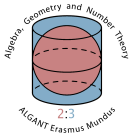

Université de Bordeaux 1

Stellenbosch University

Master Thesis

Schur products of linear codes: a study of parameters

Author:
Diego Mirandola

Supervisor:
Prof. Gilles Zémor

July 18, 2012

Contents

1 Introduction

In this Master Thesis we are interested in the study of the parameters of Schur product codes. Such a code is defined as the linear span of all componentwise products of all words of a given linear code. From the connection between linear codes and linear secret sharing schemes (LSSS), it turns out that good parameters of the product code yield a multiplication property of the associated LSSS that has applications to multi-party computation and verifiable secret sharing; for a reference about these applications, see [4]. Moreover, research is presently quite active on those topics.

First recall that codes arise as a solution to the problem of error correction, which can be easily explained by means of the following classical example. Bob asks Alice a question, and she wants to reply 1 (which means "Yes", whilst 0 means "No"), but the channel she is using to send her message does not reproduce symbols faithfully, i.e. there is some probability p that the 1 is transformed into 0; so she replies $(1,1,1,1,1,1,1,1,1)$. Now suppose that Bob receives the message $(1,1,0,0,1,1,0,1,0)$: as it contains more 1's than 0's, he interprets it correctly as a "Yes". So, what happened? Alice encoded her answer using the map "Yes"$\mapsto$ $(1,1,1,1,1,1,1,1,1)$, "No"$\mapsto (0,0,0,0,0,0,0,0,0)$, which associates to her answer an element of $C := \{(0,0,0,0,0,0,0,0,0),(1,1,1,1,1,1,1,1,1)\}$, which is a 1-dimensional sub-vector space of $\mathbb{F}_2^9$; then Bob noticed that the received message does not belong to C, interpreted it as the "closest element" (the element with the highest number of matching entries) of C, and decoded it as "Yes". In general, a q-ary linear code C of length n, dimension k and minimum distance d (or an $[n,k,d]_q$ code) is a k-dimensional sub-vector space of $\mathbb{F}_q^n$ such that any pair of elements of C differs in at least d entries (definitions 2.1.1

and 2.1.3); the elements of a code are called codewords. So, Alice used a code, precisely a repetition $[9, 1, 9]_2$ code. For a reference about coding theory see [8], [14], or any other coding theory handbook. For a more general overview of information theory see [3].

Secret sharing schemes model the following problem: we have a secret value s, and we want to distribute shares of s among n players, in a way such that, for some integers t and r, any team of t players cannot obtain any information about s (we say that the scheme achieves *t-privacy*) and any team of r players can completely recover s (we say that the scheme achieves *r-reconstruction*). The classical example is Shamir's scheme (example 2.3.4), where the secret is the value at 0 of a polynomial f of degree at most $k-1$ over a finite field and the i-th share is a point $(x_i, f(x_i))$ on the graph G of f. It is easy to see that any $k-1$ points of G give no information about $f(0)$, whilst any k points of G allow to recover f, hence $f(0)$. Therefore, Shamir's scheme achieves $(k-1)$-privacy and k-reconstruction. Now, it is easy to see that Shamir's scheme is linear: given two polynomials f_1 and f_2 of degree at most $k-1$ and the corresponding i-th shares $(x_i, f_1(x_i))$ and $(x_i, f_2(x_i))$, then $f_1 + f_2$ is again a polynomial of degree at most $k-1$ and the corresponding i-th share is the point $(x_i, (f_1+f_2)(x_i))$ on the graph of $f_1 + f_2$. The important consequence is that the i-th player can compute the i-th share of the sum of the secrets, without communicating with the other players. Similarly, given a secret f and a scalar α, the i-th player can compute the i-th share $(x_i, (\alpha f)(x_i))$ of the secret $(\alpha f)(0)$.

As shown in [1], we can canonically construct a secret sharing scheme $\Sigma(C)$ on n players from a given linear code C of length $n+1$ (proposition 2.5.1): we choose a codeword $c \in C$ whose 0-th coordinate is the secret s, and we define the i-th share to be the i-th coordinate of c; then $\Sigma(C)$ is automatically linear. Applying this construction to a Reed-Solomon code (examples 2.1.9 and 2.5.5) we obtain Shamir's scheme. Moreover, one can prove that good parameters of the code C guarantee good parameters of the associated LSSS; here the dual distance $d^\perp$ of C, i.e. the distance of the dual code (definition 2.1.5), is involved. We have that, given a code C of length $n+1$, minimum distance d and dual distance $d^\perp$, the associated LSSS $\Sigma(C)$ achieves $(d^\perp - 2)$-privacy and $(n - d + 2)$-reconstruction (theorem 2.5.4).

Interest in linear secret sharing schemes is motivated by their use in multi-party computation and verifiable secret sharing (see [4]). Multi-party computation is the problem of n players to compute an agreed function of their inputs, assuring the correctness of the output and the privacy of the inputs (even when some players cheat). A classical example is Yao's Millionaires' Problem (see [15]): two millionaires wish to know who is richer without having any additional information about each other's wealth. Verifiable secret sharing is the problem of the distribution of a secret value s among n players, where the dealer and/or some of the players may be cheating. However, as shown above for Shamir's scheme, linear secret shar-

ing schemes allow us to perform multi-party sums, but in general they do not automatically allow multi-party multiplications.

Let C be a code of length $n+1$, let $c = (s, c_1, \ldots, c_n)$ and $c' = (s', c'_1, \ldots, c'_n)$ be two codewords; from the secret sharing point of view, we may say that c_i is the i-th share of s and c'_i is the i-th share of s'. By linearity of C, $c + c' = (s + s', c_1 + c'_1, \ldots, c_n + c'_n)$ is a codeword, and $c_i + c'_i$ is the i-th share of the sum $s + s'$ of the secrets. In order to have the same property for the product ss' of the secrets, we need that the componentwise product $c * c' = (ss', c_1 c'_1, \ldots, c_n c'_n)$ is also a codeword. Unfortunately, this is not true in general, so we have to consider the Schur product $\widehat{C}$ of C, which is defined as the linear code spanned by all vectors of the form $c * c'$, for $c, c' \in C$. Hence we can canonically associate to $\widehat{C}$ an LSSS $\Sigma(\widehat{C})$, and consider its parameters: for example, if $\Sigma(\widehat{C})$ achieves r-reconstruction then any team of r players can reconstruct the secret product ss' only using their share products $c_i c'_i$.

We say that $\Sigma(C)$ has $\widehat{t}$-strong multiplication if it achieves $\widehat{t}$-privacy and $\Sigma(\widehat{C})$ achieves $(n - \widehat{t})$-reconstruction (sometimes we will refer to $\widehat{t}$ as to the multiplication parameter of $\Sigma(C)$). This models a secret sharing scenario in which two secrets s and s' are shared among n players and we assume that $\widehat{t}$ players are corrupted: we do not want these corrupted players to have enough information to recover one of the secrets, whilst the other $n - \widehat{t}$ players shall be able to compute the secret product ss' only using their share products. In particular, we obtain that if $\widehat{t} \leq d^{\perp} - 2$ and $\widehat{t} \leq \widehat{d} - 2$, for some integer $\widehat{t}$, then $\Sigma(C)$ has $\widehat{t}$-strong multiplication, where $d^{\perp}$ is the dual distance of C and $\widehat{d}$ is the minimum distance of $\widehat{C}$ (corollary 2.5.8).

All algebraic structures mentioned above (linear codes, product codes, LSSS's and LSSS's with multiplication) are properly defined and discussed in chapter 2, together with the link between them (canonical construction of an LSSS from a linear code, relations between the parameters).

In chapter 3 we discuss two recent results on this topic. Precisely, in section 3.1 we explain the construction given in [1] of a family of linear codes yielding a family of LSSS's having a good multiplication parameter. In section 3.2 we deal with the construction given in [10] of a family of linear codes with good parameters (an asymptotically good family, see definition 3.2.1) and whose product codes have good parameters as well. In both cases, first we give constructions of algebraic-geometric codes (for a reference, see [13]) with good parameters (as codes and as LSSS's) which work for large values of the field size q, and then using a field descent we construct codes over smaller fields, in order to let these results hold true for any choice of the field size q. The essential difference between the two constructions is in the choice of the field descent map: in the first construction its purpose is to preserve the multiplication parameter of the LSSS associated to the code, whilst in the second case its aim is to control the parameters of the code itself; we may say that the first construction is done from the secret sharing point of view, whilst in the second one the point

of view is more coding theoretic (actually, this is partially true: for an application, see [7]).

In chapters 4 and 5 we give our contribution to this topic. Precisely: given an $[n,k]_q$ code C, we are interested in bounding from below the dimension $\widehat{k}$ of the Schur product code $\widehat{C}$ of C as a function of n, k and $d^\perp$. The relation mentioned above between parameters of codes and parameters of associated LSSS's justifies this interest.

In chapter 4 we summarise some computational tests, performed using the computer algebra system PARI/GP, in which we computed the product dimension of some given codes. These tests gave us the feeling that this parameter quickly increases when code dimension and field size increase.

Finally, in chapter 5 we state and prove our main result.

Theorem 5.0.3. For all $\varepsilon > 0$, for all $t \in \mathbb{N}$, for all $[n,k]_q$ codes with dual distance $d^\perp \geq 2t+1$, we have

$$\widehat{k} \geq k + \left(\frac{1}{2} - \varepsilon\right) t \log_q^2(n-k) + o\left(\log_q^2(n-k)\right).$$

The asymptotic notation will be precisely defined in section 2.6. The proof of this theorem is split into three parts: first we arrange the generator matrix of the code in a way which leads us to easily find linearly independent vectors of the product code (section 5.1), then, using elementary algebraic tools, we bound from below the number of such vectors (section 5.2), and finally, using elementary analytic tools, we show that the bound found at the previous step agrees, at least for sufficiently large values of $n-k$, with the statement of the theorem (section 5.3). We also show (example 5.0.4) that this lower bound is asymptotically the best possible, at least in the case when $d^\perp = 3$.

2
Preliminaries

In this chapter we give some basic notions about linear codes and linear secret sharing schemes (LSSS) and we fix some notation which will be used in this work.

Section 2.1 deals with linear codes: here we give some basic definitions, properties and relations between their parameters; for a more exhaustive discussion we refer to [8], [14] or to any other coding theory handbook. In section 2.2 we define the Schur product of a linear code, as it is defined in [1] and [10], and we give some basic properties.

Sections 2.3 and 2.4 deal with linear secret sharing schemes, focusing on multiplication property; our main references are [1] and [2].

In section 2.5 we show how to construct an LSSS from a given code and we explain the relations between the parameters of the code and of the LSSS (dual distance and product distance on one side, privacy, reconstruction and multiplication on the other). This construction is taken from [1].

The last section is essentially taken from [12]; here we give some notation which will be useful later, in particular in chapter 5.

Through this chapter, and through the whole work, q will be a fixed prime power and $\mathbb{F}_q$ will denote the field with q elements.

2.1 Linear codes

We start by giving the classical basic definitions in linear coding theory.

Definition 2.1.1 (linear code)**.** A linear code C of length n and dimension k over $\mathbb{F}_q$ (an $[n,k]_q$ code) is a k-dimensional linear subspace of $\mathbb{F}_q^n$. The elements of a code are called codewords.

Definition 2.1.2 (generator matrix)**.** Let C be an $[n,k]_q$ code. A generator matrix G for C is a $k \times n$ matrix whose rows form an $\mathbb{F}_q$-basis of C.

By definition,

$$C = \{x^t G : x \in \mathbb{F}_q^k\};$$

equivalently, C is the image of the linear map

$$\begin{array}{ccc} \mathbb{F}_q^k & \to & \mathbb{F}_q^n. \\ x & \mapsto & G^t x \end{array}$$

With a basis change and a renumbering of the coordinates, we can write G in standard form, i.e. as

$$G = \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & G' \\ & & 1 & \end{array}\right)$$

with $G' \in M_{k,n-k}(\mathbb{F}_q)$; we will often assume that G is written in this way.

Let $c = (c_1, \ldots, c_n), c' = (c'_1, \ldots, c'_n) \in C$, we define their distance

$$d(c,c') := \left|\left\{i \in \{1,\ldots,n\} : c_i \neq c'_i\right\}\right| \in \{0,\ldots,n\},$$

i.e. $d(c,c')$ is the number of the coordinates in which c and c' are different; one can verify that d actually defines a metric on $\mathbb{F}_q^n$. Also, we define the weight $w(c)$ of c as its distance from the codeword $0 \in C$,

$$w(c) := d(c,0) \in \{0,\ldots,n\}.$$

In practice, codes are used to send messages through noisy channels. A message $m \in \mathbb{F}_q^k$ is transformed into a codeword $c := m^t G \in C$, which is sent through the channel. The received vector $x \in \mathbb{F}_q^n$ is not necessarily a codeword, but it will be decoded as a codeword c' which minimises the distance $d(x,c')$, hence transformed into the message $m' \in \mathbb{F}_q^k$ such that $c' = m'^t G$. Hopefully, $m = m'$. The distance between c and c' measures how much the noise must modify c in order to transform it in a vector x which is decoded as c', and cause a misinterpretation of the message. So we have another parameter which is relevant for a code.

