other hand, the number of words $v \in \Sigma^n$ such that $d(u,v) = i$ can be computed as follows. There are $\binom{n}{i}$ sets of size i where u and v can differ and $q-1$ choices for each coordinate in this set. Thus, the number of elements at distance i is $\binom{n}{i}(q-1)^i$. The result follows by summing up over all possible $0 \leqslant i \leqslant r$. Notice also that

$$q^n = (q-1+1)^n = \sum_{i=0}^{n} \binom{n}{i}(q-1)^i$$

so the equations agree for $r = n$. □

Given a (possibly nonlinear) code $\mathcal{C} \subseteq \Sigma^n$ of minimum distance d, if there exists a word $v \in \Sigma^n$ whose distance to all codewords is at least d, then $\mathcal{C} \cup \{v\}$ is also a code with minimum distance d. This means that we can always extend the code until the balls of radius $d-1$ centered at the codewords cover the space Σ^n.

Theorem 4.2 (Sphere-covering bound)**.** *There exists an $(n, M, d)_q$-code with*

$$M \geqslant \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i}(q-1)^i}.$$

Proof. As mentioned above, for a code of maximal size, the Hamming balls of radius $d-1$ have to cover Σ^n. Therefore,

$$M \cdot V_q^n(d-1) \geqslant q^n.$$ □

For linear codes we have the Gilbert–Varshamov bound.

Theorem 4.3 (Gilbert–Varshamov bound)**.** *There exists an $[n, k, d]_q$-linear code if*

$$q^k < \frac{q^n}{\sum_{i=0}^{d-2} \binom{n-1}{i}(q-1)^i}.$$

Proof. Let $\mathcal{C}$ be an $[n, k, d]$-linear code over $\mathbb{F}_q$ with an $(n-k) \times n$ parity-check matrix H. Recall from Theorem 3.10 that $\mathcal{C}$ has minimum distance $\geqslant d$ if any $d-1$ columns of H are linearly independent.

We will choose the columns of H in order by the following method. For the first column of H choose any nonzero vector $c_1 \in \mathbb{F}_q^{n-k}$. Say that we have chosen the first j columns $c_1, \ldots, c_j$. The column c_{j+1} cannot be in the span of any $d-1$. The number of vectors in the span of exactly i of the vectors $c_1, \ldots, c_j$ is at most $\binom{j}{i}(q-1)^i$. As $j \leqslant n-1$, we have that the number of vectors that are in the span of at most $d-2$ vectors is

$$\sum_{i=0}^{d-2} \binom{n-1}{i}(q-1)^i < q^{n-k}.$$

Thus, we can always choose the vector c_{j+1} to not be in the span of any $d-2$ of the vectors. Hence, any $d-1$ columns of H are linearly independent. □

4.2 The sphere-packing bound and perfect codes

The above two theorems give a lower bound on the maximal size of a code with given minimum distance. We can also upper bound this quantity.

Theorem 4.4 (Sphere-packing bound)**.** *Let $\mathcal{C}$ be an $(n, M, d)_q$-code. Then,*

$$M \leqslant \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i},$$

where $t = \lfloor (d-1)/2 \rfloor$.

Proof. If there exists a word $v \in \Sigma^n$ such that $d(v, x), d(v, y) \leqslant t$ for two distinct codewords $x \neq y \in \mathcal{C}$, then

$$d(x, y) \leqslant d(x, v) + d(v, y) \leqslant 2t \leqslant d-1.$$

Hence, $\mathcal{C}$ does not have minimum distance d. This means that the Hamming balls of radius t centered at the codewords have to be disjoint. Thus,

$$M \cdot V_q^n(t) \leqslant q^n.$$ □

Codes that achieve the above bound with equality are known as *perfect codes*.

An interesting family of perfect codes are known as *binary Hamming codes*. Let $r \geqslant 2$ be an integer and $n = 2^r - 1$. Let H be the $r \times n$ binary matrix whose columns are all the nonzero vectors in $\mathbb{F}_2^r$. As H has the standard basis vectors as columns, it must be that the rank of H is r, *i.e.*, H is of full rank. Therefore, the code $\mathcal{C} \subseteq \mathbb{F}_q^n$ that has H as parity-check matrix has dimension $k = n - r = 2^r - r - 1$. No two columns of H are linearly dependent by construction,

so the minimum distance of $\mathcal{C}$ is at least 3. Furthermore, the columns of H contain the vectors $100\cdots0$, $010\cdots0$ and $110\cdots0$ which are linearly dependent. Thus, $\mathcal{C}$ has minimum distance $d = 3$. Thus, the binary Hamming codes are $[n = 2^r - 1, k = 2^r - r - 1, d = 3]_2$-linear codes. As the minimum distance is 3, these codes are exactly 1-error-correcting codes.

The sphere-packing bound gives an upper bound of

$$M \leqslant \frac{2^n}{1+n}$$

for $t = 1$ and $q = 2$. Plugging in the values of the Hamming code, we find that the upper bound is $2^{n-r} = 2^k$, so the binary Hamming codes achieve this bound with equality and are perfect codes.

The decoding of binary Hamming codes is particularly simple. Let $x \in \mathcal{C}$ be a codeword and $y = x + e_i$ be the received codeword with 1 error at position i. The syndrome of y is

$$yH^T = xH^T + e_iH^T = e_iH^T = h_i,$$

where h_i is the ith column of H. As the columns of H are unique it is simple to identify the index i and compute $x = y - e_i$.

Example 4.5. Let $r = 3$. Then

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

is a parity-check matrix for the $[7, 4, 3]$ binary Hamming code. Notice that the ith column of H corresponds to the binary expansion of the number i. If $y = x + e_i$ is received, then the syndrome is $yH^T = e_iH^T = h_i$. Now we can directly read the index i from h_i. For example, after receiving the word $y = 0001101$, we get the syndrome $yH^T = 110$. Writing this in decimal gives us $i = 6$ as the location of the corrupted symbol. Therefore, we decode to $x = 0001111$.

4.3 The Singleton bound and MDS codes

The following bound is very fundamental to coding theory.

Theorem 4.6 (Singleton bound)**.** *Let $\mathcal{C}$ be an $(n, M, d)_q$-code. Then,*

$$M \leqslant q^{n-d+1}.$$

Proof. Consider the M codewords of $\mathcal{C}$ and remove the first $d-1$ coordinates from each. As every distinct pair of codewords differ in at least d coordinates, the shortened codewords must still be distinct and have length $n - (d-1) = n - d + 1$. Therefore,

$$M \leqslant |\Sigma^{n-d+1}| = q^{n-d+1}. \qquad \square$$

In particular, if $\mathcal{C}$ is an $[n, k, d]$-linear code over $\mathbb{F}_q$, then

$$k \leqslant n - d + 1$$

by taking logarithm to the base q of both sides.

Codes that achieve the above bound with equality are known as *maximum distance separable (MDS) codes*. An $[n, k]$ MDS code has minimum distance $d = n - k + 1$.

Theorem 4.7. *Let $\mathcal{C}$ be a linear MDS code. Then $\mathcal{C}^\perp$ is MDS.*

Proof. Let $\mathcal{C}$ be an $[n, k]$ MDS code with generator matrix G. Let $m \in \mathbb{F}_q^k$ be a vector and $x = mG$ the corresponding codeword. If $\mathcal{I} \subseteq [n]$, $|\mathcal{I}| = k$, is a set of indices, then $x_\mathcal{I} = mG_\mathcal{I}$, where $G_\mathcal{I}$ is the matrix formed by taking the columns of G indexed by $\mathcal{I}$. If $m \neq 0$, then $x \neq 0$ as G has full rank. As x has at least $n - k + 1$ nonzero entries $x_\mathcal{I} \neq 0$, so $G_\mathcal{I}$ has a trivial left nullspace and full rank. Hence, any set of k columns of G is linearly independent. As G is the parity-check matrix of the dual code $\mathcal{C}^\perp$, we have that $d(\mathcal{C}^\perp) \geqslant k + 1$ by Theorem 3.10. On the other hand, the Singleton bound implies that

$$d(\mathcal{C}^\perp) \leqslant n - (n - k) + 1 = k + 1$$

so $d(\mathcal{C}^\perp) = k + 1$ and $\mathcal{C}^\perp$ is MDS. □

From the above proof we see that any set of k columns of the generator matrix of an MDS code are linearly independent, *i.e.*, any set $\mathcal{I} \subseteq [n]$ of size k is an information set. This means that given any k symbols of a codeword in an MDS code x we can decode the message m.

Example 4.8. Notice that the full space code $\mathbb{F}_q^n$ and the repetition code

$$\mathcal{C} = \{(\lambda, \ldots, \lambda) \in \mathbb{F}_q^n \mid \lambda \in \mathbb{F}_q\}$$

have parameters $[n, n, 1]$ and $[n, 1, n]$, respectively. Therefore, they are MDS codes as they reach the Singleton bound.

Furthermore, their duals are the zero code $\{0\}$ and the *single parity check code*

$$\mathcal{C}^\perp = \{x \in \mathbb{F}_q^n \mid \sum_{i=1}^{n} x_i = 0\},$$

respectively. These codes must also be MDS with parameters $[n, 0, n+1]$ and $[n, n-1, 2]$. Recall that we defined $d(\{0\}) = n + 1$.

The four codes above are known as trivial MDS codes. Later in the course we will see a family of nontrivial MDS codes.

