A short introduction to secret sharing from linear codes

Pierre Karpman

December 14, 2021

These short lecture notes give a simple introduction to linear secret sharing from a coding-theory perspective. They start by presenting threshold schemes based on MDS codes, and then discuss the topic of more general access structures and their relation with minimal codewords. They conclude with a short presentation of a simple perfectly-secure message transmission protocol.

All of the studied schemes are analysed in a so-called *information-theory setting*. This means that the adversary is computationally unbounded, and that security will follow from statistical arguments.

1 Preliminaries: uniform masking in a finite group, independence

We start with a definition of independence for random variables, and then state two useful lemmas.

Definition 1 (Independence of random variables)**.** Two random variables X_1, X_2 over a finite set $\mathcal{S}$ are said to be statistically *independent* if and only if for all $\alpha_1, \alpha_2 \in \mathcal{S}$, $\Pr[X_1 = \alpha_1 : X_2 = \alpha_2] = \Pr[X_1 = \alpha_1]$ (or equivalently $\Pr[X_1 = \alpha_1 \wedge X_2 = \alpha_2] = \Pr[X_1 = \alpha_1] \times \Pr[X_2 = \alpha_2]$).

More generally, n random variables $X_1, \ldots, X_n$ are said to be statistically mutually independent if and only if for all $i \in [\![1, n]\!]$, for all $\alpha_1, \ldots, \alpha_n \in \mathcal{S}$, $\Pr[X_i = \alpha_i : \wedge_{j\neq i}^{n} X_j = \alpha_j] = \Pr[X_i = \alpha_i]$ (or equivalently for all $\Pr[\wedge_i^n X_i = \alpha_i] = \prod_i^n \Pr[X_i = \alpha_i]$).*

Lemma 2. *Let $\varphi : \mathcal{S} \to \mathcal{S}'$ be a bijection between two equally-sized finite sets $\mathcal{S}$ and $\mathcal{S}'$. Then if X is a uniform random variable over $\mathcal{S}$, $\varphi(X)$ is a uniform random variable over $\mathcal{S}'$.*

Proof. We write $\varphi^{-1} : \mathcal{S}' \to \mathcal{S}$ the inverse application of φ. It then suffices to compute the distribution of $\varphi(X)$ which for all $\alpha \in \mathcal{S}'$ is equal to $\Pr[\varphi(X) = \alpha] = \Pr[X = \varphi^{-1}(\alpha)] = 1/\#\mathcal{S} = 1/\#\mathcal{S}'$. □

Lemma 3 (Equivalence of statistical and linear independence)**.** *Let $\vec{X} = \begin{pmatrix} X_1 & \cdots & X_k \end{pmatrix}$ be a vector of k uniform and mutually independent random variables over a finite field $\mathbb{F}_q$, and $\boldsymbol{M} \in \mathbb{F}_q^{k\times n}$, $n \leq k$. Then the n random variables in the vector $\vec{Y} = \begin{pmatrix} Y_1 \cdots Y_n \end{pmatrix}$ computed as $\vec{Y} = \vec{X}\boldsymbol{M}$ are uniform and independent if and only if $\boldsymbol{M}$ is of full rank n.*

Proof. ⇐ If $\boldsymbol{M}$ is not full-rank, then its right kernel is of dimension at least 1. Let $\boldsymbol{\lambda}$ be a non-null vector in this kernel, then either it is of weight one and then one of the Y_i's follows a constant distribution, or it may be written (up to scalar multiplication) as $\boldsymbol{e}_i + \boldsymbol{\lambda}'$ for some i, where $\boldsymbol{e}_i$ is a (column) vector of the canonical basis and $\boldsymbol{\lambda}'$ is not null.

*Note that when one considers *events* rather than random variables, the generalisation to n events must consider arbitrary partitions as well.

This latter case means that there exists $i \in [\![1, n]\!]$ s.t. the i^{th} column $\boldsymbol{M}_{*,i}$ is equal to a non-trivial linear combination $\boldsymbol{M}\boldsymbol{\lambda}'$ of some other columns. It follows that $Y_i = \vec{X}\boldsymbol{M}_{*,i}$ is equal to $\vec{X}(\boldsymbol{M}\boldsymbol{\lambda}') = (\vec{X}\boldsymbol{M})\boldsymbol{\lambda}' = \vec{Y}\boldsymbol{\lambda}'$ and the n variables $Y_1, \ldots, Y_n$ are thus not mutually independent.

$\Rightarrow$ We first remark that it is equivalent for $X_1, \ldots, X_k$ to be uniform over $\mathbb{F}_q$ and mutually independent and for $\vec{X}$ to be uniform over $\mathbb{F}_q^k$: in one direction, mutual independence and uniformity gives in particular that for all $\boldsymbol{\alpha} = \begin{pmatrix}\boldsymbol{\alpha}_1 & \cdots & \boldsymbol{\alpha}_k\end{pmatrix}$, $\Pr[\vec{X} = \boldsymbol{\alpha}] = \Pr[\wedge_{i=1}^k X_i = \boldsymbol{\alpha}_i] = \prod_{i=1}^k \Pr[X_i = \boldsymbol{\alpha}_i] = q^{-k}$; in the other it follows from the fact that for all $i \in [\![1, k]\!]$, for all $\boldsymbol{\alpha} \in \mathbb{F}_q^k$, $\Pr[X_i = \boldsymbol{\alpha}_i : \wedge_{j\neq i}^k X_j = \boldsymbol{\alpha}_j] = 1/q = \Pr[X_i = \boldsymbol{\alpha}_i]$.