Definition 2.1.3 (minimum distance)**.** Let C be an $[n,k]_q$ code. We call

$$d := \min_{\substack{c,c' \in C \\ c \neq c'}} d(c,c') = \min_{c \in C \setminus \{0\}} w(c)$$

the minimum distance of C. An $[n,k,d]_q$ code is an $[n,k]_q$ code with minimum distance d.

Note that the equality in the definition of d is true because C is linear.

Definition 2.1.4 (rate, relative minimum distance)**.** Let C be an $[n,k,d]_q$ code. We call

$$R := \frac{k}{n} \quad \text{and} \quad \delta := \frac{d}{n}$$

the rate and the relative minimum distance of C.

Definition 2.1.5 (dual code)**.** Let C be an $[n,k]_q$ code. The dual code $C^\perp$ of C is the $[n,n-k]_q$ code

$$C^\perp := \{x \in \mathbb{F}_q^n : x \cdot c = 0 \text{ for all } c \in C\}.$$

Here the dot denotes the scalar product on $\mathbb{F}_q^n$: if $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$ then $x \cdot y := x^t y = \sum_{i=1}^n x_i y_i$. Note that here $x \cdot x = 0$ does not imply $x = 0$ (for example, consider the case of $q = 2$ and $x = (1,1)$), hence some classical properties of orthogonal vector spaces over $\mathbb{R}$ do not hold for dual codes: for example, it is false that $C \cap C^\perp = \{0\}$ and it may also happen that $C = C^\perp$ (in this case we say that C is self-dual). Clearly, $C = (C^\perp)^\perp$.

If we write the generator matrix G of C in standard form, then a generator matrix for $C^\perp$ is

$$H := \left(\begin{array}{c|c} & 1 \\ -G'^\perp & \ddots \\ & 1 \end{array}\right) \in M_{n-k,n}(\mathbb{F}_q);$$

H is called a parity check matrix for C. Finally, we denote by $d^\perp$ the dual distance of C, i.e. the minimum distance of its dual code $C^\perp$; it is the cardinality of a minimal linearly dependent set of columns of G.

Clearly, a "good" code is a code which allows us to send long messages with a small waste of bandwidth and a big resistance to the noise in the channel. In other words, we are interested in $[n,k,d]_q$ codes with big k, big d and small n. We are going to give some classical bounds which relate these parameters.

Theorem 2.1.6 (Singleton bound)**.** *Let C be an $[n,k,d]_q$ code. Then*

$$k + d \leq n + 1.$$

Proof. Let G be the generator matrix of C, written in standard form; then its first row is a codeword $c \in C$ and it has weight $w(c) \leq 1 + n - k$. □

Definition 2.1.7 (MDS code)**.** Let C be an $[n,k,d]_q$ code. We say that C is an MDS (maximum distance separable) code if $k + d = n + 1$.

Theorem 2.1.8. *Let C be an $[n,k,d]_q$ code. C is MDS if and only if $C^\perp$ is MDS.*

Proof. Let C be MDS, i.e. $d = n + 1 - k$, and let $H \in M_{n-k,n}(\mathbb{F}_q)$ be a generator matrix for $C^\perp$. As $C = (C^\perp)^\perp$, $C^\perp$ has dual distance d, hence any $d-1 = n-k$ columns of H are linearly independent, hence any square submatrix of H has full rank, hence any non zero codeword of $C^\perp$ has less than $n-k$ zeros. This proves that $d^\perp > k$. On the other hand $d^\perp \leq k+1$ by the Singleton bound, hence $d^\perp = k+1$ and $C^\perp$ is MDS.

The opposite implication follows since $C = (C^\perp)^\perp$. □

Example 2.1.9 (Reed-Solomon code)**.** Let $k \leq n \leq q$, fix n distinct elements $x_1, \ldots, x_n \in \mathbb{F}_q$.

Let C be the image of the (injective) evaluation map

$$\begin{array}{ccc} \mathbb{F}_q[X]_{<k} & \to & \mathbb{F}_q^n, \\ f & \mapsto & (f(x_1), \ldots, f(x_n)) \end{array}$$

i.e. $C := \{(f(x_1), \ldots, f(x_n)) : f \in \mathbb{F}_q[X], \deg f < k\} \leq \mathbb{F}_q^n$. Then C is an $[n,k]_q$ code, called a Reed-Solomon code. Moreover, C has minimum distance $d = n + 1 - k$; indeed: let $c \in C$ be such that $w(c) \leq n-k$, i.e. c is the image of a polynomial $f \in \mathbb{F}_q[X]$ of degree $\deg f < k$ with $n - w(c) \geq k$ zeros, hence $f = 0$ and $c = 0$. This proves that $d \geq n+1-k$, hence $d = n+1-k$ by the Singleton bound. Therefore, C is an MDS code. Finally, a generator matrix for C is

$$G = \begin{pmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \\ \vdots & & \vdots \\ x_1^{k-1} & \ldots & x_n^{k-1} \end{pmatrix};$$

note that $\operatorname{rk} G = k$ since it is a Vandermonde matrix.

Theorem 2.1.10 (Plotkin bound)**.** *Let C be an $[n,k,d]_q$ code. Then*

$$d \leq \frac{q^{k-1}(q-1)}{q^k - 1} n.$$

Proof. We compute the quantity $\sum_{c,c' \in C} d(c,c')$ in two different ways. First clearly

$$\sum_{c,c' \in C} d(c,c') \geq \sum_{\substack{c,c' \in C \\ c \neq c'}} d = dq^k(q^k - 1).$$

On the other hand, consider the $q^k \times n$ matrix of all the codewords of C. Let $\alpha \in \mathbb{F}_q$, we want to compute the number m of appearances of α on a column of this matrix, say the first; let g be the first column of G, we have

$$m = |\{x \in \mathbb{F}_q^k : x^t g = \alpha\}| = q^{k-1}$$

since this set is a hyperplane of $\mathbb{F}_q^k$; in particular, m does not depend on α or on the chosen column. Thus, for any column we have m occurrences

of any element, and for any of these occurrences we have a contribution of $q^k - m$ to the sum of the distances, i.e.

$$\sum_{c,c' \in C} d(c,c') = \sum_{i=1}^{n} \sum_{j=1}^{q} m(q^k - m) = nqq^{k-1}q^{k-1}(q-1) = \frac{q-1}{q}(q^k)^2 n.$$

We obtain

$$dq^k(q^k - 1) \leq \frac{q-1}{q}(q^k)^2 n$$

which is what we were looking for. □

Theorem 2.1.11 (Hamming bound)**.** *Let C be an $[n,k,d]_q$ code, put $t := \lfloor \frac{d-1}{2} \rfloor$. Then*

$$n - k \geq \log_q \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$

Proof. For all $x \in \mathbb{F}_q^n, r \in \mathbb{R}_>$, we define the ball of center x and radius r as

$$B(x,r) := \{y \in \mathbb{F}_q^n : d(x,y) \leq r\};$$

clearly

$$|B(x,r)| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i.$$

Now the balls $B(c,t)$, for $c \in C$, are disjoint, hence

$$\left| \bigcup_{c \in C} B(c,t) \right| = \sum_{c \in C} |B(c,t)| = q^k \sum_{i=0}^{r} \binom{n}{i} (q-1)^i;$$

on the other hand,

$$\bigcup_{c \in C} B(c,t) \subseteq \mathbb{F}_q^n \Longrightarrow \left| \bigcup_{c \in C} B(c,t) \right| \leq q^n$$

and the conclusion follows. □

2.2 Schur product codes

We denote by $*$ the componentwise multiplication (or Schur product) on $\mathbb{F}_q^n$, i.e. for $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$ we put

$$x * y := (x_1 y_1, \ldots, x_n y_n) \in \mathbb{F}_q^n.$$

Definition 2.2.1 (Schur product code)**.** Let C be an $[n,k,d]_q$ code. We call $\widehat{C} := \langle c * c' : c, c' \in C \rangle \leq \mathbb{F}_q^n$ the (Schur) product code of C.

Clearly $\widehat{C}$ has the same length as C. Moreover, $\widehat{C}$ contains (a copy of) C, given by the injective map $c \mapsto c * c$, hence it has dimension $\widehat{k} \geq k$ and distance $\widehat{d} \leq d$. Often we will call $\widehat{k}$ and $\widehat{d}$ the product dimension and the product distance of C. Any $\mathbb{F}_q$-basis $\{g_1, \ldots, g_k\}$ of C gives a generator system $\{g_i * g_j : 1 \leq i \leq j \leq k\}$ of $\widehat{C}$ over $\mathbb{F}_q$, hence $\widehat{k} \leq \frac{k(k+1)}{2}$.

We will be interested in more precise estimates of $\widehat{k}$. In order to do that, we decompose $\widehat{C}$ as

$$\widehat{C} = \langle g_i * g_i : 1 \leq i \leq k \rangle \oplus \langle g_i * g_j : 1 \leq i < j \leq k \rangle$$

where $g_1, \ldots, g_k$ are the rows of a generator matrix of C written in standard form. The first summand is (a copy of) C, the second summand is a subspace of $\mathbb{F}_q^{n-k}$ of dimension at most $\frac{k(k-1)}{2}$.

The following lemma will often be useful for bounding from below the dimension of the second summand.

Lemma 2.2.2. *Let $v_0 \in (\mathbb{F}_q^*)^m$, $v_1, \ldots, v_h \in \mathbb{F}_q^m$. If $v_1, \ldots, v_h$ are linearly independent over $\mathbb{F}_q$ then $v_0 * v_1, \ldots, v_0 * v_h$ are linearly independent over $\mathbb{F}_q$.*

Proof. Denote by $v_{i,j}$ the j-th coordinate of the vector v_i.

Let $\alpha_1, \ldots, \alpha_h \in \mathbb{F}_q$ be such that

$$\sum_{i=1}^{h} \alpha_i v_0 * v_i = 0;$$

then, for all $j = 1, \ldots, m$,

$$\sum_{i=1}^{h} \alpha_i v_{0,j} v_{i,j} = 0 \Longrightarrow \sum_{i=1}^{h} \alpha_i v_{i,j} = 0,$$

as $v_{0,j} \neq 0$; therefore $\sum_{i=1}^{h} \alpha_i v_i = 0$ and since the v_i's are linearly independent this yields $\alpha_i = 0$ for all $i = 1, \ldots, h$. □

Corollary 2.2.3. *Let C be an MDS $[n, k]_q$ code. If $2k - 1 \leq n$ then $\widehat{k} \geq 2k - 1$.*

Proof. Let G be a generator matrix for C, written as

$$G = \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & G' \\ & & 1 & \end{array} \right),$$

where $G' \in M_{k,n-k}(\mathbb{F}_q^*)$ (by MDS hypothesis). The MDS hypothesis and $2k - 1 \leq n$ imply that $k \leq d$, which implies that any $k - 1$ rows of G' are linearly independent. Hence we can apply the previous lemma choosing as $v_0, \ldots, v_{k-1}$ any set of rows and conclude. □

Example 2.2.4. Let $k \leq n \leq q$, fix n distinct elements $x_1, \ldots, x_n \in \mathbb{F}_q$ and let C be the Reed-Solomon $[n,k]_q$ code generated by the matrix

$$G = \begin{pmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \\ \vdots & & \vdots \\ x_1^{k-1} & \ldots & x_n^{k-1} \end{pmatrix}.$$

Then $\widehat{C}$ is generated by

$$\widehat{G} = \begin{pmatrix} 1 & \ldots & 1 \\ x_1 & \ldots & x_n \\ \vdots & & \vdots \\ x_1^{2k-2} & \ldots & x_n^{2k-2} \end{pmatrix};$$

in particular, $\operatorname{rk} \widehat{G} = \min\{2k-1, n\}$. Hence, if $2k-1 \leq n$, $\widehat{C}$ is a Reed-Solomon $[n, 2k-1]_q$ code.

Example 2.2.5 (simplex code)**.** Let q be a prime power, $k \in \mathbb{N}$. Let G be the matrix whose columns are a set of representatives of $\mathbb{F}_q^k \setminus \{0\}$ modulo scalar multiplication (i.e. the points of the $(k-1)$-dimensional projective space over $\mathbb{F}_q$). Then G has rank k and $n := \frac{q^h-1}{q-1}$ columns; hence G is the generator matrix of an $[n,k]_q$ code C, called the simplex $[n,k]_q$ code. Note that, by construction, C has dual distance $d^\perp = 3$.