Theorem 4.9. *Let $\mathcal{C}$ be an $[n, k, d]$ MDS code over $\mathbb{F}_q$. If $k \leqslant n-2$, then $k+1 \leqslant q$. Furthermore, if $k \geqslant 2$, then $d \leqslant q$.*

Proof. If $k = 1$, then the statement is clear since $q \geqslant 2$. Therefore, assume that $2 \leqslant k \leqslant n - 2$. Let G be the generator matrix of $\mathcal{C}$. Then any k columns of G are linearly independent, so we may assume that G is in standard form $G = (I \mid X)$, where X is a $k \times (n - k)$ matrix. The rows of G have weight $\leqslant n - k + 1 = d$. As all nonzero codewords in $\mathcal{C}$ have weight $\geqslant d$ we get that the rows have weight exactly $d = n - k + 1$. Therefore, X contains only nonzero entries.

Multiply the rows of G with suitable nonzero factors such that the column at index $k + 1$ is the all ones vector. This is still a generator matrix for $\mathcal{C}$, say G'. Now, assume that there exists

two rows of G' with the same coordinate at position $k+2$. Then, the difference of these two rows has weight $\leqslant n-k<d$ and is nonzero. As this is a contradiction, we must have that the k rows of G' all have a distinct entry in coordinate $k+2$. Therefore, $k \leqslant q-1$.

Finally, let $k \geqslant 2$. Then, $\mathcal{C}^{\perp}$ is an $[n, n-k, k+1]$ MDS code with dimension $n-k \leqslant n-2$, so $d = n-k+1 \leqslant q$. □

Corollary 4.9.1. *All MDS codes over $\mathbb{F}_2$ are trivial.*

Proof. Let $\mathcal{C}$ be an $[n, k, d]$ MDS code over $\mathbb{F}_2$. If $k = n$, then $\mathcal{C} = \mathbb{F}_2^n$, so $\mathcal{C}$ is trivial. Further, if $k = n-1$, then $\mathcal{C}^{\perp}$ is a $[n, 1, n]$ MDS code, *i.e.*, the repetition code of length n. Thus, $\mathcal{C}$ is the single parity check code. Finally, if $k \leqslant n-2$, then $k \leqslant 1$ by the above theorem. If $k = 1$, then $\mathcal{C}$ is the $[n, 1, n]$ repetition code and if $k = 0$, then $\mathcal{C}$ is the zero code. □

The above theorem generalizes to a conjecture that there are no nontrivial MDS codes over $\mathbb{F}_q$ that have $n > q+3$. This conjecture has been proven when q is prime. Therefore, it seems to be the case that it is not possible to construct nontrivial linear MDS codes of arbitrary lengths over a fixed alphabet.

5 Algebraic coding theory

In this section we will look at some algebraic ways of constructing linear codes. As opposed to algebraic methods, so called *modern coding theory* studies iterative algorithms and probabilistic methods as well as codes such as low-density parity-check (LDPC) codes, turbo codes and polar codes. These codes have long block lengths and are used in wireless communications, satellite communications or storage mediums. On the other hand, the algebraic codes that we will consider in this section have shorter block lengths and have applications in distributed storage systems and cryptography. Algebraic codes can be constructed over arbitrary finite fields while modern codes are often only constructed over the binary field. The main reference for this section is [LX04, Chapters 6 and 7].

5.1 Codes from other codes

Before starting to construct linear codes we will look at ways of constructing new codes from a given $[n, k, d]$-linear code $\mathcal{C}$.

The *lengthening* of $\mathcal{C}$ is a $[n+1, k, d]$ code consisting of codewords $(x_1, \ldots, x_n, 0)$, where $x = (x_1, \ldots, x_n) \in \mathcal{C}$. This code is just $\mathcal{C}$ embedded to a larger ambient space $\mathbb{F}_q^{n+1}$.

Subspaces of $\mathcal{C}$ are known as *subcodes*. The minimum distance of a subcode is at least the minimum distance of $\mathcal{C}$ as it is a subset of $\mathcal{C}$. In addition, there exists an $[n, k-1, d]$ subcode of $\mathcal{C}$ that is constructed by taking a codeword $c \in \mathcal{C}$ of weight d and considering a $(k-1)$-dimensional subspace of $\mathcal{C}$ containing c.

Codes that are obtained from $\mathcal{C}$ by (repeatedly) removing one coordinate from each of the codewords is said to be a *punctured* code. Assume $d \geqslant 2$ and let $c \in \mathcal{C}$ be a codeword of weight d. Let $i \in [n]$ be such that $c_i \neq 0$. Consider the code $\mathcal{C}_{[n]\setminus\{i\}} = \{(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \mid x \in \mathcal{C}\}$. The projection map of removing the ith coordinate is injective, since all nonzero codewords in $\mathcal{C}_{[n]\setminus\{i\}}$ contain at least $d-1 \geqslant 1$ nonzero coordinates. Thus, $\mathcal{C}_{[n]\setminus\{i\}}$ is an $[n-1, k, d-1]$-linear code.

We can recap the above three constructions in the following theorem.

Theorem 5.1. *If $\mathcal{C}$ is an $[n, k, d]$-linear code, then there exists an $[n+r, k-s, d-t]$-linear code for any $r \geqslant 0$, $0 \leqslant s \leqslant k-1$ and $0 \leqslant t \leqslant d-1$.*

All of the codes given by the above theorem are worse than the original code $\mathcal{C}$, since they have lower rate or worse minimum distance. On the other hand, the *extended code* of $\mathcal{C}$ is

$$\overline{\mathcal{C}} = \{(x_1, \ldots, x_n, -\sum_{i=1}^{n} x_i) \in \mathbb{F}_q^n \mid (x_1, \ldots, x_n) \in \mathcal{C}\}.$$

Theorem 5.2. *The extended code of an $[n, k, d]$-linear code is an $[n+1, k, d']$-linear code, where $d \leqslant d' \leqslant d+1$.*

Proof. The length and dimension of the code are clear. Furthermore, a parity-check matrix of

the code can be given by

$$H' = \begin{pmatrix} & & & 0 \\ & H & & 0 \\ & & & 0 \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

The first $n-k$ rows correspond to the fact that the first n symbols have to be in $\mathcal{C}$, while the last row means that all the symbols have to sum to zero.

As $d(\mathcal{C}) = d$, we have that any $d-1$ columns of H are linearly independent. Thus, any $d-1$ columns of H' are linearly independent by just looking at the first $n-k$ coordinates. Consider a collection of d linearly dependent columns of H. The corresponding columns of H' and the last column are also linearly dependent. Therefore, there is a set of $d+1$ columns of H' are linearly dependent. Therefore, by Theorem 3.10 $\overline{\mathcal{C}}$ has minimum distance d' with $d \leqslant d' \leqslant d+1$. □

We can also combine codes with the following methods. Given an $[n_1, k_1, d_2]$ code $\mathcal{C}_1$ and an $[n_2, k_2, d_2]$ code $\mathcal{C}_2$ we can define their *direct sum* as

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(x, y) \in \mathbb{F}_q^{n_1+n_2} \mid x \in \mathcal{C}_1, y \in \mathcal{C}_2\}.$$

Theorem 5.3. *The direct sum of an $[n_1, k_1, d_1]$ and an $[n_2, k_2, d_2]$ code is an $[n_1 + n_2, k_1 + k_2, \min\{d_1, d_2\}]$ code.*

Proof. The length and dimension are clear from linear algebra. Notice that $\mathcal{C}_1 \oplus \mathcal{C}_2$ contains codewords of the form $(0, y)$ and $(x, 0)$ for $x \in \mathcal{C}_1$ and $y \in \mathcal{C}_2$, so it contains codewords of weight $\mathrm{wt}(x)$ and $\mathrm{wt}(y)$. Furthermore, if $c = (x, y) \in \mathcal{C}_1 \oplus \mathcal{C}_2$ is a nonzero codeword, then either $x \neq 0$ or $y \neq 0$. Therefore, $\mathrm{wt}(c) \geqslant \min\{d_1, d_2\}$. □

Instead of taking the direct sum of two codes, we may use the so-called $(u, u+v)$-*construction* (also known as the *Plotkin sum*). Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be linear codes of length n. Define

$$\mathcal{C}_1 \boxplus \mathcal{C}_2 = \{(u, u + v) \in \mathbb{F}_q^{2n} \mid u \in \mathcal{C}_1, v \in \mathcal{C}_2\}.$$

Theorem 5.4. *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be $[n, k_1, d_1]$ and $[n, k_2, d_2]$ codes. Then $\mathcal{C}_1 \boxplus \mathcal{C}_2$ is a $[2n, k_1 + k_2, \min\{2d_1, d_2\}]$-linear code.*

Proof. The length of the code is clear. Let $u_1, \ldots, u_{k_1}$ and $v_1, \ldots, v_{k_2}$ be bases for $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Then, the vectors (u_i, u_i) and $(0, v_j)$ for $i = 1, \ldots, k_1$ and $j = 1, \ldots, k_2$ span $\mathcal{C}_1 \boxplus \mathcal{C}_2$. Furthermore, these vectors are linearly independent, since

$$\sum_{i=1}^{k_1} \alpha_i(u_i, u_i) + \sum_{j=1}^{k_2} \beta_j(0, v_j) = 0$$

implies immediately that $\alpha_i = 0$ for all i as the u_i are linearly independent. Furthermore, $\beta_j = 0$ for all j, since the v_j are linearly independent. Thus, the dimension of the code is $k_1 + k_2$.

