A short introduction to secret sharing from linear codes

Pierre Karpman

December 9, 2020

This short lecture note gives a simple introduction to secret sharing and shows how Shamir's scheme can easily be adapted to any MDS linear code. It then presents the code-based generalisation of Massey which introduces the important concept of *access structure* and *minimal codewords*. It concludes with a short presentation of a further generalisation, *viz.* perfectly-secure message transmission protocols.

1 Preliminaries: uniform masking in a finite group

We first recall a few essential results about the distribution of the sum of two random distributions.

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.

2 Basic secret sharing

Definition 1. We define a secret-sharing scheme in the following, slightly informal way:

A (k, n) secret-sharing scheme over a general set $\mathcal{S}$ is a randomised algorithm A that takes as input $s \in \mathcal{S}$ and returns $\boldsymbol{s} \in \mathcal{S}^n$ 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).

Example 2. A simple (n, n) scheme can be built over a finite group $(\mathbb{G}, +)$ as follows: for $i \in [\![1, n-1]\!]$ take the $\boldsymbol{s}_i$s to be uniform i.i.d. variables, 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 (2).

2.1 Shamir's scheme from polynomial interpolation

We now present an elegant secret sharing scheme due to Shamir (1979) based on univariate polynomial interpolation. 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 $P \in \mathbb{F}_q[X]$ be a polynomial of degree $\leq k-1$ whose $k-1$ non-constant coefficients $\boldsymbol{c} \in \mathbb{F}_q^{k-1}$ are sampled uniformly 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 $\boldsymbol{a} \in \mathbb{F}_q^n$ be a vector whose entries are all pairwise distinct and not equal to zero, and define the sharing $\boldsymbol{s}$ as $\mathrm{eval}(P, \boldsymbol{a})$.[*] 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. By Lagrange interpolation, $P' := \sum_{i \in \mathcal{I}} \boldsymbol{s}_i \prod_{j \in \mathcal{I} \setminus \{i\}} \frac{X - \boldsymbol{a}_j}{\boldsymbol{a}_i - \boldsymbol{a}_j}$ 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 $\mathrm{eval}(P', 0)$.

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:

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

   $\leftarrow$ 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 so is the one of $\boldsymbol{s}_{\mathcal{I}}$.

□

2.2 Generalisation to arbitrary MDS linear codes

A (not so) close inspection of Shamir's scheme reveals that it essentially relies on Reed-Solomon codes, which are specific instances of *maximum distance separable* (MDS) linear codes. We will now see how this scheme can be (somewhat) generalised and formulated entirely within the framework of such codes, using a purely linear-algebraic language. This is arguably the "correct" view of secret sharing, in particular since it later leads to natural and powerful generalisations such as sketched in the next section.

We first define linear and MDS codes and one of their characterisation that is essential to our purpose.

Definition 3 (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 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.

[*] 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.

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 4 (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 5 (Existence)**.** *For any n, $k \leq n \in \mathbb{N}$, $\exists\, q$ s.t. an $[n, k]_{\mathbb{F}_q}$ MDS linear code exists.*

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 6 (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) $n - k + 1$. □

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 generator matrix for such a code, 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 Thm. (6)) $\boldsymbol{G}' \in \mathbb{F}_q^{k \times n}$ 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 obtained after *puncturing* the first coordinate). We now prove that this defines a (k, n) secret sharing:

Proof. We proceed as previously.

1. Immediate from Thm. (6) 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:

   $\to$ 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 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 $(k-1,k)$ additive sharing from Ex. (2) 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 7 (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 8 (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 9. *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 the shares $\boldsymbol{s}_{\mathcal{X}}$ in which s appears write as $s + \boldsymbol{\lambda} \cdot \boldsymbol{c}$ for some $\boldsymbol{\lambda} \neq \boldsymbol{0}$. Let $t \leq \#\mathcal{X}$ be the number of such linearly-independent shares and $\boldsymbol{\Lambda} := \begin{pmatrix} \boldsymbol{\lambda}_{(1)} & \cdots & \boldsymbol{\lambda}_{(t)} \end{pmatrix}$ be the corresponding full-rank matrix, then from the change of variable $\boldsymbol{c}'_i := \boldsymbol{\lambda}_{(i)} \cdot \boldsymbol{c}$ the same (linearly-independent) shares can be rewritten as $s + \boldsymbol{c}'_i$ and the uniformity of $\boldsymbol{c}$ (which carries over to $\boldsymbol{c}'$ from the bijectivity of the change of variable) allows to conclude that they all are uniformly-distributed random variables independent from s.

□

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 $\text{supp}(\boldsymbol{x}) \subseteq \text{supp}(\boldsymbol{y})$ (resp. $\text{supp}(\boldsymbol{x}) \subset \text{supp}(\boldsymbol{y})$).

Definition 10 (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. (1) $\Rightarrow$ (2) $\exists\, \lambda$ s.t. $(\boldsymbol{x} - \lambda \boldsymbol{x}') \prec \boldsymbol{x}$.

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

Using a similar argument one shows that no two distinct minimal codewords may have the same support.

Massey then proves the following:

Lemma 11. *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:

Corollary 12. *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 13. *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 Lem. (9); it remains to show that no subset that does not include one of these allows to do so.

Again from Lem. (9) 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 Cor. (12) there is a minimal codeword $\boldsymbol{x}' \prec \boldsymbol{x}$ whose first coordinate is a 1; from Lem. (9) $\text{supp}(\boldsymbol{x}') \subseteq \mathcal{X}$ 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 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 14. 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 15 (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 16 (Syndrome of a vector)**.** Let $\mathcal{C}$ be an $[n, k]_{\mathbb{F}_q}$ and $\boldsymbol{H}$ one of its parity-check matrix. 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 17. *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 18. *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 $\{\boldsymbol{x}_i\}_{1 \le i \le t+1}$ independently and uniformly at random and sends them to $\mathfrak{A}$ using the n channels that they share together. Specifically, the ith channel is used to transmit the ith coordinate of all the $t+1$ codewords.

2. These codewords are received by $\mathfrak{A}$ as $\{\boldsymbol{y}_i := \boldsymbol{x}_i + \boldsymbol{e}_i\}_{1 \le i \le t+1}$, 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 $\{\boldsymbol{s}_i := \boldsymbol{H}\boldsymbol{y}_i^t\}_{1 \le i \le t+1}$ and picks i s.t. $\boldsymbol{s}_i \in \langle \boldsymbol{s}_j \rangle_{1 \le j \ne i \le t+1}$;[†] 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 $\{\boldsymbol{s}_i\}_{1 \le i \le t+1}$ and $\{\boldsymbol{y}_i\}_{2 \le i \le t+1}$.

3. $\mathfrak{B}$ computes $\{\boldsymbol{e}_i\}_{2 \le i \le t+1}$ from $\{\boldsymbol{y}_i\}_{2 \le i \le t+1}$ and its knowledge of $\{\boldsymbol{x}_i\}_{2 \le i \le t+1}$. Since the errors all have weight $t < t+1$, knowing $\{\boldsymbol{e}_i\}_{2 \le i \le t+1}$ and $\{\boldsymbol{s}_i\}_{2 \le i \le t+1}$ is then enough to recover $\boldsymbol{e}_1$ from $\boldsymbol{s}_1 \in \langle \boldsymbol{s}_i \rangle_{2 \le i \le t+1}$ by Prop. 18. 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 $\{\boldsymbol{s}_i\}_{1 \le i \le t+1}$ 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 $\{\boldsymbol{y}\}_{2 \le i \le t+1}$; 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. $\square$

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