Now we assume without loss of generality that $n = k$. Then by assumption $\boldsymbol{M}$ is invertible, and it admits a unique inverse $\boldsymbol{M}^{-1}$. Since $X_1, \ldots, X_k$ are uniform and mutually independent, $\vec{X}$ is a uniform random variable over $\mathbb{F}_q^k$ and, applying Lemma 2, so is $\vec{Y}$ and it finally follows from the above that $Y_1, \ldots, Y_k$ are mutually independent. □

We now recall an essential result about the distribution of the sum of two independent random variables over finite groups.

Let $(\mathbb{G}, +)$ be a finite group of order N; X, Y be independent random variables over $\mathbb{G}$. The distribution of $X + Y$ is given by:

$$\Pr[X + Y = \alpha] = \sum_{\beta\in\mathbb{G}} \Pr[X = \beta] \cdot \Pr[Y = \alpha - \beta], \tag{1}$$

for any $\alpha \in \mathbb{G}$. This is nothing but the *discrete convolution* of X and Y.

Now if Y is uniformly distributed, (1) simplifies to:

$$\Pr[X + Y = \alpha] = \sum_{\beta\in\mathbb{G}} \Pr[X = \beta] \cdot \frac{1}{N} = \frac{1}{N}, \tag{2}$$

i.e. the distribution of $X + Y$ is uniform regardless of the distribution of X, which can be arbitrary. Still assuming that Y is uniform, we also get the important property that the *a posteriori* knowledge on the distribution of X after learning a sample of $X + Y$ is equal to its *a priori* knowledge. This is easy to show using Bayes' formula since for every $\alpha \in \mathbb{G}$ one has that:

$$\Pr[X = \alpha \,|\, X + Y = \beta] = \frac{\Pr[X + Y = \beta \,|\, X = \alpha] \cdot \Pr[X = \alpha]}{\Pr[X + Y = \beta]} = \Pr[X = \alpha]. \tag{3}$$

It is also easy to show that the above is true for all (α, β) pair iff. X or Y is uniform.

In the case where both X and Y are uniform over a finite field, the result also follows from the independence of X and $X + Y$, which can be shown as a direct application of Lemma 3. We develop this in the following Example 4.

Example 4. Let X, Y be two uniform and independent random variables over some finite field, then applying Lemma 3 to the matrix $\begin{pmatrix}1 & 0 & 1\\ 0 & 1 & 1\end{pmatrix}$ one gets that X, Y and $X + Y$ are pairwise independent but not mutually independent uniform random variables. This easily generalises to more than two input variables, and one for instance similarly gets that X, Y, Z and $X + Y + Z$ are independent *at order three* (meaning that any three of them are mutually independent), but not mutually independent.

This example summarises much of the applications of Lemma 3 we will be interested in: by forming random variables from well-chosen linear combinations of uniform, mutually independent random variables, one obtains not mutually independent random variables whose dependencies may be carefully controlled.

2 Threshold secret sharing

Definition 5 (Threshold secret-sharing)**.** We define a threshold secret-sharing scheme in the following, slightly informal way:

A (k, n), $k \in [\![2, n]\!]$, secret-sharing scheme over a finite set $\mathcal{S}$ is a randomised algorithm A that takes as input a *secret* $s \in \mathcal{S}$ and returns n *shares* $\boldsymbol{s} \in \mathcal{S}^n$ s.t. the two following properties hold:

1. *(Decoding $(n-k)$ erasures.)* There is a deterministic algorithm that returns s when given *any* k distinct entries of $\boldsymbol{s}$.

2. *(Uniformity of $(k-1)$ tuples.)* The distribution of *any* $k-1$ distinct entries of $\boldsymbol{s}$ is uniform (over the randomness of A).

Note that in the first property, we do not require that the reconstruction algorithm be "efficient", merely that it exists. Also, since s is arbitrary and fixed in the definition, the second property directly implies that the distributions of the $(k-1)$ tuples are independent from it.

The definition may then be interpreted in the following way: if an adversary knows at least k shares of a sharing then it is *always* able to retrieve the secret s whereas if it knows only $k-1$ shares or fewer it does not learn any "information" about s, in the sense that the distribution of s conditioned on the knowledge of the at most $k-1$ shares is the same as its distribution without knowing any. This latter property is a direct consequence of the fact that since the distribution of those shares is known to be uniform, anyone can "simulate" it without knowing anything about s, and so observing them cannot reveal anything about the latter.

Example 6. A simple (n, n) scheme can be built over a finite group $(\mathbb{G}, +)$ in the following way: for $i \in [\![1, n-1]\!]$ sample the $\boldsymbol{s}_i$s uniformly at random and independently, and then define $\boldsymbol{s}_n$ as $s - \sum_{i=1}^{n-1} \boldsymbol{s}_i$. It is straightforward to see that s can be uniquely reconstructed from $\boldsymbol{s}$, and that the distribution of any $(n-1)$ tuple is uniform follows from the generalisation of Example 4 to $n-1$ input variables[†] and the application of Equation (2) to $\boldsymbol{s}_n$.

Although this first construction is quite elementary, it in fact already includes all the ingredients that we will later use in more advanced ones. Namely, it samples n uniform random variables with some prescribed dependency and uses one of those to mask the secret with a one-time pad, then given some subset of shares, either they exhibit some dependency and one of them is the one including the secret and so one can reconstruct it, or the shares are mutually independent (or do not depend on the secret) and then nothing is revealed.