Let $u, v \in \mathbb{F}_q^n$. Then

$$\mathrm{wt}(v) = \mathrm{wt}(v + u - u) \leqslant \mathrm{wt}(v + u) + \mathrm{wt}(-u) \implies \mathrm{wt}(u + v) \geqslant \mathrm{wt}(v) - \mathrm{wt}(u).$$

Let $x = (u, u+v) \neq 0$ be a codeword. If $v = 0$, then $u \neq 0$ and $\mathrm{wt}(x) = 2\,\mathrm{wt}(u) \geqslant 2d_1$. On the other hand, if $v \neq 0$, then

$$\mathrm{wt}(x) = \mathrm{wt}(u) + \mathrm{wt}(u+v) \geqslant \mathrm{wt}(u) + \mathrm{wt}(v) - \mathrm{wt}(u) = \mathrm{wt}(v) \geqslant d_1.$$

Additionally, if $u \in \mathcal{C}_1$ and $v \in \mathcal{C}_2$ are such that $\mathrm{wt}(u) = d_1$ and $\mathrm{wt}(v) = d_2$, then $x = (u, u), y = (0, v) \in \mathcal{C}_1 \boxplus \mathcal{C}_2$ are such that $\mathrm{wt}(x) = 2d_1$ and $\mathrm{wt}(y) = d_1$. □

The generator matrix of $\mathcal{C}_1 \boxplus \mathcal{C}_2$ is given by

$$G = \begin{pmatrix} G_1 & G_1 \\ 0 & G_2 \end{pmatrix},$$

where G_1 and G_2 are generator matrices for $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.

5.2 Reed–Muller codes

Define $\mathrm{RM}(r, m)$ to be a family of binary codes depending on integers $m \geqslant 0$ and $0 \leqslant r \leqslant m$. First, set $\mathrm{RM}(0, m)$ to be the binary repetition code of length 2^m, *i.e.*,

$$\mathrm{RM}(0, m) = \{(\lambda, \ldots, \lambda) \in \mathbb{F}_2^{2^m} \mid \lambda \in \mathbb{F}_2\}.$$

Then, recursively define for $m \geqslant r \geqslant 1$

$$\mathrm{RM}(r, m) = \begin{cases} \mathbb{F}_2^{2^m} & \text{if } r = m \\ \mathrm{RM}(r, m-1) \boxplus \mathrm{RM}(r-1, m-1) & \text{if } r < m \end{cases}.$$

Theorem 5.5. *The code* $\mathrm{RM}(r, m)$ *for* $m \geqslant 0$ *and* $0 \leqslant r \leqslant m$ *is a* $[2^m, k, 2^{m-r}]$*-linear code over* $\mathbb{F}_2$ *with*

$$k = \sum_{i=0}^{r} \binom{m}{i}.$$

Proof. We will prove the statement by induction on m. The base case is clear, since $\mathrm{RM}(0, 0) = \mathbb{F}_2$ is a $[1, 1, 1]$ code. Furthermore, $\mathrm{RM}(0, m)$ is a $[2^m, 1, 2^m]$ code, so the statement is also clear. Let $m \geqslant 0$ and $0 \leqslant r \leqslant m$ and assume that the statement holds for $\mathrm{RM}(r', m-1)$ for all $0 \leqslant r' \leqslant m-1$. If $r = m$, then $\mathrm{RM}(r, m) = \mathbb{F}_2^{2^m}$ is a $[2^m, 2^m, 1]$ code and

$$\sum_{i=0}^{m} \binom{m}{i} = 2^m,$$

so the code satisfies the theorem. On the other hand, if $r < m$, then

$$\mathrm{RM}(r, m) = \mathrm{RM}(r, m-1) \boxplus \mathrm{RM}(r-1, m-1).$$

By the induction assumption, $\mathrm{RM}(r, m-1)$ is a $[2^{m-1}, \sum_{i=0}^{r} \binom{m-1}{i}, 2^{m-r-1}]$ code and $\mathrm{RM}(r-1, m-1)$ is a $[2^{m-1}, \sum_{i=0}^{r-1} \binom{m-1}{i}, 2^{m-r}]$ code. Thus, by Theorem 5.4 $\mathrm{RM}(r, m)$ is an $[n, k, d]$ code

with $n = 2 \cdot 2^{m-1} = 2^m$, $d = \min\{2 \cdot 2^{m-r-1}, 2^{m-r}\} = 2^{m-r}$ and

$$\begin{aligned} k &= \sum_{i=0}^{r} \binom{m-1}{i} + \sum_{i=0}^{r-1} \binom{m-1}{i} \\ &= 1 + \sum_{i=1}^{r} \binom{m-1}{i} + \sum_{i=1}^{r} \binom{m-1}{i-1} \\ &= 1 + \sum_{i=1}^{r} \binom{m}{i} \\ &= \sum_{i=0}^{r} \binom{m}{i}. \end{aligned}$$

This matches the statement, so we have proven the statement for all m, r by induction. □

Example 5.6. Denote the generator matrix of $\mathrm{RM}(r, m)$ by $G_{r,m}$. Let us find the generator matrix for $\mathrm{RM}(2, 3)$. First, we find

$$G_{0,1} = \begin{pmatrix} 1 & 1 \end{pmatrix}, \quad G_{1,1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Then, by using the recursive formula we get

$$G_{1,2} = \begin{pmatrix} G_{1,1} & G_{1,1} \\ 0 & G_{0,1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad G_{2,2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Finally,

$$G_{2,3} = \begin{pmatrix} G_{2,2} & G_{2,2} \\ 0 & G_{1,2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Another way to construct a generator matrix of $\mathrm{RM}(r, m)$ is by considering the matrix $G_m = G^{\otimes m}$ (the m-fold tensor product of G with itself), where $G = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ and choosing the rows of G_m with weight $\geqslant 2^{m-r}$. This method will maximize the minimum distance of the resulting code. However, it turns out that there is a better way to remove some of the rows of $G^{\otimes m}$ that will achieve the channel capacity on the BSC. Such a code is known as a *polar code* and was introduced in 2009.

5.3 Cyclic codes

A linear code $\mathcal{C}$ is said to be *cyclic* if $(c_1, \ldots, c_n) \in \mathcal{C}$ implies that $(c_n, c_1, \ldots, c_{n-1}) \in \mathcal{C}$. This means that *cyclic shifts* of codewords are still codewords. Denote the cyclic shift of a vector x by $\sigma(x)$.

Example 5.7. The full space code $\mathbb{F}_q^n$ and the repetition code $\{(\lambda, \dots, \lambda) \mid \lambda \in \mathbb{F}_q\}$ are cyclic codes.

Consider the injective linear map $\pi\colon \mathbb{F}_q^n \to \mathbb{F}_q[x]/(x^n-1)$ given by

$$(a_0, \dots, a_{n-1}) \mapsto a_0 + a_1 x + \cdots + a_{n-1}x^{n-1}.$$

If $\mathcal{C} \subseteq \mathbb{F}_q^n$ is a linear code, then $\pi(\mathcal{C})$ is a subspace of $\mathbb{F}_q[x]/(x^n-1)$. Therefore, $\pi(\mathcal{C})$ is an ideal if and only if $x \cdot \pi(\mathcal{C}) \subseteq \pi(\mathcal{C})$. Notice that

$$\begin{aligned} x\pi(c) &= x \cdot (c_0 + c_1 x + \dots c_{n-1}x^{n-1}) \\ &= c_0 x + c_1 x^2 + \cdots + c_{n-1}x^n = c_{n-1} + c_0 x + \cdots + c_{n-2}x^{n-1} \\ &= \pi(\sigma(c)). \end{aligned}$$

Hence, $x \cdot \pi(c) \in \pi(\mathcal{C})$ if and only if $\sigma(c) \in \mathcal{C}$. This means that cyclic codes in $\mathbb{F}_q^n$ correspond to ideals in $\mathbb{F}_q[x]/(x^n-1)$.

Ideals of $\mathbb{F}_q[x]/(x^n-1)$ correspond to ideals of $\mathbb{F}_q[x]$ containing the ideal (x^n-1) (this is true in any commutative ring and any ideal). As ideals of $\mathbb{F}_q[x]$ containing (x^n-1) are generated by the divisors $g \in \mathbb{F}_q[x]$ of x^n-1, we get a one-to-one correspondence between cyclic codes in $\mathbb{F}_q^n$ and monic divisors of $x^n - 1 \in \mathbb{F}_q[x]$. Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a cyclic code. Then $\pi(\mathcal{C}) = \langle g \rangle \subseteq \mathbb{F}_q[x]/(x^n-1)$ for some monic polynomial g that is known as the *generator polynomial* of $\mathcal{C}$.

Let $g \in \mathbb{F}_q[x]$ be a divisor of $x^n - 1$ of degree $\deg(g) = n - k$. Then $\langle g \rangle \subseteq \mathbb{F}_q[x]/(x^n-1)$ has dimension k over $\mathbb{F}_q$ and consists of the equivalence classes of $g \cdot p$ where $p \in \mathbb{F}_q[x]$ has $\deg(p) < k$. Therefore, the subspace has basis $g, x \cdot g, \dots, x^{k-1} \cdot g$. A generator matrix can be given by

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k-1} & 0 & \cdots & 0 \\ 0 & g_0 & \cdots & g_{n-k-2} & g_{n-k-1} & \cdots & 0 \\ \vdots & & \ddots & & & \ddots & \vdots \\ 0 & 0 & \cdots & g_0 & g_1 & \cdots & g_{n-k-1} \end{pmatrix}.$$

Example 5.8. Consider $x^4 - 1 \in \mathbb{F}_3[x]$. We know that $x^4 - 1 = (x^2)^2 - 1 = (x^2-1)(x^2+1) = (x-1)(x+1)(x^2+1)$. As x^2+1 does not have roots in $\mathbb{F}_3$, we know that the above is a factorization of x^4-1 to its irreducible factors. Therefore, x^4-1 has $2^3 = 8$ divisors that can be obtained as the different combinations of the 3 irreducible factors. Each of these factors has an associated cyclic code in $\mathbb{F}_3^4$. The divisors of x^4-1 and their corresponding codes are given by the below table.