Let $\widehat{G}$ be the matrix whose rows are the vectors $g_i * g_j$ for $1 \leq i \leq j \leq k$, where $g_1, \ldots, g_k$ are the rows of G. For all $1 \leq i \leq j \leq k$, we find a column of G whose i-th and j-th entries are non-zero, and all other entries are equal to zero. The corresponding $\frac{k(k+1)}{2}$ columns of $\widehat{G}$ give a maximal rank submatrix. It follows that the product code $\widehat{C}$ has dimension $\widehat{k} = \frac{k(k+1)}{2}$.

2.3 Linear secret sharing schemes

Definition 2.3.1 (linear secret sharing scheme)**.** A linear secret sharing scheme (LSSS) is a tuple $\Sigma = (\mathbb{F}_q, n, e, v_0, V_1, \ldots, V_n)$ where:

- $\mathbb{F}_q$ is the field with q elements, where q is a prime power;
- $n, e \in \mathbb{N}$;
- $v_0 \in \mathbb{F}_q^e \setminus \{0\}$;
- $V_1, \ldots, V_n \leq \mathbb{F}_q^e$ such that $v_0 \in \sum_i V_i$.

We put $\mathcal{P} := \{1, \ldots, n\}$ (the set of players); for $A \subseteq \mathcal{P}$ we put $V_A := \sum_{i \in A} V_i$.

Definition 2.3.2 (adversary structure, access structure)**.** Let Σ be an LSSS.

a) The adversary structure $\mathcal{A}(\Sigma)$ is the set of all $A \subseteq \mathcal{P}$ satisfying the property: there exists an $\mathbb{F}_q$-linear map $\phi\colon \mathbb{F}_q^e \to \mathbb{F}_q$ such that $\phi(v_0) = 1$ and $\phi(V_A) = 0$; moreover, by convention, $\emptyset \in \mathcal{A}(\Sigma)$.

b) The access structure $\Gamma(\Sigma)$ is the set of all $B \subseteq \mathcal{P}$ such that $v_0 \in V_B$; by definition of LSSS, $\mathcal{P} \in \Gamma(\Sigma)$.

Clearly:

1. $A \in \mathcal{A}(\Sigma), A' \subseteq A \Longrightarrow A' \in \mathcal{A}(\Sigma)$;
2. $B \in \Gamma(\Sigma), B' \supseteq B \Longrightarrow B' \in \Gamma(\Sigma)$.

Moreover, $\mathcal{A}(\Sigma) \cap \Gamma(\Sigma) = \emptyset$ and $\mathcal{A}(\Sigma) \cup \Gamma(\Sigma) = 2^{\mathcal{P}}$ (the family of all subsets of $\mathcal{P}$).

Definition 2.3.3 (privacy, reconstruction)**.** Let Σ be an LSSS, $t, r \in \mathbb{N}$. We say that:

a) Σ achieves t-privacy if $\mathcal{A}(\Sigma) \supseteq \{A \subseteq \mathcal{P} : |A| = t\}$;

b) Σ achieves r-reconstruction if $\Gamma(\Sigma) \supseteq \{B \subseteq \mathcal{P} : |B| = r\}$.

$\mathcal{A}(\Sigma) \cap \Gamma(\Sigma) = \emptyset$ implies that $t < r$; we say that Σ is a threshold LSSS if it achieves t-privacy and $(t+1)$-reconstruction for some $t \in \mathbb{N}$. Note that the previous definition does not say anything about the maximality of t (respectively the minimality of r): for example if Σ achieves t-privacy then Σ achieves t'-privacy for any $t' < t$ and it may happen that Σ actually achieves $(t+1)$-privacy.

These definitions model the following idea. We want to share a secret $s \in \mathbb{F}_q$ between n players. We choose (uniformly random) an $\mathbb{F}_q$-linear map $\phi\colon \mathbb{F}_q^e \to \mathbb{F}_q$ such that $\phi(v_0) = s$; the i-th share (the private information given to the i-th player) is the image under ϕ of (a fixed basis of) V_i. Now, some players can decide to form a team, i.e. share their private information with their teammates and work together with the purpose of recovering the secret. Then $\mathcal{A}(\Sigma)$ is the family of all teams having no information about the secret and $\Gamma(\Sigma)$ is the family of all teams having enough information to recover the secret. Hence Σ achieves t-privacy if any team of t players has no information about the secret and r-reconstruction if any team of r players has enough information to recover the secret.

Example 2.3.4 (Shamir's LSSS)**.** Let $k \leq n < q$, fix $n+1$ distinct elements $x_0, \ldots, x_n \in \mathbb{F}_q$.

We want to share a secret $s \in \mathbb{F}_q$ among n players. We choose a polynomial $f \in \mathbb{F}_q[X]$ of degree $\deg f = k-1$ such that $f(x_0) = s$, and we define the i-th share to be $f(x_i)$. Then the knowledge of any k pairs $(x_i, f(x_i))$ allows to recover f, hence the secret $s = f(x_0)$, whilst the knowledge of any $k-1$ such pairs does not give any information about s.

This defines an LSSS Σ, namely a Shamir's LSSS. The parameters of Σ (notation as in definition 2.3.1) are $e = k$, $v_0 = (1, x_0, \ldots, x_0^{k-1})$and $V_i = \langle (1, x_i, \ldots, x_i^{k-1}) \rangle$ for all $i = 1, \ldots, n$. We have just said that Σ achieves $(k-1)$-privacy and k-reconstruction, hence it is a threshold scheme.

2.4 LSSS's with multiplication

We fix the following notation. Let $v = (v_1, \ldots, v_e), w = (w_1, \ldots, w_e) \in \mathbb{F}_q^e$, we put

$$v \otimes w := (v_1 w_1, \ldots, v_1 w_e, \ldots, v_e w_1, \ldots, v_e w_e) \in \mathbb{F}_q^{e^2}.$$

Let $V, W \leq \mathbb{F}_q^e$, we put $V \otimes W := \langle v \otimes w : v \in V, w \in W \rangle \leq \mathbb{F}_q^{e^2}$. In our case, for $A \subseteq \mathcal{P}$ we put $\widehat{V}_A := \sum_{i \in A} V_i \otimes V_i$. The following property links this product with the scalar product.

Lemma 2.4.1. *Let $v, v', w, w' \in \mathbb{F}_q^e$. Then*

$$(v \otimes w) \cdot (v' \otimes w') = (v \cdot v')(w \cdot w').$$

Proof. As usual, let $v = (v_1, \ldots, v_e)$ (and similarly the other vectors), then

$$\begin{aligned}(v \otimes w) \cdot (v' \otimes w') = \sum_{i,j=1}^{e} (v_i w_j)(v_i' w_j') = \left(\sum_{i=1}^{e} v_i v_i' \right) \left(\sum_{j=1}^{e} w_j w_j' \right) = \\ = (v \cdot v')(w \cdot w').\end{aligned}$$

□

Definition 2.4.2 (multiplication)**.** Let Σ be an LSSS. We say that Σ has multiplication if:

(i) Σ achieves 1-privacy,

(ii) $v_0 \otimes v_0 \in \widehat{V}_{\mathcal{P}}$.

Definition 2.4.3 ($\widehat{t}$-strong multiplication)**.** Let Σ be an LSSS, $\widehat{t} \in \mathbb{N}$. We say that Σ has $\widehat{t}$-strong multiplication if:

(i) Σ achieves $\widehat{t}$-privacy,

(ii) $v_0 \otimes v_0 \in \widehat{V}_{\mathcal{P}}$,

(iii) for all $B \subseteq \mathcal{P}$ with $|B| = n - \widehat{t}$, $v_0 \otimes v_0 \in \widehat{V}_B$.

Property (ii), common to the two definitions, says that

$$\widehat{\Sigma} := (\mathbb{F}_q, n, e^2, v_0 \otimes v_0, V_1 \otimes V_1, \ldots, V_n \otimes V_n)$$

is an LSSS. Property (iii) in the latter definition says that $\widehat{\Sigma}$ achieves $(n-\widehat{t})$-reconstruction. So, an LSSS Σ has $\widehat{t}$-multiplication if it achieves $\widehat{t}$-privacy and its product $\widehat{\Sigma}$ achieves $(n - \widehat{t})$-reconstruction.

Lemma 2.4.4. *Let* $v_0, \ldots, v_n \in \mathbb{F}_q^e$, $\lambda_1, \ldots, \lambda_n \in \mathbb{F}_q$. *We have that* $v_0 \otimes v_0 = \sum_{i=1}^n \lambda_i v_i \otimes v_i$ *if and only if*

$$\phi(v_0)\phi'(v_0) = \sum_{i=1}^n \lambda_i \phi(v_i)\phi'(v_i) \text{ for all } \mathbb{F}_q\text{-linear maps } \phi, \phi' \colon \mathbb{F}_q^e \to \mathbb{F}_q. \tag{2.4.1}$$

Proof. We have $v_0 \otimes v_0 = \sum_{i=1}^n \lambda_i v_i \otimes v_i$ if and only if

$$\left(\sum_{i=1}^n \lambda_i v_i \otimes v_i - v_0 \otimes v_0 \right) \cdot (b \otimes b') = 0 \text{ for all } b, b' \in \mathbb{F}_q^e,$$

i.e., by bilinearity of the scalar product,

$$\left(\sum_{i=1}^n \lambda_i v_i \otimes v_i \right) \cdot (b \otimes b') = (v_0 \otimes v_0) \cdot (b \otimes b') \text{ for all } b, b' \in \mathbb{F}_q^e. \tag{2.4.2}$$

Now by lemma 2.4.1 the left hand side is

$$\left(\sum_{i=1}^n \lambda_i v_i \otimes v_i \right) \cdot (b \otimes b') = \sum_{i=1}^n \lambda_i (v_i \otimes v_i) \cdot (b \otimes b') = \sum_{i=1}^n \lambda_i (v_i \cdot b)(v_i \cdot b')$$

and the right hand side is

$$(v_0 \otimes v_0) \cdot (b \otimes b') = (v_0 \cdot b)(v_0 \cdot b').$$

Hence (2.4.2) is equivalent to

$$\sum_{i=1}^n \lambda_i (v_i \cdot b)(v_i \cdot b') = (v_0 \cdot b)(v_0 \cdot b') \text{ for all } b, b' \in \mathbb{F}_q^e,$$

which is equivalent to (2.4.1) since we can identify $\mathbb{F}_q$-linear maps $\mathbb{F}_q^e \to \mathbb{F}_q$ and vectors in $\mathbb{F}_q^e$. □

Assume that we have two secrets $s, s' \in \mathbb{F}_q$ shared between the n players, i.e. two maps $\phi, \phi' \colon \mathbb{F}_q^e \to \mathbb{F}_q$ such that $\phi(v_0) = s, \phi'(v_0) = s'$. Assume that $V_i = \langle v_i \rangle$ for all $i \in \mathcal{P}$, i.e. that all V_i's have dimension 1 (this can be easily generalised to V_i's of arbitrary dimension, by generalising lemma 2.4.4), hence the i-th shares are simply $\phi(v_i)$ and $\phi'(v_i)$. By property (ii) and the previous lemma

$$ss' = \phi(v_0)\phi'(v_0) = \sum_{i=1}^n \lambda_i \phi(v_i)\phi'(v_i)$$

for some $\lambda_1, \ldots, \lambda_n \in \mathbb{F}_q$, i.e. the product of the secrets is a linear combination of the products of the shares. This says that in an LSSS with multiplication the i-th share of the product of two secrets is the product

of the i-th shares of the secrets; in particular, the i-th player can compute the i-th share of the secret product without communicating with the other players.

Property (iii) says that, for all sets $B \subseteq \mathcal{P}$ with $|B| = n - t$,

$$ss' = \phi(v_0)\phi'(v_0) = \sum_{i \in B} \lambda_i \phi(v_i)\phi'(v_i)$$

for some $\{\lambda_i\}_{i \in B} \subseteq \mathbb{F}_q$ (arguing as in the previous lemma), i.e. the secret product is a linear combination of any $n - t$ share products.

An LSSS with $\widehat{t}$-strong multiplication models the following idea. We have two secrets shared between the n players, and we suspect that $\widehat{t}$ players are corrupted. By $\widehat{t}$-privacy of Σ, the corrupted players have not enough information to recover the secrets. By $(n - \widehat{t})$-reconstruction of $\widehat{\Sigma}$, the other $n - \widehat{t}$ players can form a team and recover the product of the secrets by sharing an information which is not enough to recover one of the two secrets.

2.5 From codes to LSSS's

First we fix some notation. Let $A \subseteq \{0, \dots, n\}$, then

$$\begin{array}{rccc} \pi_A \colon & \mathbb{F}_q^{n+1} & \to & \mathbb{F}_q^{|A|} \\ & (x_i)_{i=0,\dots,n} & \mapsto & (x_i)_{i \in A} \end{array}$$

is the ($\mathbb{F}_q$-linear) projection on the coordinates $i \in A$; if $A = \{i\}$ for some i then we write π_i instead of $\pi_{\{i\}}$ and x_i instead of $\pi_i(x)$.