2.1 Threshold secret sharing from polynomial interpolation

We now briefly step back from the linear-algebraic view to present an elegant ad hoc secret sharing scheme due to Shamir (1979) based on univariate polynomial interpolation in a finite field (we will shortly see that it is a special case of the general construction one directly obtains by applying Lemma 3 to the "threshold" constraint). It allows to build (k, n) schemes for secrets in a finite field $\mathbb{F}_q$ as long as $n \leq O(q)$. It works as follows: let $P \in \mathbb{F}_q[X]_{<k}$ be a polynomial of degree $\leq k-1$ whose $k-1$ a priori non-constant coefficients $\boldsymbol{c} \in \mathbb{F}_q^{k-1}$ are sampled uniformly and independently at random and whose constant coefficient is the secret s to be shared, *i.e.* $P = s + \sum_{i=1}^{k-1} \boldsymbol{c}_i X^i$. Then let $\boldsymbol{a} \in \mathbb{F}_q^n$

[†]Since we stated Lemma 3 for variables over fields, one would also need it to suitably generalise it over groups. This is however straightforward for binary matrices.

be a vector whose entries are all pairwise distinct and not equal to zero, and define the sharing $\boldsymbol{s}$ as $\text{eval}(P, \boldsymbol{a})$.[‡] We now prove that this defines a (k, n) secret sharing:

Proof. We show that the scheme satisfies the two properties of a (k, n) sharing.

1. Let $\mathcal{I} \subseteq [\![1, n]\!]$ be a subset of indices of size at least k. Using univariate interpolation,[§] $\hat{P} := \sum_{i\in\mathcal{I}} \boldsymbol{s}_i \prod_{j\in\mathcal{I}\setminus\{i\}} \frac{X-\boldsymbol{a}_j}{\boldsymbol{a}_i-\boldsymbol{a}_j}$ is the unique polynomial of degree $\leq k-1$ that evaluates to $\boldsymbol{s}_i$ on $\boldsymbol{a}_i$ for $i \in \mathcal{I}$. It is thus equal to P, and one recovers s from $\text{eval}(\hat{P}, 0)$.

2. Let $\mathcal{I} \subseteq [\![1, n]\!]$ be a subset of indices of size $k-1$ and denote by $\boldsymbol{s}_\mathcal{I}$ the vector of corresponding entries of $\boldsymbol{s}$. Then for fixed s and $\boldsymbol{a}$, the map $\boldsymbol{c} \mapsto \boldsymbol{s}_\mathcal{I}$ is bijective:

   → Given s and $\boldsymbol{c}$, $\boldsymbol{s}_\mathcal{I}$ is obtained from the evaluation of the fully-determined polynomial P on $\boldsymbol{a}_\mathcal{I}$.

   ← Since $0 \notin \boldsymbol{a}$, one knows the evaluation of P on k distinct points $\{0\} \cup \boldsymbol{a}_\mathcal{I}$, which allows to uniquely interpolate P and then obtain $\boldsymbol{c}$.

   Since this map is bijective and the distribution of $\boldsymbol{c}$ is uniform and independent from s, then by Lemma 2 so is the one of $\boldsymbol{s}_\mathcal{I}$.

□

2.2 Generalisation to arbitrary MDS linear codes

We will now see how *maximum distance separable* (MDS) codes naturally arise from the structure of (linear) threshold secret sharing, and thus give a generalisation of the above ad hoc construction. The observation here is that to obtain a (k, n) sharing one could use n random variables over $\mathbb{F}_q$ s.t. any k of them are statistically dependent while any $k-1$ of them are mutually independent, and then use those as "masks" to hide the secret s in a one-time-pad way s.t. s can be deduced from any dependence. If we ignore for a while this last point and using Lemma 3, one may build the desired random variables from a $k \times n$ matrix $\boldsymbol{M}$ s.t. any $k-1$ of its columns are linearly independent (or equivalently, any k of its columns span $\mathbb{F}_q^k$). This condition is then in fact equivalent to asking that $\boldsymbol{M}$ generates an MDS code.

Definition 7 (Linear code)**.** A *linear code* $\mathcal{C}$ over $\mathbb{F}_q$ of *length* n and *dimension* k (written $[n, k]_{\mathbb{F}_q}$) is a k-dimensional linear subspace of $\mathbb{F}_q^n$. Its *minimum distance* (or equivalently *minimum weight*) is defined as $\min_{\boldsymbol{x},\boldsymbol{y}\neq\boldsymbol{x}\in\mathcal{C}} wt(\boldsymbol{x}-\boldsymbol{y})$, where $wt(\cdot)$ is the "Hamming" weight function that counts how many entries of its input are non-zero.
Every linear code can be *generated* by a matrix $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ in the following sense: $\forall \boldsymbol{x} \in \mathcal{C}$, $\exists! \, \boldsymbol{m} \in \mathbb{F}_q^k$ s.t. $\boldsymbol{x} = \boldsymbol{m}\boldsymbol{G}$. Note that $\boldsymbol{G}$ is in general not unique.

Definition 8 (Maximum distance separable code)**.** An $[n, k]$ linear code is said to be *maximum distance separable* (MDS) iff. its minimum distance equals $n - k + 1$.

We now prove two fundamental results about MDS linear codes.

Theorem 9 (Existence)**.** *For any n, $k \leq n \in \mathbb{N}$, $\exists q$ s.t. an $[n, k]_{\mathbb{F}_q}$ MDS linear code exists.*

[‡]We do not consider $\boldsymbol{a}$ to be part of the sharing strictly speaking since it may be fixed and publicly known, yet its knowledge is essential to a successful recovery of the secret.

[§]We use here the "Lagrange" basis, but the process is easily generalisable.

Proof. We will prove this statement constructively by defining a family of MDS codes (*viz.* the "Reed-Solomon" codes).