$g_1 = 1$	$[4, 4, 1]$
$g_2 = x - 1$	$[4, 3, 2]$
$g_3 = x + 1$	$[4, 3, 2]$
$g_4 = x^2 - 1$	$[4, 2, 2]$
$g_5 = x^2 + 1$	$[4, 2, 2]$
$g_6 = x^3 + x^2 + x + 1$	$[4, 1, 4]$
$g_7 = x^3 - x^2 + x - 1$	$[4, 1, 4]$
$g_8 = x^4 - 1$	$[4, 0, 5]$

These codes are all cyclic codes in $\mathbb{F}_3^4$ with the dimension given by $n - \deg(g_i)$. A cyclic code contains a codeword of weight 1 if and only if it contains the vector $10\cdots0$, *i.e.*, if and only

if the divisor is 1. Therefore, the other cyclic codes have minimum distance $\geqslant 2$. The codes corresponding to g_2, g_3, g_4, g_5 contain a codeword of weight 2, so they all have minimum distance 2. Further, the codes generated by g_6 and g_7 are spanned by one full weight vector so they have minimum distance 4.

6 Reed–Solomon codes

In this section we will look at linear codes coming from polynomial evaluations. These codes include the family of Reed–Muller codes that we have seen in the previous section.

6.1 Polynomial interpolation

Let $n \geqslant 0$ be an integer and consider the vector space of polynomials of degree $< n$ over $\mathbb{F}_q$, denoted by $\mathbb{F}_q[x]^{<n}$. It is clear that the elements $1, x, \ldots, x^{n-1}$ are linearly independent and span $\mathbb{F}_q[x]^{<n}$. Thus, $\dim(\mathbb{F}_q[x]^{<n}) = n$.

Let $\alpha \in \mathbb{F}_q$ and consider the evaluation map

$$\mathbb{F}_q[x] \to \mathbb{F}_q, \quad f \mapsto f(\alpha).$$

It is clear that this map is linear, since $(\lambda f + \mu g)(\alpha) = \lambda f(\alpha) + \mu g(\alpha)$ for $\lambda, \mu \in \mathbb{F}_q$ and $f, g \in \mathbb{F}_q[x]$. Similarly, if $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ are distinct elements, then we can define the linear map, known as the *evaluation map*, as

$$\mathrm{ev}\colon \mathbb{F}_q[x] \to \mathbb{F}_q^n, \quad f \mapsto \mathrm{ev}(f) = (f(\alpha_1), \ldots, f(\alpha_n)).$$

Let $f \in \ker(\mathrm{ev})$, *i.e.*, $f(\alpha_i) = 0$ for $i = 1, \ldots, n$. The polynomial f has n distinct roots, so it must have degree $\geqslant n$ or be the zero polynomial. By restricting the domain of ev to polynomials of degree $< n$, we get that ev is injective and $\dim(\mathrm{ev}(\mathbb{F}_q[x]^{<n})) = \dim(\mathbb{F}_q[x]^{<n}) = n$. This means that given any vector $\beta \in \mathbb{F}_q^n$, there exists a unique polynomial of degree $< n$ such that

$$f(\alpha_i) = \beta_i \quad \text{for all } i = 1, \ldots, n.$$

The above fact is known as *polynomial interpolation*. Let $\beta = e_j$ be a standard basis vector. Then we want to find a polynomial $f_j \in \mathbb{F}_q[x]^{<n}$ such that $f_j(\alpha_i) = 0$ for $i \neq j$ and $f_j(\alpha_j) = 1$. As the first try, let us use

$$f_j = \prod_{\substack{i=1 \\ i \neq j}}^{n} (x - \alpha_i),$$

since this polynomial has roots at the required places and has $\deg(f_j) = n - 1$. It is clear that $f_j(\alpha_j) \neq 0$, so let us redefine f_j by normalizing such that $f_j(\alpha_j) = 1$. Therefore,

$$f_j = \prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{x - \alpha_i}{\alpha_j - \alpha_i}$$

has the property that $f_j(\alpha_i) = 0$ for $i \neq j$ and $f_j(\alpha_j) = 1$. The above polynomials are known as *Lagrange interpolation polynomials* and form a basis for $\mathbb{F}_q[x]^{<n}$ since $\mathrm{ev}(f_j) = e_j$ are linearly independent. For arbitrary $\beta \in \mathbb{F}_q^n$ we can just do linear combinations of the f_j's to find the suitable polynomial that interpolates through the points (α_i, β_i):

$$f = \sum_{j=1}^{n} \beta_j f_j = \sum_{j=1}^{n} \beta_j \prod_{\substack{i=1 \\ i \neq j}}^{n} \frac{x - \alpha_i}{\alpha_j - \alpha_i}.$$

6.2 Univariate polynomial evaluation codes

To use polynomial interpolation in coding theory we can regard the polynomial of degree $< k$ as the message and sample it at k places. From the previous section we know that we can always fit a unique polynomial of degree $< k$ through these points, so we can recover the original polynomial. However, if some of the points gets moved (there are errors), then the associated polynomial will change. To still be able to recover the original polynomial we can instead sample the polynomial at $n \geqslant k$ places. Now, any k of these n places is sufficient to decode the original polynomial in case no errors have occurred. On the other hand, if errors have occurred, then we may look for a degree $< k$ polynomial that passes through the most number of the points. It turns out that this polynomial will be the original polynomial given that sufficiently few errors have occurred. We will formalize this type of code in the following.

Let $0 \leqslant k \leqslant n$ be an integer. Instead of considering the entire space $\mathbb{F}_q[x]^{<n}$ of polynomials of degree $< n$, let us consider the subspace $\mathbb{F}_q[x]^{<k}$. We can now define the *Reed–Solomon code* as

$$\mathrm{RS}_k(\alpha) = \mathrm{ev}(\mathbb{F}_q[x]^{<k}) = \{(f(\alpha_1), \ldots, f(\alpha_n)) \mid f \in \mathbb{F}_q[x], \deg(f) < k\}.$$

Here $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ is the vector of *evaluation points*.

Theorem 6.1. *The code* $\mathrm{RS}_k(\alpha)$ *is an* $[n, k, n-k+1]$ *MDS code.*

Proof. By injectivity of ev, we get that $\dim(\mathrm{ev}(\mathbb{F}_q[x]^{<k})) = k$, so $\mathrm{RS}_k(\alpha)$ has dimension k. Furthermore, let $f \in \mathbb{F}_q[x]^{<k}$ be a nonzero polynomial. Then $\mathrm{ev}(f) \in \mathbb{F}_q^n$ has at most $k-1$ zeros, so $\mathrm{wt}(\mathrm{ev}(f)) \geqslant n-k+1$. On the other hand, the Singleton bound states that the minimum distance of an $[n, k]$ code is at most $n-k+1$. Therefore, Reed–Solomon codes are MDS as they reach the Singleton bound. □

If $f = f_0 + f_1 x + \cdots + f_{k-1}x^{k-1}$, then $f(\alpha) = f_0 + f_1\alpha + \cdots + f_{k-1}\alpha^{k-1}$. Therefore,

$$(f(\alpha_1), \ldots, f(\alpha_n)) = (f_0, \ldots, f_{k-1}) \underbrace{\begin{pmatrix} 1 & \cdots & 1 \\ \alpha_1 & \cdots & \alpha_n \\ \vdots & \ddots & \vdots \\ \alpha_1^{k-1} & \cdots & \alpha_n^{k-1} \end{pmatrix}}_{=G}.$$

Therefore, G is a generator matrix for $\mathrm{RS}_k(\alpha)$. The transpose of G above is known as a *Vandermonde matrix*. It is well known that any $k \times k$ submatrix of an $n \times k$ Vandermonde matrix with distinct evaluation points $\alpha_1, \ldots, \alpha_n$ is invertible. This also follows from the proof of Theorem 4.7, since $\mathrm{RS}_k(\alpha)$ is MDS.

Consider just the points $\alpha_1, \ldots, \alpha_k$ and denote the jth Lagrange interpolation polynomial by λ_j. From the earlier section we know that $f = \sum_{j=1}^{k} f(\alpha_j)\lambda_j$. Therefore,

$$(f(\alpha_1), \ldots, f(\alpha_n)) = (f(\alpha_1), \ldots, f(\alpha_k)) \underbrace{\begin{pmatrix} \lambda_1(\alpha_1) & \ldots & \lambda_1(\alpha_n) \\ \vdots & \ddots & \vdots \\ \lambda_k(\alpha_1) & \ldots & \lambda_k(\alpha_n) \end{pmatrix}}_{=G'}.$$

By using the property that $\lambda_j(\alpha_i) = \delta_{ij}$ we have that the above matrix can be written as

$$G' = \begin{pmatrix} 1 & 0 & \cdots & 0 & \lambda_1(\alpha_{k+1}) & \cdots & \lambda_1(\alpha_n) \\ 0 & 1 & \cdots & 0 & \lambda_2(\alpha_{k+1}) & \cdots & \lambda_2(\alpha_n) \\ \vdots & & \ddots & & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & \lambda_k(\alpha_{k+1}) & \cdots & \lambda_k(\alpha_n) \end{pmatrix}.$$

This is a generator matrix for $\mathrm{RS}_k(\alpha)$ in standard form.