Let C be an $[n+1, k]_q$ code. Assume that $e_0 \notin C$ and $e_0 \notin C^\perp$, where e_i, for $i = 0, \dots, n$, denotes the i-th vector of the standard basis of $\mathbb{F}_q^{n+1}$. Then we have a standard way to construct an LSSS from C.

Proposition 2.5.1. *Let C be an $[n+1, k]_q$ code, with generator matrix G. If $e_0 \notin C$ and $e_0 \notin C^\perp$ then the tuple $\Sigma(C) = (\mathbb{F}_q, n, k, v_0, \langle v_1 \rangle, \dots, \langle v_n \rangle)$, where v_i, for $i = 0, \dots, n$, is the i-th column of G, is an LSSS.*

Proof. Clearly $v_0 \neq 0$ since $e_0 \notin C^\perp$.

We only have to prove that $v_0 \in \sum_i \langle v_i \rangle$. As $e_0 \notin C$, we can define the $\mathbb{F}_q$-linear map

$$\begin{array}{rccc} \rho_{\{1,\dots,n\}} \colon & \pi_{\{1,\dots,n\}}(C) & \to & \mathbb{F}_q \\ & \pi_{\{1,\dots,n\}}(c) = (c_1, \dots, c_n) & \mapsto & c_0 \end{array}$$

(otherwise, if there exists $c, c' \in C$ such that $c_i = c'_i$ for $i = 1, \dots, n$ but $c_0 \neq c'_0$ then by linearity $e_0 = \frac{c - c'}{c_0 - c'_0} \in C$). Hence, we can associate to $\rho_{\{1,\dots,n\}}$ a matrix $b = (b_1, \dots, b_n) \in \mathbb{F}_q^n$ such that, for all $c \in C$, $c_0 = \sum_{i=1}^n b_i c_i$. In particular, this implies that $v_0 = \sum_{i=1}^n b_i v_i$. □

Clearly, the same construction gives an LSSS $\Sigma(C, i)$ for any choice of a coordinate $i \in \{0, \dots, n\}$ such that $e_i \notin C$ and $e_i \notin C^\perp$.

Note that in the previous proposition we can replace G with any matrix whose rows form a system of generators of C, and are not necessarily linearly independent. In this case the parameter e (in the tuple defining an LSSS) is the number of the rows of this matrix.

The previous proposition says that we can use a code C as an LSSS as follows. Let $s \in \mathbb{F}_q$ be the secret, then we choose (uniformly random) $c \in C$ such that $c_0 = s$ and we share the c_i's. This is equivalent to choose the $\mathbb{F}_q$-linear map

$$\begin{array}{rccc} \phi\colon & \mathbb{F}_q^k & \to & \mathbb{F}_q, \\ & (x_1, \dots, x_k) & \mapsto & \sum_{i=1}^k a_i x_i \end{array}$$

where $a = (a_1, \dots, a_k)$ is such that $aG = c$, and share the $\phi(v_i)$'s.

Now we are going to investigate the relation between the parameters of the code (length, distance and dual distance) and the parameters of the associated LSSS (privacy and reconstruction). Again, we put $\mathcal{P} := \{1, \dots, n\}$.

Lemma 2.5.2. *Let C be an $[n+1, k]_q$ code such that $e_0 \notin C$ and $e_0 \notin C^\perp$, let $A \subseteq \mathcal{P}$. We have that $A \in \mathcal{A}(\Sigma(C))$ if and only if there exists $c \in C$ with $c_0 = 1$ and $\pi_A(c) = 0$.*

Proof. Let $c = aG \in C$, with $a \in \mathbb{F}_q^k$, be such that $c_0 = 1$ and $\pi_A(c) = 0$; this is equivalent to say that the $\mathbb{F}_q$-linear map $\phi\colon \mathbb{F}_q^k \to \mathbb{F}_q$ associated to a is such that $\phi(v_0) = c_0 = 1$ and $\phi(v_i) = c_i = 0$ for any $i \in A$, i.e. that $A \in \mathcal{A}(\Sigma(C))$. □

Lemma 2.5.3. *Let C be an $[n+1, k]_q$ code such that $e_0 \notin C$ and $e_0 \notin C^\perp$, let $B \subseteq \mathcal{P}$. We have that $B \in \Gamma(\Sigma(C))$ if and only if there exists $b = (b_i)_{i \in B} \in \mathbb{F}_q^{|B|}$ such that, for all $c \in C$, $c_0 = b \cdot \pi_B(c) = \sum_{i \in B} b_i c_i$.*

Proof. Clearly $c_0 = \sum_{i \in B} b_i c_i$ for all $c \in C$ if and only if $v_0 = \sum_{i \in B} b_i v_i \in \sum_{i \in B} \langle v_i \rangle$, i.e. $B \in \Gamma(\Sigma(C))$. □

Note that in the previous lemmas we assumed that $e_0 \notin C$ and $e_0 \notin C^\perp$ only to be allowed to construct $\Sigma(C)$; these hypotheses are not actually used in the proofs. In particular, we have proved that:

1. if $\tau \in \mathbb{N}$ is such that any $A \subseteq \mathcal{P}$ with $|A| = \tau$ satisfies the hypothesis of lemma 2.5.2 then $\Sigma(C)$ achieves τ-privacy;

2. if $\rho \in \mathbb{N}$ is such that any $B \subseteq \mathcal{P}$ with $|B| = \rho$ satisfies the hypothesis of lemma 2.5.3 then $\Sigma(C)$ achieves ρ-reconstruction.

We denote by $t(C)$ the largest $\tau \in \mathbb{N}$ such that $\Sigma(C)$ achieves τ-privacy, or we put $t(C) := 0$ if there is no such a τ, and by $r(C)$ the smallest $\rho \in \mathbb{N}$ such

that $\Sigma(C)$ achieves ρ-reconstruction. The assumptions $e_0 \notin C$ and $e_0 \notin C^\perp$ guarantee that $t(C)$ and $r(C)$ are well defined and satisfy $0 \leq t(C) < n$ and $0 \leq r(C) \leq n$. In the same way, we may define $t_i(C)$ and $r_i(C)$ for any $i \in \{0, \ldots, n\}$ such that $e_i \notin C$ and $e_i \notin C^\perp$.

We have the following important theorem.

Theorem 2.5.4. *Let C be an $[n+1, k, d]_q$ code with dual distance $d^\perp$ such that $e_0 \notin C$ and $e_0 \notin C^\perp$. Then $\Sigma(C)$ achieves:*

1. *$(d^\perp - 2)$-privacy;*
2. *$(n - d + 2)$-reconstruction.*

Proof. We have to prove that all sets $A \subseteq \mathcal{P}$ with $|A| = d^\perp - 2$ satisfy the hypothesis of lemma 2.5.2 and all sets $B \subseteq \mathcal{P}$ with $|B| = n - d + 2$ satisfy the hypothesis of lemma 2.5.3. Clearly, if $d^\perp \leq 2$ and $d \leq 2$ then there is nothing to prove, so we may assume $d^\perp > 2$ and $d > 2$.

Let $A \subseteq \mathcal{P}$ with $|A| = d^\perp - 2$, we claim that the map

$$\begin{array}{rccc} \pi_{0,A}\colon & C & \to & \left(\mathbb{F}_q, \mathbb{F}_q^{|A|}\right) = \mathbb{F}_q^{d^\perp - 1} \\ & c & \mapsto & (c_0, \pi_A(c)) \end{array}$$

is surjective, hence it suffices to pick $c \in {\pi_{0,A}}^{-1}(1, 0)$. Suppose $\pi_{0,A}$ is not surjective, i.e. $W := \operatorname{im} \pi_{0,A} \subsetneq \mathbb{F}_q^{d^\perp - 1}$; hence $W^\perp \neq 0$, i.e. we can find a codeword $c' \in C^\perp$ of weight $w(c') \leq d^\perp - 1$, a contradiction.

Let $B \subseteq \mathcal{P}$ with $|B| = n - d + 2$, consider the $[1 + |B|, k']_q$ code $C' := \pi_{0,B}(C)$, i.e. the code generated by the matrix $G' \in M_{k,1+|B|}(\mathbb{F}_q)$ whose columns are v_0 and v_i for $i \in B$ (notation as in proposition 2.5.1). Let $e'_0 \in \mathbb{F}_q^{1+|B|}$ be the first vector of the standard basis of $\mathbb{F}_q^{1+|B|}$; clearly $e'_0 \notin C'$, otherwise we can find $c \in C$ of weight $w(c) \leq 1 + (n + 1 - (1 + |B|)) = d - 1$, a contradiction, and $e'_0 \notin C'^\perp$, otherwise $v_0 = 0$ and $e_0 \in C^\perp$. Hence we can argue as in the proof of proposition 2.5.1 and find $b = (b_i)_{i \in B} \in \mathbb{F}_q^{|B|}$ such that, for all $c \in C$, $c_0 = \sum_{i \in B} b_i c_i$. □

Note that this result is independent of the choice of the coordinate $i \in \{0, \ldots, n\}$, provided that $e_0 \notin C$ and $e_0 \notin C^\perp$, i.e. that $\Sigma(C, i)$ is defined.

Example 2.5.5. The application of proposition 2.5.1 to a Reed-Solomon $[n+1, k]_q$ code C gives Shamir's LSSS $\Sigma(C)$. As C is MDS, it has minimum distance $d = n + 2 - k$ and dual distance $d^\perp = k + 1$, hence the previous theorem says (again) that $\Sigma(C)$ achieves $(k-1)$-privacy and k-reconstruction.

In the previous sections we have seen that:

- given a code C we can define his product code $\widehat{C}$;

- given an LSSS Σ satisfying some properties (see definitions 2.4.2 and 2.4.3) we can define an LSSS $\widehat{\Sigma}$.

Now, proposition 2.5.1 tells us that if $e_0 \notin \widehat{C}$ and $e_0 \notin \widehat{C}^\perp$ then we can define $\Sigma(\widehat{C})$, so it is natural to ask whether $\Sigma(\widehat{C}) = \widehat{\Sigma}(C)$. The answer is yes. More precisely, the following holds.

Proposition 2.5.6. *Let C be an $[n+1,k]_q$ code such that $e_0 \notin \widehat{C}$ and $e_0 \notin \widehat{C}^\perp$; then:*

1. *$e_0 \notin C$ and $e_0 \notin C^\perp$, hence*

$$\Sigma(C) = (\mathbb{F}_q, n, k, v_0, \langle v_1\rangle, \dots, \langle v_n\rangle),$$

   where v_i, for $i = 0, \dots, n$, is the i-th column of a (fixed) generator matrix G for C, is defined;

2. *$v_0 \otimes v_0 \neq 0$ and $v_0 \otimes v_0 \in \sum_{i=1}^n \langle v_i\rangle \otimes \langle v_i\rangle$, hence*

$$\widehat{\Sigma}(C) = (\mathbb{F}_q, n, k^2, v_0 \otimes v_0, \langle v_1\rangle \otimes \langle v_1\rangle, \dots, \langle v_n\rangle \otimes \langle v_n\rangle)$$

   is defined;

3. *if $g_1, \dots, g_k$ are the rows of G then $\widehat{G} := \{g_1 * g_1, \dots, g_1 * g_k, \dots, g_k * g_1, \dots, g_k * g_k\}$ is a system of generators of $\widehat{C}$, hence*

$$\Sigma(\widehat{C}) = (\mathbb{F}_q, n, k^2, w_0, \langle w_1\rangle, \dots, \langle w_n\rangle),$$

   where w_i, for $i = 0, \dots, n$, is the i-th column of the matrix whose rows are the vectors of $\widehat{G}$, is defined;

4. *$w_i = v_i \otimes v_i$ for all $i = 0, \dots, n$, hence*

$$\widehat{\Sigma}(C) = \Sigma(\widehat{C}).$$

Proof. It is easy to see that, for any $i \in \{0, \dots, n\}$, $e_i \notin \widehat{C} \implies e_i \notin C$ and $e_i \notin \widehat{C}^\perp \iff e_i \notin C^\perp$, hence $\Sigma(C)$ is defined by proposition 2.5.1.