Let $\mathbb{F}_q[X]_{<k}$ denote the k-dimensional vector space of univariate polynomials over $\mathbb{F}_q$ of degree $< k$, and $\boldsymbol{a} \in \mathbb{F}_q^n$ be a vector of $k \leq n \leq q$ pairwise-distinct elements of $\mathbb{F}_q$, then the map $\mathbb{F}_q^k \to \mathbb{F}_q^n$, $P \in \mathbb{F}_q[X]_{<k} \mapsto \mathrm{eval}(P, \boldsymbol{a})$ is linear. Since P has at most $k-1$ distinct roots, any input that is not the zero polynomial has an image of weight at least $n-k+1$, and this map thus defines an $[n, k, n-k+1]$ MDS linear code. □

Theorem 10 (Optimality for erasure decoding)**.** *An $[n, k]$ linear code, is MDS iff. all its generator matrices are s.t. every k of their columns are linearly independent.*

Proof. $\Rightarrow$ Let $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ be a generator matrix of an $[n, k]$ MDS code, and suppose that there is a subset $\mathcal{I} \subseteq [\![1, n]\!]$ of k indices such that the columns indexed by $\mathcal{I}$ are not linearly independent. Equivalently this means that the submatrix $\boldsymbol{G}_\mathcal{I}$ has rank $< k$; thus its left kernel has dimension ≥ 1 and $\exists\, \boldsymbol{m} \neq \boldsymbol{0} \in \mathbb{F}_q^k$ s.t. $\boldsymbol{m}\boldsymbol{G}_\mathcal{I} = \boldsymbol{0}$, and thusly a non-zero codeword $\boldsymbol{m}\boldsymbol{G}$ of weight at most $n-k < n-k+1$, which is a contradiction with the fact that the code generated by $\boldsymbol{G}$ is MDS.

$\Leftarrow$ Let $\boldsymbol{G} \in \mathbb{F}_q^{k \times n}$ be as required. Then given $\boldsymbol{x} := \boldsymbol{m}\boldsymbol{G}$, one can recover $\boldsymbol{m}$ from the restriction $\boldsymbol{x}_\mathcal{I}$ of $\boldsymbol{x}$ to any $\mathcal{I} \subseteq [\![1, n]\!]$ of size k by inverting the full-rank submatrix $\boldsymbol{G}_\mathcal{I}$. Consequently, the encodings of two distinct messages cannot be equal in more than $k-1$ positions and the minimum distance of the code generated by $\boldsymbol{G}$ is thus at least (in fact exactly) $n-k+1$. □

This last property is exactly the one we were asking for in the above. It now only remains to specify how to use the masks with carefully-crafted dependence to build a full threshold scheme; this only requires a slight additional control over their dependency.

We now show how to define a (k, n) secret sharing scheme in $\mathbb{F}_q$ from an $[n+1, k]$ MDS linear code over the same field.

Let $\boldsymbol{G}$ be a publicly-known generator matrix for an $[n+1, k]$ MDS linear code over $\mathbb{F}_q$, and reduce it in its first row to obtain the block matrix $\begin{pmatrix} \boldsymbol{e}_1 & \boldsymbol{G}' \end{pmatrix}$, where $\boldsymbol{e}_1 = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^t$ is the first vector of the canonical basis and (by Theorem 10) $\boldsymbol{G}' \in \mathbb{F}_q^{k \times n}$ generates an $[n, k]$ MDS code. Then a sharing of s is obtained by sampling $\boldsymbol{c}$ uniformly at random in $\mathbb{F}_q^{k-1}$, setting $\boldsymbol{m} := \begin{pmatrix} s & \boldsymbol{c} \end{pmatrix}$, and computing $\boldsymbol{s}$ as $\boldsymbol{m}\boldsymbol{G}'$ (or equivalently $\boldsymbol{s}$ is formed from the last n coordinates of $\boldsymbol{m}\boldsymbol{G}$ *i.e.* it is a codeword of the code obtained from the one generated by $\boldsymbol{G}$ after *puncturing* it in its first position).

The fact that this defines a (k, n) scheme essentially follows from the structure of the shares as $s + Y_i$ where the Y_i's are k-dependent and $(k-1)$-indpendent random variables. We also provide a direct proof below.

Proof. We proceed as for the interpolation-based scheme.

1. Immediate from Theorem 10 and the fact that $\boldsymbol{G}'$ is MDS.
2. Let $\mathcal{I} \subseteq [\![1, n]\!]$ be a subset of indices of size $k-1$, then for a fixed s and $\boldsymbol{G}'$, the map $\boldsymbol{c} \mapsto \boldsymbol{s}_\mathcal{I}$ is bijective:
   - $\rightarrow$ Given s and $\boldsymbol{c}$, one can easily compute $\boldsymbol{m}\boldsymbol{G}'_\mathcal{I} = \boldsymbol{s}_\mathcal{I}$.
   - $\leftarrow$ By construction and the MDS property of the codes generated by $\boldsymbol{G}$ and $\boldsymbol{G}'$, $\boldsymbol{G}'_\mathcal{I}$ is of rank $k-1$ and $\boldsymbol{e}_1$ is not in its span, hence $\boldsymbol{R} := \begin{pmatrix} \boldsymbol{e}_1 & \boldsymbol{G}'_\mathcal{I} \end{pmatrix}$ is invertible, and $\boldsymbol{c}$ can be computed from $\begin{pmatrix} s & \boldsymbol{s}_\mathcal{I} \end{pmatrix} \cdot \boldsymbol{R}^{-1} = \boldsymbol{m}$.

   Since this map is bijective and the distribution of $\boldsymbol{c}$ is uniform and independent from s, then by Lemma 2 so is the one of $\boldsymbol{s}_\mathcal{I}$.