It turns out that the dual of a Reed–Solomon code is not generally a Reed–Solomon code, but it will be equivalent to one. In particular, it will be a *generalized Reed–Solomon code*. Define the generalized Reed–Solomon code as

$$\mathrm{GRS}_k(\alpha, \nu) = \{(\nu_1 f(\alpha_1), \ldots, \nu_n f(\alpha_n)) \mid f \in \mathbb{F}_q[x], \deg(f) < k\},$$

where $\nu \in (\mathbb{F}_q^*)^n$ is a vector with nonzero entries known as the *column multipliers*.

As Reed–Solomon codes of length n require n distinct evaluation points from $\mathbb{F}_q$, we have that $n \leqslant q$. Therefore, Reed–Solomon codes are bounded in length by the alphabet size.

We can consider also so-called *extended Reed–Solomon* codes that are defined by the generator matrix

$$G = \begin{pmatrix} 1 & \cdots & 1 & 0 \\ \alpha_1 & \cdots & \alpha_n & 0 \\ \vdots & \ddots & \vdots & 0 \\ \alpha_1^{k-1} & \cdots & \alpha_n^{k-1} & 1 \end{pmatrix},$$

where $\alpha_1, \ldots, \alpha_n$ range over all elements in $\mathbb{F}_q$. Therefore, this code has length $n = q + 1$ and dimension k. Furthermore, any k of the first n columns are linearly independent. Finally, any $k - 1$ of the first n columns and the last column are also linearly independent. Thus, any k columns of G are linearly independent, so the code generated by G is MDS.

6.3 Star products of codes

We may define a product on vectors $x, y \in \mathbb{F}_q^n$ known as the *star product* by

$$x \star y = (x_1 y_1, \ldots, x_n y_n).$$

Further, we may define the *star product* of linear codes $\mathcal{C}, \mathcal{D} \subseteq \mathbb{F}_q^n$ as

$$\mathcal{C} \star \mathcal{D} = \mathrm{span}\{c \star d \mid c \in \mathcal{C}, d \in \mathcal{D}\}.$$

We take the span in the above definition to make sure that $\mathcal{C} \star \mathcal{D} \subseteq \mathbb{F}_q^n$ is a subspace.

Let $f, g \in \mathbb{F}_q[x]$ be polynomials and $\mathrm{ev}(f), \mathrm{ev}(g)$ be their evaluation vectors at some n fixed points $\alpha_1, \ldots, \alpha_n$. Then, $(\mathrm{ev}(f) \star \mathrm{ev}(g))_i = \mathrm{ev}(f)_i \, \mathrm{ev}(g)_i = f(\alpha_i) g(\alpha_i) = (fg)(\alpha_i) = \mathrm{ev}(fg)_i$, so $\mathrm{ev}(f) \star \mathrm{ev}(g) = \mathrm{ev}(fg)$. Similarly, $\mathrm{ev}(f) + \mathrm{ev}(g) = \mathrm{ev}(f + g)$.

Consider $f \in \mathbb{F}_q[x]^{<k}$ and $g \in \mathbb{F}_q[x]^{<\ell}$. Then, $fg \in \mathbb{F}_q[x]^{<k+\ell-1}$, since

$$\deg(fg) = \deg(f) + \deg(g) \leqslant k - 1 + \ell - 1 < k + \ell - 1.$$

If $k + \ell - 1 > n$, then there is a polynomial h of degree $< n$ that agrees with fg at all points $\alpha_1, \ldots, \alpha_n$. Therefore, $\mathrm{RS}_k(\alpha) \star \mathrm{RS}_\ell(\alpha) \subseteq \mathrm{RS}_{\min\{n,k+\ell-1\}}(\alpha)$. Furthermore, x^m for $m = 0, 1, \ldots, \min\{n, k+\ell-1\} - 1$ can be written as $x^i \cdot x^j = x^{i+j}$ for some $i \in \{0, 1, \ldots, k-1\}$ and $j \in \{0, 1, \ldots, \ell - 1\}$. Hence, $\mathrm{RS}_k(\alpha) \star \mathrm{RS}_\ell(\alpha) = \mathrm{RS}_{\min\{n,k+\ell-1\}}(\alpha)$. This can further be generalized to

$$\mathrm{GRS}_k(\alpha, \nu) \star \mathrm{GRS}_\ell(\alpha, \omega) = \mathrm{GRS}_{\min\{n,k+\ell-1\}}(\alpha, \nu \star \omega).$$

The star products of codes are important in applications such as cryptanalysis, coded computation, and private information retrieval. For more properties of star product codes, see [Ran15].

6.4 Decoding of Reed–Solomon codes

In this section we will describe the Berlekamp–Welch algorithm for correcting errors in a Reed–Solomon code $\mathrm{RS}_k(\alpha)$. The minimum distance of $\mathrm{RS}_k(\alpha)$ is $d = n - k + 1$, so the unique decoding radius is $\lfloor \frac{n-k}{2} \rfloor$. Decoding generalized Reed–Solomon codes can be done by considering the vector $y' = \nu^{-1} \star y$, instead of y.

Let $y \in \mathbb{F}_q^n$ the received vector such that there exists $f \in \mathbb{F}_q[x]^{<k}$ with $d(\mathrm{ev}(f), y) \leqslant \lfloor \frac{n-k}{2} \rfloor$. Let $\mathcal{E} \subseteq [n]$, $|\mathcal{E}| = t$, be the location of the errors, *i.e.*, $y_i = f(\alpha_i)$ if and only if $i \in [n] \setminus \mathcal{E}$. We will define an *error-locator polynomial* that has roots at the error locations α_i for $i \in \mathcal{E}$. In particular, let $e \in \mathbb{F}_q[x]$ be a monic polynomial of degree t. Then we have that

$$y_i e(\alpha_i) = e(\alpha_i) f(\alpha_i)$$

for all $i \in [n]$. It is clear that the equation holds for $i \in [n] \setminus \mathcal{E}$. Furthermore, it is clear that for $i \in \mathcal{E}$, $e(\alpha_i) = 0$, so the equation holds. Define $r = -ef \in \mathbb{F}_q[x]$ to be a polynomial of degree $\leqslant t + k - 1$. We have the equation $y_i e(\alpha_i) + r(\alpha_i) = 0$ for all $i \in [n]$. We may write this as

$$y_i(e_0 + e_1\alpha_i + \ldots e_{t-1}\alpha_i^{t-1}) + (r_0 + r_1\alpha_1 + \cdots + r_{t+k-1}\alpha_i^{t+k-1}) = -\alpha_i^t$$

as e is monic of degree t. The above equation hold for all $i \in [n]$, so we may write

$$(e_0, \ldots, e_{t-1}, r_0, \ldots, r_{t+k-1}) \begin{pmatrix} y_1 & \cdots & y_n \\ y_1\alpha_1 & \cdots & y_n\alpha_n \\ \vdots & \ddots & \vdots \\ y_1\alpha_1^{t-1} & \cdots & y_n\alpha_n^{t-1} \\ 1 & \cdots & 1 \\ \alpha_1 & \cdots & \alpha_n \\ \vdots & \ddots & \vdots \\ \alpha_1^{t+k-1} & \cdots & \alpha_n^{t+k-1} \end{pmatrix} = -(\alpha_1^t, \ldots, \alpha_n^t).$$

By definition of y it is clear that this system has a solution[6] that defines the polynomials e and r. Thus, we may solve $f = -\frac{r}{e}$.

The coefficient matrix is $(2t + k) \times n$, so it will have nontrivial nullspace if $2t + k > n$, *i.e.*, if $t > \frac{n-k}{2}$. On the other hand, we know that there can not be multiple solutions for f if $t \leqslant \lfloor \frac{d-1}{2} \rfloor = \lfloor \frac{n-k}{2} \rfloor$. Thus, if $t \leqslant \lfloor \frac{n-k}{2} \rfloor$, then the above has a unique solution for the message polynomial f.

[6]The received vector y is assumed to contain at most $\lfloor \frac{n-k}{2} \rfloor$ errors.

6.5 Multivariate polynomial evaluation codes

Recall the recursive definition of binary Reed–Muller codes from the previous section. We may also see these codes as evaluation codes of subspaces of $\mathbb{F}_2[x_1, \ldots, x_m]$. Let $n = 2^m$ and $P_1, \ldots, P_n$ be all the points in $\mathbb{F}_2^m$. Then, define

$$\mathrm{RM}(r, m) = \{(f(P_1), \ldots, f(P_n)) \mid f \in \mathbb{F}_2[x_1, \ldots, x_m], \deg(f) \leqslant r\}.$$

Let us show that this definition satisfies the recursive formula for the Reed–Muller codes. From the definition,

$$\mathrm{RM}(0, m) = \{(f(P_1), \ldots, f(P_n)) \mid f \in \mathbb{F}_2[x_1, \ldots, x_m], \deg(f) \leqslant 0\} = \{(\lambda, \ldots, \lambda) \mid \lambda \in \mathbb{F}_2\},$$

since $\deg(f) \leqslant 0$ implies that f is a constant. Furthermore, let $r = m$. Consider the polynomials $f = (x_1 - \alpha_1 - 1) \cdots (x_m - \alpha_m - 1)$ of degree $m = r$. Then $f(P) = 0$ if and only if $P \neq (\alpha_1, \ldots, \alpha_m)$. Hence, the evaluation vector of f is all zeros, except at the point $P_i = (\alpha_1, \ldots, \alpha_m)$. Therefore, the linear code contains the standard basis vectors of $\mathbb{F}_2^{2^m}$ so it is the full space code $\mathbb{F}_2^{2^m}$.