Since $e_0 \notin C^\perp$, $v_0 \otimes v_0 \neq 0$. By lemma 2.4.4, $v_0 \otimes v_0 = \sum_{i=1}^n \lambda_i v_i \otimes v_i$ for some $\lambda_i \in \mathbb{F}_q$ if and only if

$$\sum_{i=1}^n \lambda_i (v_i \cdot b)(v_i \cdot b') = (v_0 \cdot b)(v_0 \cdot b') \text{ for all } b, b' \in \mathbb{F}_q^k,$$

and since the v_i's are the columns of a generator matrix for C this is equivalent to

$$\sum_{i=1}^n \lambda_i c_i c_i' = c_0 c_0' \text{ for all } c, c' \in C.$$

Since $e_0 \notin \widehat{C}$, this is true; indeed, arguing as in the proof of proposition 2.5.1, we can define an $\mathbb{F}_q$-linear map

$$\begin{array}{rccc} \rho_{\{1,\dots,n\}}: & \pi_{\{1,\dots,n\}}(\widehat{C}) & \to & \mathbb{F}_q \\ & \pi_{\{1,\dots,n\}}(c*c') = (c_1c'_1,\dots,c_nc'_n) & \mapsto & c_0c'_0 \end{array}$$

and take as $(\lambda_1,\dots,\lambda_n)$ the matrix associated to this map. Hence $\widehat{\Sigma}(C)$ is an LSSS.

The third statement is clear from the definition of $\widehat{C}$ and the last follows by construction of the w_i's and definition of $\otimes$. □

We define another important parameter. Let C be an $[n+1,k]_q$ code such that $e_0 \notin \widehat{C}$ and $e_0 \notin \widehat{C}^\perp$; we put

$$\widehat{t}(C) := \min\{t(C), n - r(\widehat{C})\}.$$

Again, we may define $\widehat{t}_i(C) := \min\{t_i(C), n - r_i(\widehat{C})\}$ for any $i \in \{0,\dots,n\}$ such that $e_i \notin \widehat{C}$ and $e_i \notin \widehat{C}^\perp$. In practice, it is useful to assume (renumbering the coordinates) that $\widehat{t}(C)$ is the biggest among these numbers.

We have the following important theorem.

Theorem 2.5.7. *Let C be an $[n+1,k]_q$ code such that $e_0 \notin \widehat{C}$ and $e_0 \notin \widehat{C}^\perp$. Then:*

1. *$\Sigma(C)$ has $\widehat{t}(C)$-strong multiplication;*
2. *$\Sigma(C)$ achieves $(n-2\widehat{t}(C))$-reconstruction;*
3. *$\widehat{t}(C) \leq \frac{n-1}{3}$.*

Proof. By definition $\widehat{t}(C) \leq t(C)$, hence $\Sigma(C)$ achieves $\widehat{t}(C)$-privacy, and $r(\widehat{C}) \leq n - \widehat{t}(C)$, hence $\widehat{\Sigma}(C)$ achieves $(n-\widehat{t}(C))$-reconstruction. This proves the first statement.

Let $B \subseteq \mathcal{P}$ with $|B| = n - 2\widehat{t}(C)$; hence we can write $\mathcal{P} \setminus B = A \cup A'$ with $|A| = |A'| = \widehat{t}(C)$. Let $c \in C$, we have to prove that there exists $b = (b_i)_{i\in B} \in \mathbb{F}_q^{|B|}$ such that $c_0 = \sum_{i\in B} b_i c_i$. By $\widehat{t}(C)$-privacy of $\Sigma(C)$ and lemma 2.5.2, there exists $c' \in C$ such that $c'_0 = 1$ and $\pi_A(c') = 0$. Consider $c * c' \in \widehat{C}$, note that $\pi_0(c*c') = c_0$ and $\pi_A(c*c') = 0$. Let $B' := \mathcal{P} \setminus A' = B \cup A$, $|B'| = n - \widehat{t}(C)$; by $(n-\widehat{t}(C))$-reconstruction of $\widehat{\Sigma}(C)$ and lemma 2.5.3, there exists $b' = (b'_i)_{i\in B'} \in \mathbb{F}_q^{|B'|}$ such that $\pi_0(c*c') = \sum_{i\in B'} b'_i \pi_i(c*c')$; hence

$$c_0 = \pi_0(c*c') = \sum_{i\in B'} b'_i\pi_i(c*c') = \sum_{i\in B} b'_i\pi_i(c*c') + \sum_{i\in A} b'_i\pi_i(c*c') = \sum_{i\in B}(b'_ic'_i)c_i,$$

so if we pick $b_i := b'_ic'_i$ for all $i \in B$ we conclude.

Finally, since $\Sigma(C)$ achieves $\widehat{t}(C)$-privacy and $(n-2\widehat{t}(C))$-reconstruction, we have $\widehat{t}(C)+1 \leq n-2\widehat{t}(C)$, which is equivalent to the last statement. □

Clearly, the same result holds replacing $\widehat{t}(C)$ by any integer $1 \leq t \leq \widehat{t}(C)$.

Later on, the following corollary will be useful.

Corollary 2.5.8. *Let C be an $[n+1,k]_q$ code with dual distance $d^\perp$ and product distance $\widehat{d}$ such that $e_0 \notin \widehat{C}$ and $e_0 \notin \widehat{C}^\perp$, $\widehat{t} \in \mathbb{N}$. If $\widehat{t} \leq d^\perp - 2$ and $\widehat{t} \leq \widehat{d} - 2$ then $\Sigma(C)$ has $\widehat{t}$-strong multiplication.*

Proof. It follows from theorem 2.5.4 and theorem 2.5.7. □

Example 2.5.9. Let C be a Reed-Solomon $[n+1,k]_q$ code; assume $2k-1 \leq n+1$, hence we can consider the Reed-Solomon $[n+1,2k-1]_q$ code $\widehat{C}$. We have $t(C) = k-1, r(\widehat{C}) = 2k-1$ and $\widehat{t}(C) = \min\{k-1, n-2k+1\}$. Assume $k = \frac{n+2}{3}$, then $\widehat{t}(C) = \frac{n-1}{3}$, which is the maximum value attainable by $\widehat{t}(C)$. The previous theorem says that Shamir's LSSS $\Sigma(C)$ has $(k-1)$-strong multiplication and achieves k-reconstruction.

2.6 Asymptotic notation

We write "for sufficiently large n" to mean "there exists $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$".

Definition 2.6.1 (big-O, little-o)**.** Let $g\colon \mathbb{N} \to \mathbb{R}_>$.

a) $O(g)$ is the set of all functions $f\colon \mathbb{N} \to \mathbb{R}$ such that, for some $c \in \mathbb{R}_>$ and for sufficiently large n, $|f(n)| \leq cg(n)$.

b) $o(g)$ is the set of all functions $f\colon \mathbb{N} \to \mathbb{R}$ such that $\frac{f(n)}{g(n)} \to 0$.

We can actually define the relations $f \in O(g)$ (f is big-O of g) and $f \in o(g)$ (f is little-o of g) for functions g and f defined on subsets of $\mathbb{N}$ of the form $\mathbb{N} \setminus S$, where S is a finite set. Moreover, by abuse of notation, we will often write (e.g. in sums) "$O(g(n))$" instead of "$f(n)$, for some $f \in O(g)$", and similarly for $f \in o(g)$.

3

Recent results

In this chapter we collect some recent results about the parameters of Schur product codes.

Section 3.1 is taken from [1]. Here we outline the proof of a lower bound for the asymptotic optimal corruption tolerance $\widehat{\tau}(q)$ (definition 3.1.3), which is a function of a prime power q (the size of a fixed field). The fact that $\widehat{\tau}(q) > 0$ allows us to use the construction given in section 2.5 to obtain a family (Σ_i) of LSSS's with $\widehat{t_i}$-strong multiplication over $\mathbb{F}_q$, on n_i players, such that $n_i \to \infty$ and $\widehat{t_i}$ is arbitrarily close to $\frac{n_i}{3}\tau(q)$. For example this means that, for any q and for any $\widehat{t}$, we can find an LSSS over $\mathbb{F}_q$ with $\widehat{t}$-strong multiplication.

Section 3.2 is taken from [10]. Here we construct a family of codes with good parameters (rate and relative minimum distance) whose products form a family which also has good parameters.

In these sections we actually use very similar techniques. First we give constructions of algebraic-geometric codes with good parameters (as codes and as LSSS's) which work for large values of the field size q, then using a field descent we construct codes over smaller fields, in order to let these results hold true for any choice of the field size q. The essential difference between the two construction is in the choice of the field descent map: in the first construction its purpose is to preserve the multiplication parameter of the LSSS associated to the code, whilst in the second case its aim is to control the parameters of the code itself. We may say that [1] is done from the secret sharing point of view, whilst in [10] the point of view is more coding theoretic (actually, this is partially true: for an application of [10], see [7]).

3.1 Bound on the corruption tolerance

Theorem 2.5.7 naturally yields the following definition.

Definition 3.1.1 (corruption tolerance)**.** Let C be an $[n+1,k]_q$ code such that $e_0 \notin \widehat{C}$ and $e_0 \notin \widehat{C}^{\perp}$. We call

$$\widehat{\tau}(C) := \frac{3\widehat{t}(C)}{n-1}$$

the corruption tolerance of C.

By theorem 2.5.7, $\widehat{\tau}(C) \leq 1$. For convenience, from here on we denote by $\mathcal{C}^{\dagger}(\mathbb{F}_q)$ the set of all codes C over $\mathbb{F}_q$ such that $e_0 \notin \widehat{C}$ and $e_0 \notin \widehat{C}^{\perp}$, i.e. such that the corruption tolerance $\widehat{\tau}(C)$ is defined.

Example 3.1.2. If C is a Reed-Solomon $[n+1, \frac{n+2}{3}]_q$ code then $\widehat{\tau}(C) = 1$ (see example 2.5.9).

Definition 3.1.3 (asymptotic optimal corruption tolerance)**.** Let q be a prime power. We call

$$\widehat{\tau}(q) := \limsup_{C \in \mathcal{C}^{\dagger}(\mathbb{F}_q)} \widehat{\tau}(C)$$

the asymptotic optimal corruption tolerance over $\mathbb{F}_q$.

Our purpose is to prove that $\widehat{\tau}(q) > 0$ for every prime power q. This means that for every prime power q there exists a family (C_i) of codes over $\mathbb{F}_q$, of length $n_i + 1$, such that $n_i \to \infty$ and

$$\frac{3\widehat{t}(C_i)}{n_i - 1} \to \widehat{\tau}(q) > 0.$$

This gives a family $(\Sigma(C_i))$ of LSSS's with $\widehat{t}_i$-strong multiplication over $\mathbb{F}_q$, on n_i players, such that $n_i \to \infty$ and $\widehat{t}_i$ is arbitrarily close to $\frac{n_i}{3}\tau(q)$.

The first step is to construct an algebraic code with some particular properties. For all the notions that are needed (function fields, places, divisors, genus, weak approximation theorem, Riemann-Roch theorem) one may refer to [13]. In particular, given a function field $\mathbb{F}$ over $\mathbb{F}_q$, $\mathbb{P}(\mathbb{F})$ denotes the set of all places of degree 1 of $\mathbb{F}$ and $g(\mathbb{F})$ denotes the genus of $\mathbb{F}$.

Definition 3.1.4 (algebraic-geometric code)**.** Let $\mathbb{F}$ be an algebraic function field over $\mathbb{F}_q$; let $P_0, \ldots, P_n \in \mathbb{P}(\mathbb{F})$ be pairwise distinct places of degree 1, $D := \sum_{i=0}^{n} P_i$, G a divisor such that $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$. We call

$$C(G, D) := \{(f(P_0), \ldots, f(P_n)) : f \in \mathcal{L}(G)\}$$

an algebraic-geometric code or Goppa code.

The code $C(G, D)$ defined above is an $[n+1, k, d]_q$ code. We have some information about its parameters:

1. if $\deg G \geq g(\mathbb{F})$ then $k \geq \deg G + 1 - g(\mathbb{F})$ and $d \geq n + 1 - \deg G$;
2. if $\deg G > 2g(\mathbb{F}) - 2$ then $k = \deg G + 1 - g(\mathbb{F})$.

Theorem 3.1.5. *Let $\mathbb{F}$ be an algebraic function field over $\mathbb{F}_q$. For all $\widehat{t}, n \in \mathbb{N}$ such that $1 \leq \widehat{t} < n$, $|\mathbb{P}(\mathbb{F})| \geq n+1$ and $3\widehat{t} < n - 4g(\mathbb{F})$ there exists a code $C \in \mathcal{C}^{\dagger}(\mathbb{F}_q)$ of length $n+1$ such that $\widehat{t}(C) \geq \widehat{t}$.*

Proof. For simplicity, put $g := g(\mathbb{F})$. Let $P_0, \ldots, P_n \in \mathbb{P}(\mathbb{F})$ be pairwise distinct places of degree 1, let $D := \sum_{i=0}^{n} P_i$. By the weak approximation theorem, there exists a divisor G such that $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$ and $\deg G = 2g + \widehat{t}$. Consider the algebraic-geometric code

$$C := C(G, D) = \{(f(P_0), \ldots, f(P_n)) : f \in \mathcal{L}(G)\},$$

we claim that the dual distance $d^{\perp}$ and the product distance $\widehat{d}$ of C satisfy:

1. $d^{\perp} > \widehat{t} + 1$;
2. $\widehat{d} > \widehat{t} + 1$.