□

Finally, let us remark that there is nothing special about the choice of $\boldsymbol{e}_1$ in the above construction of $\boldsymbol{G}'$ from $\boldsymbol{G}$, and that any non-null vector would be similarly suitable to embed the secret as the corresponding linear combination of elements of $\boldsymbol{m}$. For instance one can check that using instead the all-1 vector and setting $\boldsymbol{m}$ to the (k, k) additive sharing from Example 6 also results in a (k, n) sharing.

3 Access structures and minimal codewords

A natural generalisation of secret sharing as described above given by Massey (1993) is to make it possible to recover the secret from subsets of shares of different sizes. More generally, we would ideally be able to have a fine-grained control over which specific subsets allow to do so. For instance we may want a scheme s.t. $\boldsymbol{s}_{\{1,2\}}$ and $\boldsymbol{s}_{\{2,3,4\}}$ allow to recover the secret $\boldsymbol{s}_1$, with the constraint that only sets with one of $\{1, 2\}$ or $\{2, 3, 4\}$ as a subset allow to do so. We will see how to relate such a requirement to the structure of a linear code used to perform the sharing; however to achieve such an imbalance between the different shares' roles the codes will not be MDS any more.

In all of the following we will drop the k parameter used previously in the definition of (k, n) secret sharing schemes, since the knowledge of which subsets of shares determine the secret will be *a priori* independent from the size of the subset. We replace this with the notion of *access structure* defined below.

3.1 Access structures

We start with the following:

Definition 11 (Access structure)**.** The *access structure* for a secret-sharing scheme with n shares over a general set $\mathcal{S}$ is a subset $\mathcal{A}$ of $\wp(\mathcal{S})$ s.t. for all sharings $\boldsymbol{s} \in \mathcal{S}^n$ of an arbitrary secret s one has that:

1. There is a deterministic algorithm that returns s on input of any $\mathcal{X} \in \mathcal{A}$.
2. The distribution of the elements of any $\mathcal{X} \in \wp(\mathcal{S})$ for which $\nexists\, \mathcal{X}' \in \mathcal{A}$ s.t. $\mathcal{X}' \subseteq \mathcal{X}$ is uniform over the randomness used to compute the sharing.

In words, this means that the access structure contains all the "minimal" subsets of shares that allow to recover the secret: any of these is enough to do so, and any subset that does not include at least one of them is statistically independent from the secret (and thence does not allows to learn anything about it). It is however not obvious that this definition is not contradictory, *i.e.* that a secret sharing scheme does possess an access structure. In the following we will show that this is always the case for schemes built over linear codes.

3.2 Dual characterisation

We first show how to use the *dual* of the code used to build the secret sharing to determine if a set of shares allows to recover the secret, and address the minimality condition next. We consider as in the previous section (and w.l.o.g.) a sharing obtained as the n last coordinates of $\begin{pmatrix} s & \boldsymbol{c} \end{pmatrix} \boldsymbol{G}$ for a uniform $\boldsymbol{c}$ and $\boldsymbol{G}$ a generator matrix of an $[n + 1, k]$ linear code $\mathcal{C}$ whose first column is equal to $\boldsymbol{e}_1$.

Let us give the following:

Definition 12 (Parity-check matrix)**.** Let $\mathcal{C}$ be an $[n, k]_{\mathbb{F}_q}$. A *parity-check matrix* for $\mathcal{C}$ is any full-rank matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ s.t. for any generator matrix $\boldsymbol{G} \in \mathbb{F}_q^{k\times n}$ of $\mathcal{C}$ one has $\boldsymbol{G}\boldsymbol{H}^t = \boldsymbol{0}^{k\times(n-k)}$.

Definition 13 (Dual code)**.** The *dual* $C^\perp$ of an $[n, k]$ linear code $\mathcal{C}$ is the $[n, n-k]$ code s.t. $\forall\, \boldsymbol{x} \in \mathcal{C},\ \forall\, \boldsymbol{y} \in \mathcal{C}^\perp,\ \boldsymbol{x} \cdot \boldsymbol{y} = 0$. Equivalently, it is the code generated by a parity-check matrix for $\mathcal{C}$.

And then:

Lemma 14. *Consider an n-share secret sharing scheme defined from a linear code $\mathcal{C}$ as above; a subset of shares of indices in $\mathcal{X}$ allows to recover the secret iff. $\exists\, \boldsymbol{x} \in \mathcal{C}^\perp$ of support $\{1\} \cup \mathcal{X}'$, $\mathcal{X}' \subseteq \mathcal{X}$. Furthermore, in the negative case the subset of shares is statistically independent from the secret.*

Proof. $\Rightarrow$ Let $\boldsymbol{x} \in \mathcal{C}^\perp$ be as required. By definition of the dual code this means that for every sharing $\boldsymbol{s}$ of s, since $\begin{pmatrix} s & \boldsymbol{s} \end{pmatrix} \in \mathcal{C}$ one has $\begin{pmatrix} s & \boldsymbol{s} \end{pmatrix} \cdot \boldsymbol{x} = 0$. Let us write $\boldsymbol{x}$ as $\begin{pmatrix} \xi & \boldsymbol{x}' \end{pmatrix}$ where $\xi \neq 0$, then we have $s\xi = -\boldsymbol{s} \cdot \boldsymbol{x}'$ which means that s can be uniquely recovered as an appropriate linear combination of the known shares $\mathcal{X}$.