Let $f \in \mathbb{F}_2[x_1, \ldots, x_m]$, $\deg(f) \leqslant r$. As we only care about the evaluation vector of f, we may assume that f contains no higher powers of x_i, since $\beta^j = \beta$ for all $\beta \in \mathbb{F}_2$ and $j = 1, 2, \ldots$. Then we can write $f = g + x_m h$, where $g, h \in \mathbb{F}_2[x_1, \ldots, x_{m-1}]$, $\deg(g) \leqslant r$, $\deg(h) \leqslant r - 1$. Without loss of generality, interpret $P_1, \ldots, P_n$ as binary representations of integers (where $(1, 0, \ldots, 0)$ corresponds to 1) and order them accordingly, *i.e.*, $P_1 = (0, 0, \ldots, 0), P_2 = (1, 0, \ldots, 0), \ldots, P_n = (1, 1, \ldots, 1)$. Therefore,

$$\mathrm{ev}(f) = \mathrm{ev}(g + x_m h) = (\mathrm{ev}_0(g + x_m h), \mathrm{ev}_1(g + x_m h)) = (\mathrm{ev}_0(g), \mathrm{ev}_1(g) + \mathrm{ev}_1(h)),$$

where ev_0 and ev_1 are the evaluation functions on the points where $x_m = 0$ and $x_m = 1$, respectively. Furthermore, $\mathrm{ev}_0(g) = \mathrm{ev}_1(g)$ and $\mathrm{ev}_1(h)$ are the evaluation vectors of g and h on all points in $\mathbb{F}_2^{2^{m-1}}$, since $g, h \in \mathbb{F}_2[x_1, \ldots, x_{m-1}]$. From the final step we see that the codeword is of the form $(u, u + v)$, where $u \in \mathrm{RM}(r, m - 1)$ and $v \in \mathrm{RM}(r - 1, m - 1)$. Therefore, this definition also satisfies the recursive formula, so it matches with the earlier description of Reed–Muller codes.

7 Secret sharing

In this section we will discuss one particular application of coding theory to data security and cryptography, namely secret sharing. Secret sharing was introduced independently by Shamir and Blakley in 1979, and has since seen a lot of attention from the cryptography and coding theory communities. For a general reference to secret sharing, see [Pad12]. For the linear secret sharing section we follow the presentation in [Che+07].

7.1 Perfect security

An encryption scheme turns a message m to a ciphertext c by a random transformation. The randomness of the encryption process may come from a key that can be used to invert the process and obtain the original message from the ciphertext. There are many definitions for the security of such schemes, but we will consider the strongest possible form of security. An encryption scheme is said to have *perfect security* (or *information-theoretic security*) if seeing the ciphertext yields no additional information about the message. One way to state this is that

$$\mathbb{P}[\texttt{ciphertext} = c \mid \texttt{message} = m] = \mathbb{P}[\texttt{ciphertext} = c \mid \texttt{message} = m']$$

for all possible messages m, m' and ciphertext c. The probabilities are taken over the randomness in the encryption process.

One way to achieve perfect security is by *one-time padding*, where the message is encrypted by adding a random *mask* to the message. In particular, let m be a message in some finite additive group G and choose the noise $r \in G$ uniformly at random independently from m. Then, let $c = m + r$ be the ciphertext. Clearly, having c and r allows decoding of $m = c - r$. For all $r \in G$ we have that $\mathbb{P}[\texttt{noise} = r] = \frac{1}{|G|}$. Let $m, m' \in G$ be two messages. Then,

$$\begin{aligned}\mathbb{P}[\texttt{ciphertext} = c \mid \texttt{message} = m] &= \mathbb{P}[\texttt{noise} = c - m] \\ &= \mathbb{P}[\texttt{noise} = c - m'] \\ &= \mathbb{P}[\texttt{ciphertext} = c \mid \texttt{message} = m'].\end{aligned}$$

Therefore, the one-time pad encryption method has perfect security. The group is assumed to be finite for the existence of the uniform distribution.

7.2 Introduction to secret sharing

Secret sharing is a cryptographic method of *sharing* a secret value to a set of n participants such that only some *admissible sets* of participants may reveal the secret, while some *forbidden sets* of participants will not learn anything about the secret. A common choice of the admissible and forbidden sets is determined by a threshold t, meaning that any set of participants of size $\leqslant t$ should not be able to reveal the secret, while any set of participants of size $> t$ will be able to reveal the secret. Such a secret sharing system is known as *threshold secret sharing*. Notice that the admissible sets do not need any additional information, such as an encryption key, in addition to their shares for reconstruction.

Secret sharing has been used to store important cryptographic keys, such as the DNSSEC root

keys, which are needed to reset some certificates on the internet[7]. Secret sharing is advantageous, since these keys are vitally important, so not all of the shares should be needed for reconstruction. On the other hand, sufficiently many of the shares should be needed for reconstruction to not compromise the security of the key. The DNSSEC system uses seven shares, where any five can be used to reconstruct the root key.

Example 7.1 (Shamir secret sharing)**.** Let $s \in \mathbb{F}_q$ be the secret value and choose $f \in \mathbb{F}_q[x]^{<t}$ uniformly at random. Denote $\widehat{f} = s + x \cdot f$ the random polynomial of degree $\leqslant t$ with the property that $\widehat{f}(0) = s$. Let $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ be distinct nonzero values and share $\widehat{f}(\alpha_1), \ldots, \widehat{f}(\alpha_n)$ with the n participants.

As $\deg(\widehat{f}) \leqslant t$, any $t+1$ of the n participants are able to interpolate the polynomial $\widehat{f}$ and reveal $s = \widehat{f}(0)$. Further, consider a set of t participants $i_1, \ldots, i_t \in [n]$. Let $s \in \mathbb{F}_q$. Then there is a unique polynomial of degree $\leqslant t$ running through the points $(0, s), (\alpha_{i_1}, \widehat{f}(\alpha_{i_1})), \ldots, (\alpha_{i_t}, \widehat{f}(\alpha_{i_t}))$. This means that *any* secret value is possible given the shares held by the participants.

7.3 Linear secret sharing

There are many different (and somewhat equivalent) ways of describing general secret sharing schemes. We will follow the presentation in [Che+07, Section 4.2] as it fits nicely within the coding theory topics that we have discussed previously.

We may build a linear secret sharing scheme by choosing two linear codes $\mathcal{S}, \mathcal{C} \subseteq \mathbb{F}_q^n$. Let us denote $\widehat{\mathcal{C}} = \mathcal{S} + \mathcal{C}$. The secret value is $s \in \mathcal{S}$, which means that there are a total of $q^{\dim(\mathcal{S})}$ possible secrets. Furthermore, we choose $c \in \mathcal{C}$ uniformly at random and set $\widehat{c} = s + c$. The n participants are given one coordinate of $\widehat{c}$ each. This looks a bit like the one-time pad encryption scheme, except that the noise is not chosen uniformly at random from the ambient space $\mathbb{F}_q^n$, but rather from the subspace $\mathcal{C}$. However, the security of secret sharing comes from the fact that the noise will still look uniform in the ambient space when we restrict to some subset of the coordinates.

Given $\widehat{c} \in \widehat{\mathcal{C}}$ we may uniquely decode s if and only if $\mathcal{S} \cap \mathcal{C} = \{0\}$, since then the decompositions in $\widehat{\mathcal{C}}$ are unique. This means that $\widehat{\mathcal{C}}$ is a direct sum of $\mathcal{S}$ and $\mathcal{C}$, so $\dim(\widehat{\mathcal{C}}) = \dim(\mathcal{S}) + \dim(\mathcal{C})$.

The idea of secret sharing is that the secret stays hidden to some subsets of the participants, while other subsets can reveal the secret. Let $\mathcal{I} \subseteq [n]$ be a subset of the participants. We say that $\mathcal{I}$ is *forbidden* if the encryption scheme given by ciphertext $\widehat{c}_{\mathcal{I}} = s_{\mathcal{I}} + c_{\mathcal{I}}$ and message s has perfect security. Further, we say that $\mathcal{I}$ is *admissible* if $\widehat{c}_{\mathcal{I}}$ uniquely determines the secret s.

Theorem 7.2. *Let $\mathcal{I} \subseteq [n]$. Then, $\mathcal{I}$ is forbidden if and only if* $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) = \dim(\mathcal{C}_{\mathcal{I}})$*. Further, $\mathcal{I}$ is admissible if and only if* $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) = \dim(\mathcal{S}) + \dim(\mathcal{C}_{\mathcal{I}})$.

Proof. Forbidden sets:

$\boxed{\Leftarrow}$ Let $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) = \dim(\mathcal{C}_{\mathcal{I}})$. Then $\widehat{\mathcal{C}}_{\mathcal{I}} = \mathcal{C}_{\mathcal{I}}$ and the noise $c_{\mathcal{I}} \in \mathcal{C}_{\mathcal{I}}$ is distributed uniformly, so the encryption is a one-time pad. Thus, $\mathcal{I}$ is forbidden.