This clearly implies $C \in \mathcal{C}^{\dagger}(\mathbb{F}_q)$, and $\widehat{t}(C) \geq \widehat{t}$ follows by theorem 2.5.4.

Let $i \in \{0, \ldots, n\}$, $A \subseteq \{0, \ldots, n\} \setminus \{i\}$ with $|A| = \widehat{t}$. Since

$$2g - 2 < \deg\left(G - P_i - \sum_{j \in A} P_j\right) < \deg\left(G - \sum_{j \in A} P_j\right)$$

the Riemann-Roch theorem yields

$$\dim \mathcal{L}\left(G - P_i - \sum_{j \in A} P_j\right) < \dim \mathcal{L}\left(G - \sum_{j \in A} P_j\right),$$

hence there exists $f \in \mathcal{L}(G)$ such that $f(P_i) = 1$ and $f(P_j) = 0$ for all $j \in A$. This proves that $d^{\perp} > \widehat{t} + 1$.

As to the second claim, note that $f, g \in \mathcal{L}(G)$ implies $fg \in \mathcal{L}(2G)$, hence $\widehat{C} \subseteq C(2G, D)$, hence $\widehat{C}$ has minimum distance

$$\widehat{d} \geq n + 1 - \deg 2G = n + 1 - 2(2g + \widehat{t}) > \widehat{t} + 1.$$

□

Note that, in general, there may be no $\widehat{t}, n \in \mathbb{N}$ satisfying the hypotheses of the theorem. However, they exist under the additional assumption (on the function field $\mathbb{F}$) that $|\mathbb{P}(\mathbb{F})| > 4(g(\mathbb{F}) + 1)$.

We want to use this theorem to construct a family of codes which allows us to bound $\widehat{\tau}(q)$ from below.

Definition 3.1.6 (Ihara's constant)**.** Let q be a prime power. We call

$$A(q) := \limsup_{g\to\infty} \frac{N_q(g)}{g},$$

where $N_q(g)$ denotes the maximum, over all function fields $\mathbb{F}$ of genus g whose constant field is $\mathbb{F}_q$, of $|\mathbb{P}(\mathbb{F})|$, Ihara's constant.

The Drinfeld-Vladuts bound states that $A(q) \leq \sqrt{q} - 1$. Moreover, we have the following bounds on Ihara's constant.

Theorem 3.1.7 (Ihara [6])**.** *Let q be a prime power, If q is a square then $A(q) = \sqrt{q} - 1$.*

Theorem 3.1.8 (Serre [11])**.** *There exists $c \in \mathbb{R}_>$ such that, for all prime powers q, $A(q) \geq c \log q$.*

Putting together these theorems, we get the following result.

Theorem 3.1.9 (Chen and Cramer [2])**.** *Let q be a prime power.*

1. *If $A(q) > 4$ then*
$$\widehat{\tau}(q) \geq 1 - \frac{4}{A(q)} > 0.$$
2. *If $q \geq 49$ and q is a square then*
$$\widehat{\tau}(q) \geq 1 - \frac{4}{\sqrt{q} - 1}.$$

Moreover, $\lim_{q\to\infty} \widehat{\tau}(q) = 1$.

Proof. By definition of $A(q)$, we can find a family $(\mathbb{F}_i)$ of function fields with $g(\mathbb{F}_i) \to +\infty$ such that $\frac{|\mathbb{P}(\mathbb{F}_i)|}{g(\mathbb{F}_i)} \to A(q) > 4$.

For any function field $\mathbb{F}_i$ in the family, put $n_i := |\mathbb{P}(\mathbb{F}_i)| - 1$, $\widehat{t}_i := \lfloor \frac{n_i - 4g(\mathbb{F}_i)}{3} \rfloor$. Note that if $\frac{n_i - 4g(\mathbb{F}_i)}{3} \geq 1$ then $\widehat{t}_i$ and n_i satisfy the hypotheses of the theorem, and as

$$n_i - 4g(\mathbb{F}_i) = |\mathbb{P}(\mathbb{F}_i)| - 4g(\mathbb{F}_i) - 1 = \left(\frac{\mathbb{P}(\mathbb{F}_i)}{g(\mathbb{F}_i)} - 4 \right) g(\mathbb{F}_i) - 1$$

and $A(q) > 4$ this is true at least for sufficiently large $g(\mathbb{F}_i)$. Hence we get a family (C_i) of codes with

$$\widehat{\tau}(C_i) = \frac{3\widehat{t}(C_i)}{n_i - 1} \geq \frac{3\widehat{t}_i}{n_i - 1} \to 1 - \frac{4}{A(q)}.$$

This proves the first claim.

The second claim follows from the first claim and theorem 3.1.7, the final statement from the first claim and theorem 3.1.8. □

Theorem 3.1.9 gives a lower bound for $\tau(q)$ in the case of $q \geq 49$, q square. It remains to bound $\widehat{\tau}(q)$ in the following cases:

- $2 \leq q < 49$;
- $q > 49$ is not a square.

We deal with these cases using the following notion.

Definition 3.1.10 (multiplication-friendly embedding)**.** A multiplication-friendly embedding of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ with expansion r is a pair (σ, ψ) of $\mathbb{F}_q$-linear maps $\sigma\colon \mathbb{F}_{q^m} \to \mathbb{F}_q^r$, $\psi\colon \mathbb{F}_q^r \to \mathbb{F}_{q^m}$ such that, for all $x, y \in \mathbb{F}_{q^m}$,

$$xy = \psi(\sigma(x) * \sigma(y)).$$

The existence of a multiplication-friendly embedding of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ allows us to associate to a code C over $\mathbb{F}_{q^m}$ of length $n+1$ a code C' over $\mathbb{F}_q$ of length $rn+1$ such that $\widehat{t}(C') \geq \widehat{t}(C)$. Precisely, the following theorem holds.

Theorem 3.1.11. *Let $C \in \mathcal{C}^\dagger(\mathbb{F}_{q^m})$ of length $n+1$, $1 \leq \widehat{t} \leq \widehat{t}(C)$; let (σ, ψ) be a multiplication-friendly embedding of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ with expansion r. Then there exists $C' \in \mathcal{C}^\dagger(\mathbb{F}_q)$ of length $rn+1$ such that $\widehat{t}(C') \geq \widehat{t}$.*

Proof. Consider $G := \{c \in C : c_0 \in \mathbb{F}_q\} \subseteq \mathbb{F}_q \oplus (\mathbb{F}_{q^m})^n$, which is an $\mathbb{F}_q$-linear subspace of C, and the $\mathbb{F}_q$-linear map

$$\begin{array}{rccc} \chi\colon & \mathbb{F}_q \oplus (\mathbb{F}_{q^m})^n & \to & \mathbb{F}_q^{1+rn}. \\ & (c_0, c_1, \ldots, c_n) & \mapsto & (c_0, \sigma(c_1), \ldots, \sigma(c_n)) \end{array}$$

Let $C' := \chi(G)$, it is easy to see that $C' \in \mathcal{C}^\dagger(\mathbb{F}_q)$; we claim that:

1. $\Sigma(C')$ achieves $\widehat{t}$-privacy;
2. $\widehat{\Sigma}(C')$ achieves $(rn - \widehat{t})$-reconstruction.

For convenience, we fix some notation. Put $\mathcal{P} := \{1, \ldots, n\}$ and $\mathcal{P}' := \{1, \ldots, rn\}$. For $A' \subseteq \mathcal{P}'$ we define the set $\beta(A') \subseteq \mathcal{P}$ of all "parents" of indexes in A' and the set $\alpha(A') \subseteq \mathcal{P}'$ of all "siblings" of indexes in $\beta(A')$ (for example, $1 \in \mathcal{P}$ is the parent of $1, \ldots, r \in \mathcal{P}'$, which are the siblings of $1 \in \mathcal{P}$, $2 \in \mathcal{P}$ is the parent of $r+1, \ldots, 2r \in \mathcal{P}'$, and so on). Finally, we distinguish the $\mathbb{F}_{q^m}$-linear projection $\pi_A\colon \mathbb{F}_{q^m}^{n+1} \to \mathbb{F}_{q^m}^{|A|}$ (for $A \subseteq \mathcal{P}$) and the $\mathbb{F}_q$-linear projection $\pi'_{A'}\colon \mathbb{F}_q^{rn+1} \to \mathbb{F}_q^{|A'|}$ (for $A' \subseteq \mathcal{P}'$).

Let $A' \subseteq \mathcal{P}'$ with $|A'| = \widehat{t}$. As $|\beta(A')| \leq |A'| = \widehat{t} \leq \widehat{t}(C)$, by lemma 2.5.2 there exists $c \in C$ such that $c_0 = 1$ and $\pi_{\beta(A')}(c) = 0$, hence $c' := \chi(c) \in C'$ is such that $c'_0 = 1$ and $\pi'_{\alpha(A')}(c') = 0$. By lemma 2.5.2 this implies the first claim.

Let $B' \subseteq \mathcal{P}'$ with $|B'| = rn - \widehat{t}$, let $A' := \mathcal{P}' \setminus B'$, then $|A'| = \widehat{t}$. Let $B := \mathcal{P} \setminus \beta(A') \subseteq \mathcal{P}$, then $|B| \geq n - \widehat{t}$, hence by lemma 2.5.3 there exists

$b = (b_i)_{i\in B} \in \mathbb{F}_{q^m}^{|B|}$ such that, for all $c, c' \in C$, $c_0 c'_0 = b \cdot \pi_B(c * c')$. We can extend $\psi\colon \mathbb{F}_q^r \to \mathbb{F}_{q^m}$ to $\overline{\psi}\colon \mathbb{F}_q \oplus (\mathbb{F}_q^r)^n \to \mathbb{F}_q \oplus \mathbb{F}_{q^m}^n$ in the natural way; note that, for all $c, c' \in G$, $\overline{\psi}(\chi(c) * \chi(c')) = c * c'$ and that if $x, y \in \mathbb{F}_q^{rn+1}$ are such that $\pi'_{B'}(x) = \pi'_{B'}(y)$ then $\pi_B(\overline{\psi}(x)) = \pi_B(\overline{\psi}(y))$. Hence we can define a linear form

$$\begin{array}{ccc} \mathbb{F}_q^{|B'|} & \to & \mathbb{F}_q, \\ \pi'_{B'}(x) & \mapsto & b \cdot \pi_B(\overline{\psi}(x)) \end{array}$$

let $b' \in \mathbb{F}_q^{|B'|}$ be the vector associated to this map. Now, for all $c, c' \in G$, we have

$$b' \cdot \pi'_{B'}(\chi(c) * \chi(c')) = b \cdot \pi_B(\overline{\psi}(\chi(c) * \chi(c'))) = b \cdot \pi_B(c * c') = c_0 c'_0$$

and by lemma 2.5.3 this implies the second claim. □

It is now natural to compare this theorem with theorem 3.1.5, as both of them construct a code C and prove that $C \in \mathcal{C}^\dagger(\mathbb{F}_q)$ and $\widehat{t}(C') \geq \widehat{t}$. However, their proofs are essentially different: in theorem 3.1.5 we proved that the code we constructed has good dual distance and product distance and has henceforth good secret sharing parameters, whilst in theorem 3.1.11 we proved directly that the secret sharing parameters of the code we constructed are good, ignoring the parameters of the code itself which, indeed, may be not good.

However, this theorem is useful only if we can prove the existence of the embeddings we need. The following theorem guarantees this, for the proof (which is an explicit construction of the embeddings) see [1].

Theorem 3.1.12. *1. There exists a multiplication-friendly embedding of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ with expansion $\binom{m+1}{2}$.*

2. Let $m \in \mathbb{N}$, $2 \leq m \leq \frac{q+2}{2}$. Then there exists a multiplication-friendly embedding of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$ with expansion $2m-1$.

Putting everything together we get the following result.

Theorem 3.1.13. *Let q be a prime power. Then $\widehat{\tau}(q) \geq \nu(q)$, where*

$$\nu(q) := \begin{cases} \frac{1}{35} & \text{if } q = 2 \\ \frac{1}{18} & \text{if } q = 3 \\ \frac{3}{35} & \text{if } q = 4 \\ \frac{5}{54} & \text{if } q = 5 \\ 1 - \frac{4}{\sqrt{q}-1} & \text{if } q \geq 49, q \text{ square} \\ \frac{1}{3}\left(1 - \frac{4}{q-1}\right) & \text{otherwise} \end{cases}.$$

Proof. The case of $q \geq 49$, q square is theorem 3.1.9.

Let $7 \leq q < 49$ or $q \geq 49$, q not square. Then we know that $\widehat{\tau}(q^2) \geq 1 - \frac{4}{q-1}$ and using a multiplication-friendly embedding of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$ with

expansion 3 (it exists by theorem 3.1.12) we get $\widehat{\tau}(q) \geq \frac{1}{3}\widehat{\tau}(q^2)$ by theorem 3.1.11.