$\Leftarrow$ By the assumption that there is no $\boldsymbol{x}$ satisfying the requirement we have that the first column $\boldsymbol{G}_1$ of $\boldsymbol{G}$ is not in the column space of the submatrix $\boldsymbol{G}_\mathcal{X}$ with columns in $\mathcal{X}$. Thus by Lemma 3 the observed shares $\begin{pmatrix} s & \boldsymbol{s} \end{pmatrix} \boldsymbol{G}_\mathcal{X}$ are statistically independent from $s = \begin{pmatrix} s & \boldsymbol{s} \end{pmatrix} \boldsymbol{G}_1$.[¶]

□

3.3 Minimal codewords

The previous lemma can be used to determine if a subset of shares allows to recover the secret or if it is statistically independent from it; to further determine if it is an access structure (and to also justify the well-foundedness of this definition) we will need the additional concept of *minimal* codewords. In all of the following we use the notation $\boldsymbol{x} \preccurlyeq \boldsymbol{y}$ (resp. $\boldsymbol{x} \prec \boldsymbol{y}$) to mean that $\mathrm{supp}(\boldsymbol{x}) \subseteq \mathrm{supp}(\boldsymbol{y})$ (resp. $\mathrm{supp}(\boldsymbol{x}) \subset \mathrm{supp}(\boldsymbol{y})$).

Definition 15 (Minimal codeword)**.** A *minimal codeword* of a linear code $\mathcal{C}$ is a codeword $\boldsymbol{x}$ with its first non-zero coordinate equal to 1 s.t.: 1) there is no distinct $\boldsymbol{x}' \in \mathcal{C}$ with its first non-zero coordinate equal to 1 and $\boldsymbol{x}' \preccurlyeq \boldsymbol{x}$; 2) or equivalently there is no non-zero $\boldsymbol{x}' \in \mathcal{C}$ s.t. $\boldsymbol{x}' \prec \boldsymbol{x}$.

Proof. $\neg(1) \Rightarrow \neg(2)$ $\exists\, \lambda$ s.t. $(\boldsymbol{x} - \lambda \boldsymbol{x}') \prec \boldsymbol{x}$.
$\neg(2) \Rightarrow \neg(1)$ $\exists\, \lambda$ s.t. the first coordinate of $\lambda \boldsymbol{x}'$ equals 1. □

The first characterisation implies in particular that no two distinct minimal codewords may have the same support.

Massey then proves the following:

Lemma 16. *Every codeword $\boldsymbol{x}$ of $\mathcal{C}$ is a linear combination of minimal codewords $\{\boldsymbol{x}_i\}$ s.t. for all i one has $\boldsymbol{x}_i \preccurlyeq \boldsymbol{x}$.*

Proof. Let $\boldsymbol{x} \in \mathcal{C}$ be a non-zero codeword. If it is minimal (up to scaling by a scalar) then we are done; otherwise there must exist a non-zero codeword $\boldsymbol{x}_1 \prec \boldsymbol{x}$ which by induction may be assumed to be minimal. Thus $\exists\, \lambda$ s.t. $(\boldsymbol{x} - \lambda \boldsymbol{x}_1) \prec \boldsymbol{x}$ and the first part of the result follows by induction; the second part immediately follows from the base case and the two inclusions $\boldsymbol{x}_1 \prec \boldsymbol{x}$ and $(\boldsymbol{x} - \lambda \boldsymbol{x}_1) \prec \boldsymbol{x}$. □

This leads to the following:

[¶] Our statement of Lemma 3 would only allow us to apply it in the case where s is a uniform random variable. However it is easy to extend it for arbitrary distributions in the present case, as long as we only require the independence of the shares from s.

Corollary 17. *For every codeword $\boldsymbol{x} \in \mathcal{C}$ that is not minimal (up to rescaling), there is a minimal codeword $\boldsymbol{x}' \prec \boldsymbol{x}$ whose first non-zero coordinate is at the same index as the first non-zero coordinate of $\boldsymbol{x}$.*

This finally leads to the main result we want to show:

Theorem 18. *The access structure of a secret sharing scheme built from a code $\mathcal{C}$ is the set of the supports of the minimal codewords of $\mathcal{C}^\perp$ whose first coordinate is non-zero (with 1 then excluded from these sets).*

Proof. The fact that subsets of shares of this form allow to reconstruct the secret is a consequence of Lemma 14; it remains to show that no subset that does not include one of these allows to do so.

Again from Lemma 14 we have that if $\mathcal{X}$ allows to reconstruct the secret then there must exist $\boldsymbol{x} \in \mathcal{C}^\perp$ with support included in $\{1\} \cup \mathcal{X}$ and whose first coordinate is non-zero. Now either $\boldsymbol{x}$ is minimal (up to rescaling) and the previous inclusion is an equality and we are done, or by Corollary 17 there is a minimal codeword $\boldsymbol{x}' \prec \boldsymbol{x}$ whose first coordinate is a 1; from Lemma 14 $\mathrm{supp}(\boldsymbol{x}') \subseteq \mathcal{X}$ still allows to reconstruct the secret and we can conclude. □

This result is useful in that it justifies the fine-grained notion of access structure for linear secret sharing schemes and gives a conceptually simple characterisation thereof. In turn one may hope to use this characterisation to build secret sharing schemes with specific access structures as soon as this one can be mirrored by the minimal codewords of some linear code.

4 Perfectly-secure message transmission

We now briefly introduce *perfectly-secure message transmission* (or *PSMT*) as a generalisation of secret sharing as presented above. The generalisation is in two ways: 1) we may now consider interactive protocols where several participants may exchange information in several "rounds" of communication; 2) the adversary (somewhat implicit in the above secret-sharing scenario) is now *active*, so that it may tamper with exchanged messages.

In short the goal of PSMT is for two (or more) participants to exchange a secret in a way such that the adversary does not learn anything about it. We again wish to do this w.r.t. a very strong security objective, *viz.* that whatever the adversary may learn is statistically independent from the secret. To achieve this goal, the participants have access to a certain number of n channels with the guarantee (or in fact, the assumption) that the adversary controls at most t of them.