[7] Clay Dillow. *An Order of Seven Global Cyber-Guardians Now Hold Keys to the Internet.* `https://www.popsci.com/technology/article/2010-07/order-seven-cyber-guardians-around-world-now-hold-keys-internet/`. Accessed: 2024-05-05. 2010.

$\boxed{\Rightarrow}$ As $\widehat{\mathcal{C}}_{\mathcal{I}} = \mathcal{S}_{\mathcal{I}} + \mathcal{C}_{\mathcal{I}}$, it is clear that $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) \geqslant \dim(\mathcal{C}_I)$. Assume that $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) > \dim(\mathcal{C}_{\mathcal{I}})$. Then, there is $s \in \mathcal{S}$ and $c \in \mathcal{C}$ such that $s_{\mathcal{I}} + c_{\mathcal{I}} \notin \mathcal{C}_{\mathcal{I}}$ so $s_{\mathcal{I}} \notin \mathcal{C}_{\mathcal{I}}$. The ciphertexts of $s_{\mathcal{I}}$ will therefore not contain 0, but 0 is a possible ciphertext for $s = 0$. Thus, the ciphertext 0 does not have the same probability to be generated for the messages s and 0. This means that the set $\mathcal{I}$ is not forbidden, as the encryption scheme does not have perfect security.

Admissible sets:

$\boxed{\Leftarrow}$ Let $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) = \dim(\mathcal{S}) + \dim(\mathcal{C}_{\mathcal{I}})$. Then,

$$\begin{aligned}\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) &= \dim(\mathcal{S}_{\mathcal{I}}) + \dim(\mathcal{C}_{\mathcal{I}}) - \dim(\mathcal{S}_{\mathcal{I}} \cap \mathcal{C}_{\mathcal{I}}) \\ &\leqslant \dim(\mathcal{S}) + \dim(\mathcal{C}_{\mathcal{I}}).\end{aligned}$$

Therefore, we must have that $\dim(\mathcal{S}_{\mathcal{I}}) = \dim(\mathcal{S})$ and $\mathcal{S}_{\mathcal{I}} \cap \mathcal{C}_{\mathcal{I}} = \{0\}$. Hence, $\widehat{c}_{\mathcal{I}} = s_{\mathcal{I}} + c_{\mathcal{I}}$ for unique $s_{\mathcal{I}} \in \mathcal{S}_{\mathcal{I}}$ and $c_{\mathcal{I}} \in \mathcal{C}_{\mathcal{I}}$. Finally, $s_{\mathcal{I}}$ uniquely determines $s \in \mathcal{S}$ as the projection to $\mathcal{S}_{\mathcal{I}}$ is injective.

$\boxed{\Rightarrow}$ As $\widehat{\mathcal{C}}_{\mathcal{I}} = \mathcal{S}_{\mathcal{I}} + \mathcal{C}_{\mathcal{I}}$, it is clear that $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) \leqslant \dim(S) + \dim(\mathcal{C}_{\mathcal{I}})$. Assume that $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) < \dim(S) + \dim(\mathcal{C}_{\mathcal{I}})$. Then, either $\mathcal{S}_{\mathcal{I}} \cap \mathcal{C}_{\mathcal{I}} \neq \{0\}$ or $\dim(S_{\mathcal{I}}) < \dim(S)$. In the first case, there is $0 \neq s \in \mathcal{S}$ and $c \in \mathcal{C}$ such that $s_{\mathcal{I}} = c_{\mathcal{I}}$. Then, $\widehat{c} = s + 0$ and $\widehat{c}' = 0 + c$ have the property that $\widehat{c}_{\mathcal{I}} = s_{\mathcal{I}} = c_{\mathcal{I}} = \widehat{c}'_{\mathcal{I}}$, so $s \neq 0$ cannot be uniquely determined from $\widehat{c}_{\mathcal{I}}$. In the second case, there is $0 \neq s \in \mathcal{S}$ such that $s_{\mathcal{I}} = 0$. Then, $\widehat{c} = s + 0$ and $\widehat{c}' = 0 + 0$ have the property that $\widehat{c}_{\mathcal{I}} = s_{\mathcal{I}} = 0 = \widehat{c}'_{\mathcal{I}}$, so $s \neq 0$ cannot be uniquely determined from $\widehat{c}_{\mathcal{I}}$. □

Corollary 7.2.1. *Let $\mathcal{I} \subseteq [n]$. Then $\mathcal{I}$ is forbidden if $|\mathcal{I}| < d(\mathcal{C}^\perp)$. Further, $\mathcal{I}$ is admissible if $|\mathcal{I}| > n - d(\widehat{\mathcal{C}})$.*

Proof. If $|\mathcal{I}| < d(\mathcal{C}^\perp)$, then the projection from $\mathcal{C}$ onto $\mathcal{C}_{\mathcal{I}}$ is surjective by Theorem 3.10. Hence, $\dim(\mathcal{C}_{\mathcal{I}}) = |\mathcal{I}| \geqslant \dim(\widehat{\mathcal{C}}_{\mathcal{I}})$. It is also clear that $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) \geqslant \dim(\mathcal{C}_{\mathcal{I}})$, so $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) = \dim(\mathcal{C}_{\mathcal{I}})$. Hence, $\mathcal{I}$ is forbidden.

If $|\mathcal{I}| > n - d(\widehat{\mathcal{C}})$, then the projection of $\widehat{\mathcal{C}}$ onto $\widehat{\mathcal{C}}_{\mathcal{I}}$ is injective, since otherwise $\widehat{\mathcal{C}}$ would contain a nonzero codeword of weight $\leqslant n - |\mathcal{I}| < d(\widehat{\mathcal{C}}_{\mathcal{I}})$. Therefore,

$$\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) = \dim(\widehat{\mathcal{C}}) = \dim(\mathcal{S}) + \dim(\mathcal{C}) \geqslant \dim(\mathcal{S}) + \dim(\mathcal{C}_{\mathcal{I}}).$$

It is also clear that

$$\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) \leqslant \dim(\mathcal{S}_{\mathcal{I}}) + \dim(\mathcal{C}_{\mathcal{I}}) \leqslant \dim(\mathcal{S}) + \dim(\mathcal{C}_{\mathcal{I}}),$$

so $\dim(\widehat{\mathcal{C}}_{\mathcal{I}}) = \dim(\mathcal{S}) + \dim(\mathcal{C}_{\mathcal{I}})$. Hence, $\mathcal{I}$ is admissible. □

Example 7.3. In Shamir secret sharing we choose $\mathcal{S} = \{(\lambda, \ldots, \lambda) \mid \lambda \in \mathbb{F}_q\}$ to be the repetition code. Further, $\mathcal{C} = \{\mathrm{ev}(x \cdot p) \mid p \in \mathbb{F}_q[x], \deg(p) < t\}$, where $\mathrm{ev}\colon \mathbb{F}_q[x] \to \mathbb{F}_q^n$ denotes the evaluation map on the points $\alpha_1, \ldots, \alpha_n$. We may also write $\mathcal{S} = \mathrm{RS}_1(\alpha)$ and $\mathcal{C} = \alpha \star \mathrm{RS}_t(\alpha) = \mathrm{GRS}_t(\alpha, \alpha)$. Then, $\widehat{\mathcal{C}} = \mathcal{S} + \mathcal{C} = \mathrm{RS}_{t+1}(\alpha)$. By Corollary 7.2.1 any set of size $\leqslant d(\mathcal{C}^\perp) - 1 = (t+1) - 1 = t$ is forbidden, while any set of size $> n - d(\widehat{\mathcal{C}}) = n - (n - (t+1) - 1) = t$ is admissible.

Remark 7.3.1. Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code. As we want to find $\mathcal{S} \subseteq \mathbb{F}_q^n$ such that $\mathcal{S} \cap \mathcal{C} = \{0\}$, we may choose $\mathcal{S} = \operatorname{span}\{s\}$, where $s \neq 0$ and $\operatorname{wt}(s) \leqslant d(\mathcal{C}) - 1$. This makes $\mathcal{S}$ a one-dimensional code that is guaranteed to intersect $\mathcal{C}$ trivially. In general, if $\mathcal{S}$ is supported on some $\leqslant d(\mathcal{C}) - 1$ coordinates, then $\mathcal{S} \cap \mathcal{C} = \{0\}$.

7.4 Homomorphic secret sharing

In many applications of secret sharing we are interested in computing with data instead of just storing and retrieving data. In particular, given the secret shares of some secrets we want to be able to compute the secret share of a function of these secrets. This type of secret sharing is known as *homomorphic secret sharing*.

Example 7.4. Let $s^{(1)}, s^{(2)} \in \mathbb{F}_q$ denote two secret values and $\widehat{f}^{(1)}, \widehat{f}^{(2)}$ be their Shamir secret sharing polynomials of degree $\leqslant t$ whose evaluations are shared to the participants. Therefore, $(\widehat{f}^{(1)} + \widehat{f}^{(2)})(0) = s^{(1)} + s^{(2)}$ and $(\widehat{f}^{(1)}\widehat{f}^{(2)})(0) = s^{(1)}s^{(2)}$. The participants can compute

$$(\widehat{f}^{(1)} + \widehat{f}^{(2)})(\alpha_i) = \widehat{f}^{(1)}(\alpha_i) + \widehat{f}^{(2)}(\alpha_i)$$

and

$$(\widehat{f}^{(1)}\widehat{f}^{(2)})(\alpha_i) = \widehat{f}^{(1)}(\alpha_i)\widehat{f}^{(2)}(\alpha_i).$$

from their local data. Therefore, we may interpolate the polynomials $\widehat{f}^{(1)} + \widehat{f}^{(2)}$ (degree $\leqslant t$) and $\widehat{f}^{(1)}\widehat{f}^{(2)}$ (degree $\leqslant 2t$) from any $t+1$ and $2t+1$ shares, respectively.