The other cases are similar: for $q = 4$ use an embedding of $\mathbb{F}_{64}$ over $\mathbb{F}_4$ with expansion 5, for $q = 2, 3, 5$ use embeddings of $\mathbb{F}_{q^2}$ over $\mathbb{F}_q$ with expansion 3. □

Example 3.1.14. In the case of $q = 2$ the previous theorem says that we can find a family (Σ_i) of LSSS's with $\widehat{t}_i$-strong multiplication over $\mathbb{F}_2$, on n_i players, such that $n_i \to \infty$ and $\widehat{t}_i$ is arbitrarily close to $\frac{n_i}{3}\tau(2) \geq \frac{n_i}{105}$. Hence we can find an LSSS Σ with $\widehat{t}$-strong multiplication over $\mathbb{F}_2$, on n players, such that $\widehat{t} \geq \frac{1}{105}n$.

Finally, we state the following non-asymptotic upper bound on corruption tolerance. For a proof, see [1].

Theorem 3.1.15. *Let $C \in \mathcal{C}^{\dagger}(\mathbb{F}_q)$ of length $n + 1$. Then*

$$\widehat{\tau}(C) \leq 1 - \frac{\log_q(n+2) - 2}{2n - 2}.$$

3.2 An asymptotically good family

In this section we report a Randriambololona's construction (see [10]), which gives an asymptotically good family (C_i) of codes over $\mathbb{F}_q$, of length going to infinity, such that the family $(\widehat{C}_i)$ is also asymptotically good.

Recall that for an $[n, k, d]_q$ code C we have defined the rate $R(C) := \frac{k}{n}$ and the relative minimum distance $\delta(C) := \frac{d}{n}$. Moreover, in this section, we denote by $d(C)$ the distance of the code C.

Definition 3.2.1 (asymptotically good family)**.** Let (C_i) be a family of codes over $\mathbb{F}_q$, of length going to infinity. We say that the family is asymptotically good if both $R(C_i)$ and $\delta(C_i)$ admit a positive asymptotic lower bound, i.e. if there exist $\varepsilon, \varepsilon' > 0$ such that

$$\liminf_i R(C_i) \geq \varepsilon \quad \text{and} \quad \liminf_i \delta(C_i) \geq \varepsilon'.$$

As $R(\widehat{C}_i) \geq R(C_i)$ and $\delta(\widehat{C}_i) \leq \delta(C_i)$, we are looking for a family (C_i), of length going to infinity, such that there exist $\varepsilon, \varepsilon' > 0$ such that

$$\liminf_i R(C_i) \geq \varepsilon \quad \text{and} \quad \liminf_i \delta(\widehat{C}_i) \geq \varepsilon'.$$

The core in Randriambololona's construction is a method which allows to construct a code over a small alphabet from a code over a large alphabet: precisely, it associates to any code C over $\mathbb{F}_{q^{2s+1}}$ of length n a code C' over $\mathbb{F}_q$ of length $(s + 1)(2s + 1)n$.

Put $r := 2s+1$, let $\{\gamma_1, \ldots, \gamma_r\}$ be an $\mathbb{F}_q$-basis of $\mathbb{F}_{q^r}$ To each $a \in \mathbb{F}_{q^r}$ we can associate a linear form

$$\begin{array}{rccc} t_a\colon & \mathbb{F}_{q^r} & \to & \mathbb{F}_q, \\ & x & \mapsto & \mathrm{Tr}(ax) \end{array}$$

where $\mathrm{Tr}\colon \mathbb{F}_{q^r} \to \mathbb{F}_q$ denotes the trace map. Then $\{t_{\gamma_1}, \ldots, t_{\gamma_r}\}$ is an $\mathbb{F}_q$-basis of the dual space of $\mathbb{F}_{q^r}$.

We order the set $\{t_{\gamma_i} : 1 \leq i \leq r\} \cup \{t_{\gamma_i+\gamma_j} : 1 \leq i < j \leq r\}$ and we rename its elements as $\phi_1, \ldots, \phi_{\frac{r(r+1)}{2}} = \phi_{(s+1)(2s+1)}$ (note that $\frac{r(r+1)}{2} = (s+1)(2s+1)$). This naturally defines an injective map

$$\phi = (\phi_1, \ldots, \phi_{(s+1)(2s+1)})\colon \mathbb{F}_{q^{2s+1}} \to \mathbb{F}_q^{(s+1)(2s+1)},$$

hence associates to any $[n,k]_{q^{2s+1}}$ code C an $[(s+1)(2s+1)n, (2s+1)k]_q$ code $C' := \phi(C)$. However, in order to have some information about its minimum distance, we need to do some additional work which involves bilinear algebra.

Given an $\mathbb{F}_q$-vector space V, we denote by $\mathrm{Sym}(V, \mathbb{F}_q)$ the $\mathbb{F}_q$-vector space of symmetric bilinear forms on V. For all $i = 1, \ldots, \frac{r(r+1)}{2}$, we define $\phi_i^{\otimes 2} \in \mathrm{Sym}(\mathbb{F}_{q^r}, \mathbb{F}_q)$ as

$$\begin{array}{rccc} \phi_i^{\otimes 2}\colon & \mathbb{F}_{q^r} \times \mathbb{F}_{q^r} & \to & \mathbb{F}_q; \\ & (x,y) & \mapsto & \phi_i(x)\phi_i(y) \end{array}$$

one can prove that

$$\{\phi_i^{\otimes 2}\}_{i=1,\ldots,\frac{r(r+1)}{2}}$$

is an $\mathbb{F}_q$-basis of $\mathrm{Sym}(\mathbb{F}_{q^r}, \mathbb{F}_q)$. For all $j \in \mathbb{N}_>$, we define a symmetric $\mathbb{F}_q$-bilinear map

$$\begin{array}{rccc} m_j\colon & \mathbb{F}_{q^r} \times \mathbb{F}_{q^r} & \to & \mathbb{F}_{q^r} \\ & (x,y) & \mapsto & xy^{q^j} + x^{q^j}y \end{array}$$

and we denote by m_0 the usual multiplication law on $\mathbb{F}_{q^r}$; one can prove that

$$\{t_{\gamma_i} \circ m_j\}_{\substack{i=1,\ldots,r \\ j=0,\ldots,s}}$$

is an $\mathbb{F}_q$-basis of $\mathrm{Sym}(\mathbb{F}_{q^r}, \mathbb{F}_q)$. We can naturally define two symmetric $\mathbb{F}_q$-bilinear maps

$$\Phi := (\phi_1^{\otimes 2}, \ldots, \phi_{(s+1)(2s+1)}^{\otimes 2})\colon \mathbb{F}_{q^{2s+1}} \times \mathbb{F}_{q^{2s+1}} \to \mathbb{F}_q^{(s+1)(2s+1)}$$

and

$$\Psi := (m_0, \ldots, m_s)\colon \mathbb{F}_{q^{2s+1}} \times \mathbb{F}_{q^{2s+1}} \to (\mathbb{F}_{q^{2s+1}})^{s+1}.$$

The following lemma holds.

Lemma 3.2.2. *Notation as above. There exists an $\mathbb{F}_q$-vector space isomorphism*

$$\theta\colon \mathbb{F}_q^{(s+1)(2s+1)} \to (\mathbb{F}_{q^{2s+1}})^{s+1}$$

such that $\theta \circ \Phi = \Psi$.

Proof. See [10]. □

Let C be an $[n,k]_{q^{2s+1}}$ code, let $C' := \phi(C)$ as above. For all $j \in \mathbb{N}$, we can consider the $\mathbb{F}_q$-vector space generated by $m_j(C,C)$ and define its minimum distance d_j as the distance taken in $(\mathbb{F}_{q^{2s+1}})^n$. Note that this is not a code in the usual sense, since it is an $\mathbb{F}_q$-vector subspace of an $\mathbb{F}_{q^{2s+1}}$-vector space.

Lemma 3.2.3. *Notation as above; then*

$$d(\widehat{C'}) \geq \min_{j=0,\dots,s} d_j.$$

Proof. Let $c \in \widehat{C}$, of weight $w < \min_{j=0,\dots,s} d_j$; we claim that $c = 0$.

By definition $c \in \mathbb{F}_q^{(s+1)(2s+1)n}$, but we can also see c as a word of length n over $\mathbb{F}_q^{(s+1)(2s+1)}$ and of weight $\tilde{w} \leq w$. Applying the isomorphism θ defined by lemma 3.2.2, we get a codeword $\theta(c) \in \langle \Psi(C,C) \rangle$ of weight $\tilde{w}$. Now consider, for $j = 0, \dots, s$, the j-th projection $\pi_j\colon (\mathbb{F}_{q^{2s+1}})^{s+1} \to \mathbb{F}_{q^{2s+1}}$ and note that by definition $m_j = \pi_j \circ \Psi$; the vector $\pi_j(\theta(c)) \in \langle m_j(C,C) \rangle$ has weight at most $\tilde{w} \leq w < d_j$, hence $\pi_j(\theta(c)) = 0$. As this holds for all j, we obtain that $\theta(c) = 0$, hence $c = 0$. □

The following lemma gives us the information about the minimum distance of $\widehat{C'}$ that we need. Here C^{1+q^s} denotes the $(1+q^s)$-th Schur product code,which can be formally defined by induction, in the obvious way.

Lemma 3.2.4. *Let C be an $[n,k]_{q^{2s+1}}$ code, $C' := \phi(C)$. Then*

$$d(\widehat{C'}) \geq d(C^{1+q^s}).$$

Proof. As $\langle m_j(C,C) \rangle \subseteq C^{1+q^j}$ we have $d_j \geq d(C^{1+q^j})$. It is easy to see that, in general, $d(C^t) \geq d(C^{t+1})$, hence $d(C^{1+q^j}) \geq d(C^{1+q^s})$ for all $j = 0, \dots, s$. Finally by lemma 3.2.3 we conclude. □

We construct an algebraic-geometric code as follows. Again, for the basic notions about function fields and algebraic-geometric codes, we refer to [13].

Proposition 3.2.5. *Let $\mathbb{F}$ be an algebraic function field over $\mathbb{F}_{q^{2s+1}}$, $P_1, \dots, P_n \in \mathbb{P}(\mathbb{F})$ pairwise distinct places of degree 1 with $n > (1+q^s)g(\mathbb{F})$; let $D := \sum_{i=1}^n P_i$, G a divisor of degree $g(\mathbb{F}) \leq \deg G < (1+q^s)^{-1}n$ such that $\operatorname{supp} G \cap \operatorname{supp} D = \emptyset$. Then we can define the algebraic-geometric code $C := C(G,D)$ and the corresponding concatenated code $C' := \phi(C)$ is such that:*

1. $\dim C' \geq (2s+1)(\deg G + 1 - g(\mathbb{F}))$;
2. $d(\widehat{C'}) \geq n - (1+q^s)\deg G$;
3. $R(C') \geq \frac{1}{s+1}\frac{\deg G+1-g(\mathbb{F})}{n}$;
4. $\delta(\widehat{C'}) \geq \frac{1}{(s+1)(2s+1)}\left(1 - \frac{(1+q^s)\deg G}{n}\right)$.

Proof. We know that $\dim C' = (2s+1)\dim C$ and $\dim C \geq \deg G+1-g(\mathbb{F})$, hence the first claim follows and the third claim follows from the first one.

The second claim follows from

$$d(\widehat{C'}) \geq d(C(G,D)^{1+q^s}) \geq d(C((1+q^s)G, D)) \geq n - (1+q^s)\deg G;$$

the only non trivial inequality is the first one, which follows from lemma 3.2.4. Finally, the last claim follows from the second one. □

Comparing this theorem with theorem 3.1.11, we see that in both of them we construct a code C' from an algebraic-geometric code C using a field descent. The essential difference is that the descent used in this theorem controls the parameters of the code, whilst in theorem 3.1.11 we were interested only in the parameters of the LSSS associated to the code we constructed.

Recall the definition of Ihara's constant (definition 3.1.6), one can prove that there exists $s \in \mathbb{N}$ such that $A(q^{2s+1}) > 1 + q^s$ (see [5]).

The main result of [10] is the following theorem.

Theorem 3.2.6. *Let s be such that $A(q^{2s+1}) > 1 + q^s$. Then, for all $1 < \mu < \frac{A(q^{2s+1})}{1+q^s}$, there exists a family (C_i) of codes over $\mathbb{F}_q$, of length going to infinity, such that*

$$\liminf_i R(C_i) \geq \frac{1}{s+1}\frac{\mu - 1}{A(q^{2s+1})}$$

and

$$\liminf_i \delta(\widehat{C_i}) \geq \frac{1}{(s+1)(2s+1)}\left(1 - \frac{(1+q^s)\mu}{A(q^{2s+1})}\right).$$

Proof. By definition of $A(q)$, we can find a family $(\mathbb{F}_i)$ of function fields over $\mathbb{F}_{q^{2s+1}}$ with $g(\mathbb{F}_i) \to +\infty$ such that $\frac{|\mathbb{P}(\mathbb{F}_i)|}{g(\mathbb{F}_i)} \to A(q^{2s+1})$. Let $(m_i) \subseteq \mathbb{N}$ be a sequence such that $\frac{m_i}{g(\mathbb{F}_i)} \to \mu$.