Remark 19. If in the above scenario the adversary is further assumed to be passive, the problem is immediately solved in all the relevant cases $t < n$ in only one round of communication from the existence of (n, n) secret sharing.

This remark justifies the fact that we only need to further develop techniques against active adversaries.

The goal of this section is to present an elegant scheme from Spini and Zémor (2016) for two participants $\mathfrak{A}$ and $\mathfrak{B}$, in its simple and unoptimised form; it is a two-round protocol in the setting $n = 2t + 1$, which for that number of rounds is minimal (Dolev *et al.*, 1993).

A first simple observation is that since in the chosen setting $n > 2t$, reliable communication between $\mathfrak{A}$ and $\mathfrak{B}$ is trivial to achieve as it is enough to broadcast any message over the n channels and decode with a majority vote. It is however equally obvious that this method does not provide any confidentiality since the adversary may read any message from one of the channels it controls.

The idea is then for $\mathfrak{B}$ to transmit a number of independent codewords from an $[n, t+1, t+1]_{\mathbb{F}_q}$ MDS code, where each coordinate $i \in [\![1, n]\!]$ is sent over the corresponding channel. Note that from the previous sections these codewords all naturally induce $(t+1, n)$ secret-sharing, and $\mathfrak{A}$ may pick one codeword to recover the secret and use the latter as a one-time-pad to reliably *and* confidentially broadcast a message. However because of adversarial tampering and since $t > (d-1)/2 = t/2$, $\mathfrak{A}$ will in general not be able to recover the original codeword and the derived secret it uses may thus be "noisy". The crux of the method is then to provide $\mathfrak{B}$ with sufficient information so that it may itself remove the noise from the secret and finally recover the message, without revealing anything to the adversary.

We now describe the techniques used by Spini and Zémor to resolve this last point.

4.1 Decoding from a syndrome-spanning subset

We first recall the following:

Definition 20 (Syndrome of a vector)**.** Let $\mathcal{C}$ be an $[n, k]_{\mathbb{F}_q}$ and $\boldsymbol{H}$ one of its parity-check matrices. The *syndrome* (usually written $\boldsymbol{s}$) of a vector $\boldsymbol{x} \in \mathbb{F}_q^n$ w.r.t. $\boldsymbol{H}$ is defined as $\boldsymbol{H}\boldsymbol{x}^t$.

Note that from the definition of a parity-check matrix, $\boldsymbol{H}\boldsymbol{x}^t = \boldsymbol{0}^{n-k}$ iff. $\boldsymbol{x} \in \mathcal{C}$. In particular if one has $\boldsymbol{y} := \boldsymbol{x} + \boldsymbol{e}$ for some $\boldsymbol{e} \in \mathbb{F}_q^n$ and some $\boldsymbol{x} \in \mathcal{C}$, then $\boldsymbol{H}\boldsymbol{y}^t = \boldsymbol{H}\boldsymbol{e}^t$.

We now wish to show that $\mathfrak{B}$ is able to recover an error $\boldsymbol{e}_1$ from its syndrome $\boldsymbol{H}\boldsymbol{e}_1^t$ as long as the former belongs to a suitable subspace for which $\mathfrak{B}$ possesses a basis (or "spanning subset") and its image through the syndrome map. We will then show that this can be ensured in the above protocol, which will allow us to conclude.

We start with:

Lemma 21. *Let $\mathcal{C}$ be an $[n, k, d]_{\mathbb{F}_q}$ linear code for which $\boldsymbol{H}$ is a parity-check matrix. Let $\mathcal{E}$ be a linear subspace of $\mathbb{F}_q^n$ whose elements all have weight $d-1$ or less. Then the map $\sigma_{\mathcal{E}} : \mathcal{E} \to \mathbb{F}_q^{n-k}$, $\boldsymbol{e} \mapsto \boldsymbol{H}\boldsymbol{e}^t$ is injective.*

Proof. By definition the (right) kernel of $\boldsymbol{H}$ is $\mathcal{C}$. From $\mathcal{C} \cap \mathcal{E} = \{\boldsymbol{0}^n\}$ the kernel of $\sigma_{\mathcal{E}}$ is trivial and the map is thence injective. □

Now since $\sigma_{\mathcal{E}}$ is injective and linear it must admit an efficiently-computable inverse $\sigma_{\mathcal{E}}^{-1} : \mathrm{Im}(\sigma_{\mathcal{E}}) \subset \mathbb{F}_q^{n-k} \to \mathcal{E}$ which is also linear. We formalise this as:

Proposition 22. *Let $\mathcal{C}$, $\boldsymbol{H}$, $\mathcal{E}$, $\sigma_{\mathcal{E}}$ be as above and $t := \dim(\mathcal{E})$. Let $\boldsymbol{B}_{\mathcal{E}} := \begin{pmatrix} \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_t \end{pmatrix}$ be a basis for $\mathcal{E}$ (where the blocks are here row-wise; note also that in this case $\boldsymbol{e}_i$ may not necessarily be equal to the i^{th} vector of the canonical basis of the ambient space) and $\boldsymbol{B}_{\mathcal{S}} := \begin{pmatrix} \boldsymbol{s}_1 & \cdots & \boldsymbol{s}_t \end{pmatrix} = \boldsymbol{H}\boldsymbol{B}_{\mathcal{E}}^t$. Then one may efficiently compute $\sigma_{\mathcal{E}}$ and $\sigma_{\mathcal{E}}^{-1}$ from the knowledge of $\boldsymbol{B}_{\mathcal{E}}$ and $\boldsymbol{B}_{\mathcal{S}}$.*

Proof. This follows immediately from the definitions and by linearity:

— $\sigma_{\mathcal{E}}(\boldsymbol{e} := \sum_{i=1}^t \lambda_i \boldsymbol{e}_i) = \lambda_i \sum_{i=1}^t \sigma_{\mathcal{E}}(\boldsymbol{e}_i) = \lambda_i \sum_{i=1}^t \boldsymbol{s}_i$.