Let $\widehat{c}^{(1)} = s^{(1)} + c^{(1)}$ and $\widehat{c}^{(2)} = s^{(2)} + c^{(2)}$ denote two secret shares. Then, $\widehat{c}^{(1)} + \widehat{c}^{(2)} = (s^{(1)} + s^{(2)}) + (c^{(1)} + c^{(2)})$ is a secret share of the sum of the secrets $s^{(1)} + s^{(2)} \in \mathcal{S}$. Indeed, the noise term $c^{(1)} + c^{(2)} \in \mathcal{C}$ will also be uniformly distributed. Furthermore, the participants can compute their shares using just their local data as $\widehat{c}_i^{(1)} + \widehat{c}_i^{(2)}$.

Scalar multiplications of the secret by a public scalar $\lambda \in \mathbb{F}_q$ are similarly easy. In particular, $\lambda\widehat{c} = \lambda s + \lambda c$ is a secret share of λs, where the noise term $\lambda c \in \mathcal{C}$ is uniformly distributed provided that $\lambda \neq 0$. This means that any linear function of the secret shares can easily be computed from the shares held by the participants.

Instead of scalar multiplication, we may consider multiplication by another (public) vector. In particular, let $\mathcal{E} = \operatorname{span}\{e\}$ for some $e \neq 0$ and secret share it with $\widehat{d} = e + d$, where $d \in \mathcal{D}$ is chosen uniformly at random. If $c \in \mathcal{C}$, then $c \star \widehat{d} = c \star e + c \star d$, which looks a bit like the secret share of $c \star e$, where the noise is drawn from the space $\mathcal{C} \star \mathcal{D}$. However, $c \star d \in \mathcal{C} \star \mathcal{D}$ is not distributed uniformly at random in the subspace. Nevertheless, if $\operatorname{wt}(e) \leqslant d(\mathcal{C} \star \mathcal{D}) - 1$, then $c \star e$ can be uniquely decoded from $c \star \widehat{d}$ as $(\mathcal{C} \star \mathcal{E}) \cap (\mathcal{C} \star \mathcal{D}) = \{0\}$. This follows directly from Remark 7.3.1.

If we have two secret shares coming from the (potentially different) codes $\mathcal{S}^{(1)}, \mathcal{S}^{(2)}$ and $\mathcal{C}^{(1)}, \mathcal{C}^{(2)}$. Then the star product of the shares will be

$$(s^{(1)} + c^{(1)}) \star (s^{(2)} + c^{(2)}) = s^{(1)} \star s^{(2)} + s^{(1)} \star c^{(2)} + c^{(1)} \star s^{(2)} + c^{(1)} \star c^{(2)}.$$

The first term here contains the information we care about, namely $s^{(1)} \star s^{(2)}$, while the last three terms contain some noise. In this case it is a bit more difficult to control the intersection between the defining codes of this secret sharing scheme. In particular, the dimension of the space $\mathcal{C}^{(1)} \star \mathcal{C}^{(2)}$ may be very large for generic codes $\mathcal{C}^{(1)}$ and $\mathcal{C}^{(2)}$.

8 Private information retrieval

One application of homomorphic secret sharing is private information retrieval, where the goal is to retrieve one file from a database without revealing the index of that file to the database owner. For a reference on private information retrieval coming from linear codes, see [Fre+17].

8.1 Private information retriveal from homomorphic secret sharing

Consider a distributed storage system consisting of n nodes (=servers). We store m files in the storage system, where each file $x^1, \dots, x^m$ is a vector in $\mathbb{F}_q^k$. The files are encoded to vectors $y^1, \dots, y^m$ using an $[n, k]$ linear code $\mathcal{C}$ such that each node holds one coordinate of $y^1, \dots, y^m$.

To retrieve one of these files, say file x^ℓ, we can compute the linear combination

$$y^\ell = \sum_{j \in [m]} \delta_{\ell,j} y^j,$$

where $\delta_{\ell,j} = 1$ when $\ell = j$ and 0 otherwise. In particular, it is sufficient to recover the coordinates of y^j in some information set of $\mathcal{C}$, since these uniquely determine the codeword y^j and therefore the associated file x^j. We will retrieve the coordinates iteratively over some number of rounds. Let $J \subseteq [n]$ be some subset and $\mathbb{1}_J$ be the vector that has 1's at J and 0's elsewhere. To obtain the coordinates in J, we may compute

$$y^\ell \star \mathbb{1}_J = \sum_{j \in [m]} \delta_{\ell,j} \mathbb{1}_J \star y^j = \sum_{j \in [m]} q^j \star y^j,$$

where $q^j = \delta_{\ell,j} \mathbb{1}_J$.

In *private information retrieval* (*PIR*), we want to retrieve one of the files without revealing the index of that file in the database. To do this, we may use secret sharing. In particular, we secret share the coefficients $q^j \in \mathcal{S} = \operatorname{span}\{\mathbb{1}_J\}$. Let $\mathcal{D} \subseteq \mathbb{F}_q^n$ be a linear code and choose codewords $d^j \in \mathcal{D}$ uniformly at random. The shares corresponding to the jth file are then $\widehat{q}^j = q^j + d^j$. The ith coordinate of these shares are given to the ith node. All the nodes compute the result

$$\sum_{j \in [m]} \widehat{q}^j \star y^j = \sum_{j \in [m]} \delta_{\ell,j} \mathbb{1}_J \star y^j + \sum_{j \in [m]} y^j \star d^j = y^\ell \star \mathbb{1}_J + \underbrace{\sum_{j \in [m]} y^j \star d^j}_{\in \mathcal{C} \star \mathcal{D}}.$$

As long as $|J| = \operatorname{wt}(\mathbb{1}_J) \leqslant d(\mathcal{C} \star \mathcal{D}) - 1$, we can always decode $y^j \star \mathbb{1}_J$ from the result (this follows directly from Remark 7.3.1). Thus, we can get any $d(\mathcal{C} \star \mathcal{D}) - 1$ coordinates of y^ℓ in one iteration. As we need k coordinates, we can repeat the process $s = \frac{k}{d(\mathcal{C} \star \mathcal{D}) - 1}$ many times[8] for different choices of J, such that the coordinates cover an information set. In each iteration we download a total of n symbols in $\mathbb{F}_q$. The *rate* of the PIR scheme will be

$$\mathcal{R} = \frac{k}{sn} = \frac{d(\mathcal{C} \star \mathcal{D}) - 1}{n}.$$

Recall from Corollary 7.2.1, that observing any $d(\mathcal{D}^\perp) - 1$ coordinates of the queries $\widehat{q}^j$ reveals nothing about the secret values $\delta_{\ell,j} \mathbb{1}_J$, so the index of the desired file x^ℓ is kept secret.

[8] We do not have to assume divisibility of k and $d(\mathcal{C} \star \mathcal{D}) - 1$. Instead, we can assume that the file consists of $b = d(\mathcal{C} \star \mathcal{D}) - 1$ "fragments", which are each encoded to a codeword in $\mathcal{C}$. Then, we need to retrieve bk symbols to assemble the entire file.

8.2 Private information retrieval from Reed–Solomon codes

In the description in the previous section, we may choose the codes $\mathcal{C} = \mathrm{RS}_k(\alpha)$ and $\mathcal{D} = \mathrm{RS}_t(\alpha)$. Then the rate of the scheme will be

$$\mathcal{R} = \frac{d(\mathcal{C} \star \mathcal{D}) - 1}{n} = \frac{n - (k + t - 1)}{n},$$

since $\mathcal{C} \star \mathcal{D} = \mathrm{RS}_k(\alpha) \star \mathrm{RS}_t(\alpha) = \mathrm{RS}_{k+t-1}(\alpha)$ assuming that $k + t - 1 \leqslant n$. Furthermore, the index of the desired file is kept secret from any $d(\mathcal{D}^{\perp}) - 1 = t$ nodes by Corollary 7.2.1, since $\mathcal{D}$ is MDS.

References

[Che+07] Hao Chen et al. "Secure computation from random error correcting codes". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2007, pp. 291–310.

[Dil10] Clay Dillow. *An Order of Seven Global Cyber-Guardians Now Hold Keys to the Internet*. `https://www.popsci.com/technology/article/2010-07/order-seven-cyber-guardians-around-world-now-hold-keys-internet/`. Accessed: 2024-05-05. 2010.

[Fre+17] Ragnar Freij-Hollanti et al. "Private information retrieval from coded databases with colluding servers". In: *SIAM Journal on Applied Algebra and Geometry* 1.1 (2017), pp. 647–664.

[Kop23] Swastik Kopparty. *Lecture Notes: Topics in Finite Fields*. `https://www.math.toronto.edu/swastik/courses/finitefields-F23/`. 2023.

[LX04] San Ling and Chaoping Xing. *Coding theory: a first course*. Cambridge University Press, 2004.

[Pad12] Carles Padró. "Lecture notes in secret sharing". In: *Cryptology ePrint Archive* (2012).

[Ran15] Hugues Randriambololona. "On products and powers of linear codes under componentwise multiplication". In: *Algorithmic arithmetic, geometry, and coding theory* 637 (2015), pp. 3–78.