— $\sigma_{\mathcal{E}}^{-1}(\boldsymbol{s} := \sum_{i=1}^t \lambda_i \boldsymbol{s}_i) = \lambda_i \sum_{i=1}^t \sigma_{\mathcal{E}}^{-1}(\boldsymbol{s}_i) = \lambda_i \sum_{i=1}^t \boldsymbol{e}_i$.

□

4.2 A perfectly-secure message transmission protocol

We are now ready to fully describe and analyse the two-round PSMT protocol from Spini and Zémor, in its simple form.

Using all notations as above, one does the following:

0. In order to transmit one message defined over $\mathbb{F}_q$, $\mathfrak{A}$, $\mathfrak{B}$ and the adversary agree on an $[n, t+1, t+1]_{\mathbb{F}_q}$ code $\mathcal{C}$ and on a parity-check matrix $\boldsymbol{H}$. They also equip $\mathcal{C}$ with a $(t+1, n)$ secret-sharing scheme; from the previous sections and w.l.o.g. we assume that the secret is recovered from a codeword $\boldsymbol{x}$ as $\boldsymbol{x} \cdot \boldsymbol{h}$ where $\boldsymbol{h} \in \mathbb{F}_q^n$ is a suitable fixed, publicly-known vector.

1. $\mathfrak{B}$ selects $t+1$ codewords $\{\boldsymbol{x}_i\}_{1 \leq i \leq t+1}$ independently and uniformly at random and sends them to $\mathfrak{A}$ using the n channels that they share together. Specifically, the i^{th} channel is used to transmit the i^{th} coordinate of all the $t+1$ codewords.

2. These codewords are received by $\mathfrak{A}$ as $\{\boldsymbol{y}_i := \boldsymbol{x}_i + \boldsymbol{e}_i\}_{1 \leq i \leq t+1}$, where the $\boldsymbol{e}_i$'s are error vectors of $\mathbb{F}_q^n$ of weight less than t introduced by the adversary. $\mathfrak{A}$ computes $\{\boldsymbol{s}_i := \boldsymbol{H}\boldsymbol{y}_i^t\}_{1 \leq i \leq t+1}$ and picks i s.t. $\boldsymbol{s}_i \in \langle \boldsymbol{s}_j \rangle_{1 \leq j \neq i \leq t+1}$;[‖] in the following we assume w.l.o.g. that $i = 1$. It then computes $s := \boldsymbol{y}_1 \cdot \boldsymbol{h}$, encrypts the message $m \in \mathbb{F}_q$ that it wants to send as $c := m + s$ and broadcasts it over the n channels. Finally it further broadcasts $\{\boldsymbol{s}_i\}_{1 \leq i \leq t+1}$ and $\{\boldsymbol{y}_i\}_{2 \leq i \leq t+1}$.

3. $\mathfrak{B}$ computes $\{\boldsymbol{e}_i\}_{2 \leq i \leq t+1}$ from $\{\boldsymbol{y}_i\}_{2 \leq i \leq t+1}$ and its knowledge of $\{\boldsymbol{x}_i\}_{2 \leq i \leq t+1}$. Since the errors all have weight $t < t+1$, knowing $\{\boldsymbol{e}_i\}_{2 \leq i \leq t+1}$ and $\{\boldsymbol{s}_i\}_{2 \leq i \leq t+1}$ is then enough to recover $\boldsymbol{e}_1$ from $\boldsymbol{s}_1 \in \langle \boldsymbol{s}_i \rangle_{2 \leq i \leq t+1}$ by Proposition 22. From then it can easily recompute s as $(\boldsymbol{x}_1 + \boldsymbol{e}_1) \cdot \boldsymbol{h}$ and decrypt c.

The correctness of the protocol follows from its description and the previous results, and we conclude with a short, somewhat informal security analysis. We only need to show that the distribution of s is uniform and independent from the data known to the adversary at the end of a run. This follows immediately from the remarks that: 1) the broadcast values $\{\boldsymbol{s}_i\}_{1 \leq i \leq t+1}$ are already known by the adversary since for all i one has $\boldsymbol{s}_i = \boldsymbol{H}\boldsymbol{y}_i^t = \boldsymbol{H}\boldsymbol{e}_i^t$ and the $\boldsymbol{e}_i$'s are error terms that it itself introduces; 2) the non-revealed noisy codeword $\boldsymbol{y}_1$ is independent from the revealed ones $\{\boldsymbol{y}\}_{2 \leq i \leq t+1}$; 3) $s = \boldsymbol{y}_1 \cdot \boldsymbol{h} = (\boldsymbol{x}_1 + \boldsymbol{e}_1) \cdot \boldsymbol{h} = (\boldsymbol{x}_1 \cdot \boldsymbol{h}) + (\boldsymbol{e}_1 \cdot \boldsymbol{h})$, where $(\boldsymbol{e}_1 \cdot \boldsymbol{h})$ is known by the adversary, and $\boldsymbol{x}_1 \cdot \boldsymbol{h}$ uniform and statistically independent from the (at most) t shares of the $(t+1, n)$ sharing that it controls. □

[‖] This is always possible since the errors span a subspace of dimension at most t.