For any function field $\mathbb{F}_i$ in the family we can find $|\mathbb{P}(\mathbb{F}_i)|$ places of degree 1 $P_0, \ldots, P_{n_i}$, where $n_i := |\mathbb{P}(\mathbb{F}_i)| - 1$; put $G := m_i P_0$, $D := \sum_{j=1}^{n_i} P_j$. Then by proposition 3.2.5 we get a code $C_i \subseteq \mathbb{F}_q^{(s+1)(2s+1)n_i}$ with

$$R(C_i) \geq \frac{1}{s+1}\frac{m_i + 1 - g(\mathbb{F}_i)}{n_i}$$

and

$$\delta(\widehat{C_i}) \geq \frac{1}{(s+1)(2s+1)}\left(1 - \frac{(1+q^s)m_i}{n_i}\right).$$

Now it is easy to see that the family (C_i) has the required properties. □

The existence of an integer s such that $A(q^{2s+1}) > 1 + q^s$ is guaranteed by what we have said above.

Note that the construction given by the theorem actually works for any family $(\mathbb{F}_i)$ of function fields over $\mathbb{F}_{q^{2s+1}}$ with $g(\mathbb{F}_i) \to +\infty$ such that $\liminf_i \frac{|\mathbb{P}(\mathbb{F}_i)|}{g(\mathbb{F}_i)} \geq A'$, for some $1 + q^s < A' \leq A(q^{2s+1})$; all occurrences of $A(q^{2s+1})$ in the theorem are henceforth replaced by A'.

Example 3.2.7. [5] gives a family $(\mathbb{F}_i)$ of function fields over $\mathbb{F}_{2^9}$ with $g(\mathbb{F}_i) \to +\infty$ such that

$$\liminf_i \frac{|\mathbb{P}(\mathbb{F}_i)|}{g(\mathbb{F}_i)} \geq \frac{465}{23} > 17 = 1 + 2^4.$$

Let $\mu := \frac{186}{161}$. Then the previous theorem gives a family (C_i) of binary codes, of length going to infinity, such that

$$\liminf_i R(C_i) \geq \frac{1}{651} \quad \text{and} \quad \liminf_i \delta(\widehat{C}_i) \geq \frac{1}{1575}.$$

We consider this result from the secret sharing point of view. From what we have just said, we get a code C, of length n, with product distance $\widehat{d}$ arbitrarily close to $\frac{1}{1575}n$. Assume that $d^\perp \geq \widehat{d}$; then by corollary 2.5.8 $\Sigma(C)$ has $(\widehat{d} - 2)$-strong multiplication, so the LSSS associated to C has $\widehat{t}$-strong multiplication, with $\widehat{t}$ close to $\frac{1}{1575}n$. Comparing this result with example 3.1.14, we see that the parameters guaranteed for the codes constructed by proposition 3.2.5 are worse than the ones guaranteed for the codes constructed by theorem 3.1.11 from the secret sharing point of view.

4 Computer tests

We have seen in the section 2.5 the link between codes and LSSS's, as well as the relations between their parameters. For example, corollary 2.5.8 tells us that if a code C has dual distance $d^\perp > 2$ and its product $\widehat{C}$ has distance $\widehat{d} > 2$ then the LSSS $\Sigma(C)$ associated to C has 1-strong multiplication. So, it is natural to look for codes whose product dimension is not too large, otherwise the product distance would be too small.

This is indeed the purpose of this section, in which we list some experimental results: first we deal with random codes (over a fixed field, with fixed length and dimension, and with dual distance greater than 2), then with binary cyclic codes. In the second section we also state some basic facts about cyclic codes; for a more exhaustive discussion, we refer again to [8], [14] or to any other coding theory handbook. For all the computations, the computer algebra system PARI/GP has been used.

4.1 Random codes

In this section, through computer computations, we want to understand the likelihood of the Schur product of a code being trivial.

In each test, we fix a prime q, a dimension k and a number N of iterations. We always choose $n := 3k$ as length. We randomly generate N matrices of size $k \times n$ and rank k with entries in $\mathbb{F}_q$, of the form

$$G = \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & G' \\ & & 1 & \end{array}\right),$$

such that the code C generated by G has dual distance $d^\perp > 2$. Then we

compute the dimension of the product code $\widehat{C}$. We gather all outcomes in a table of the following form.

Product dimension	$\cdots$	$\widehat{k_i}$	$\cdots$
#	$\cdots$	# of codes of product dimension $\widehat{k_i}$	$\cdots$

In all but the last test, we fix $N := 1000$. We start with $k := 5$, so $n = 3k = 15$. The following five tables are the outcomes in the cases of $q = 2, 3, 5, 7, 11$ respectively.

Product dimension	12	13	14	15
#	2	87	578	333

Product dimension	12	13	14	15
#	2	74	536	388

Product dimension	13	14	15
#	6	266	728

Product dimension	14	15
#	168	832

Product dimension	14	15
#	92	908

Now let $k := 7$, so $n = 3k = 21$. We only report, in the following tables, the outcomes in the cases of $q = 2$ and $q = 3$. The same test with $q = 5, 7, 11$ produced N matrices of product rank 21.

Product dimension	19	20	21
#	1	42	957

Product dimension	20	21
#	20	980

Finally, for $k = 10$, $n = 3k = 30$ and $q = 2$, we obtained $N = 1000$ matrices of full product rank. The same test with the same parameters but with $N = 10000$ produced the following outcome.

Product dimension	29	30
#	2	9998

Of course, these tests do not prove anything. Anyway, they give the feeling that the product dimension of a code quickly becomes the maximum possible if parameters as code dimension and field size increase, suggesting that some lower bound on the product dimension could be obtained.

4.2 Cyclic codes

We start by defining cyclic codes and giving their basic properties.

Definition 4.2.1 (cyclic codes)**.** Let C be an $[n,k]_q$ code. We say that C is cyclic if $(c_0, c_1, \ldots, c_{n-1}) \in C$ implies $(c_1, \ldots, c_{n-1}, c_0) \in C$.

The map

$$\begin{array}{ccc} C & \hookrightarrow & \frac{\mathbb{F}_q[X]}{X^n-1} \\ (c_0, c_1, \ldots, c_{n-1}) & \mapsto & c_0 + c_1 X + \cdots + c_{n-1} X^{n-1} \end{array}$$

lets us identify (and we will always implicitly use this identification) any cyclic code of length n with an ideal of the ring $R := \frac{\mathbb{F}_q[X]}{X^n-1}$. Moreover, as R is a P.I.D., such an ideal is of the form (g), with $g \mid X^n - 1$ and $\deg g = n - k$. Therefore a generator matrix for C is

$$G = \begin{pmatrix} g \\ gX \\ \vdots \\ gX^{k-1} \end{pmatrix}.$$

Note that multiplication by X in R corresponds to a shift in C.

Now, it is easy to see that the product code $\widehat{C}$ of a cyclic code C of length n is again a cyclic code of length n, hence also $\widehat{C}$ can be identified with an ideal of R, hence with a polynomial $\widehat{g} \in \mathbb{F}_q[X]$ such that $\widehat{g} \mid X^n - 1$ and $\deg \widehat{g} = n - \widehat{k}$.

In particular, in the case of $q = 2$, as $C \subseteq \widehat{C}$, we have that $\widehat{g} \mid g$.

The following proposition tells us that we can easily compute $\widehat{g}$.

Proposition 4.2.2. *Let $C = (g)$ be a cyclic $[n,k]_q$ code. Then $\widehat{C} = (\widehat{g})$, where*

$$\widehat{g} := \gcd(g*g, g*gX, \ldots, g*gX^{k-1}) = \gcd\left(g*g, \frac{g*gX}{X}, \ldots, \frac{g*gX^{k-1}}{X^{k-1}}\right).$$

Proof. As $X^j \mid g * gX^j$ for $j = 0, \ldots, k-1$ and $X \nmid \widehat{g}$ (otherwise $X \mid \widehat{g} \mid X^n - 1$), the two gcd's are equal.

Clearly $\widehat{C}$ is generated by $\widehat{g} := \gcd(gX^i * gX^j : 0 \le i \le j \le k-1)$. Also note that if $i \le j$ then $gX^i * gX^j = X^i(g * gX^{j-i})$. Hence, for some

polynomials $a_{ij}, a'_{ij} \in \mathbb{F}_q[X]$, we can write

$$\begin{aligned}
\widehat{g} &= a_{00}g * g + a_{01}g * gX + \cdots + a_{0\,k-1}g * gX^{k-1} + \\
&+ a_{11}gX * gX + a_{12}gX * gX^2 + \cdots + a_{1\,k-1}gX * gX^{k-1} + \\
&+ \cdots + \\
&+ a_{k-1\,k-1}gX^{k-1} * gX^{k-1} = \\
&= a_{00}g * g + a_{01}g * gX + \cdots + a_{0\,k-1}g * gX^{k-1} + \\
&+ a'_{11}g * g + a'_{12}g * gX + \cdots + a'_{1\,k-1}g * gX^{k-2} + \\
&+ \cdots + \\
&+ a'_{k-1\,k-1}g * g = \\
&= a'_{00}g * g + a'_{01}g * gX + \cdots + a'_{0\,k-1}g * gX^{k-1}
\end{aligned}$$

and the conclusion follows. □

We now repeat the work done in the previous section, using binary cyclic codes instead of randomly generated codes. Note that proposition 4.2.2 gives us an easy way to compute the product dimension, hence we are allowed to perform our computations on codes of higher dimension. Again, we are interested in counting the cyclic codes (of given dimension k and length n) with dual distance $d^\perp > 2$ which have a certain product dimension $\widehat{k}$.

The outcomes are listed in the following table.

k	n	outcomes	k	n	outcomes
5	15	2 codes with $\widehat{k} = 11$	7	21	2 codes with $\widehat{k} = 19$
8	24	1 code with $\widehat{k} = 20$	10	30	2 codes with $\widehat{k} = 22$ 3 codes with $\widehat{k} = 24$ 1 code with $\widehat{k} = 28$ 2 codes with $\widehat{k} = 29$ 3 codes with $\widehat{k} = 30$
11	33	2 codes with $\widehat{k} = 31$	12	36	1 code with $\widehat{k} = 30$ 1 code with $\widehat{k} = 36$
13	39	2 codes with $\widehat{k} = 39$	14	42	2 codes with $\widehat{k} = 33$ 3 codes with $\widehat{k} = 34$ 4 codes with $\widehat{k} = 38$ 6 codes with $\widehat{k} = 39$ 1 code with $\widehat{k} = 40$ 2 codes with $\widehat{k} = 41$ 4 codes with $\widehat{k} = 42$
15	45	2 codes with $\widehat{k} = 33$ 3 codes with $\widehat{k} = 45$	16	48	2 codes with $\widehat{k} = 40$ 1 code with $\widehat{k} = 44$
17	51	14 codes with $\widehat{k} = 51$	19	57	2 codes with $\widehat{k} = 55$

k	n	outcomes	k	n	outcomes
20	60	2 codes with $\widehat{k}=44$ 4 codes with $\widehat{k}=48$ 3 codes with $\widehat{k}=50$ 6 codes with $\widehat{k}=52$ 4 codes with $\widehat{k}=54$ 4 codes with $\widehat{k}=55$ 13 codes with $\widehat{k}=56$ 2 codes with $\widehat{k}=57$ 10 codes with $\widehat{k}=58$ 2 codes with $\widehat{k}=59$ 33 codes with $\widehat{k}=60$	21	63	8 codes with $\widehat{k}=54$ 6 codes with $\widehat{k}=56$ 8 codes with $\widehat{k}=57$ 8 codes with $\widehat{k}=60$ 6 codes with $\widehat{k}=61$ 43 codes with $\widehat{k}=62$ 208 codes with $\widehat{k}=63$
22	66	3 codes with $\widehat{k}=54$ 2 codes with $\widehat{k}=62$ 3 codes with $\widehat{k}=64$	23	69	2 codes with $\widehat{k}=67$
24	72	4 codes with $\widehat{k}=60$ 1 code with $\widehat{k}=64$ 1 code with $\widehat{k}=66$ 3 codes with $\widehat{k}=72$	25	75	2 codes with $\widehat{k}=55$ 2 codes with $\widehat{k}=71$ 4 codes with $\widehat{k}=75$
26	78	3 codes with $\widehat{k}=64$ 1 code with $\widehat{k}=76$ 2 codes with $\widehat{k}=77$ 2 codes with $\widehat{k}=78$	28	84	4 codes with $\widehat{k}=66$ 3 codes with $\widehat{k}=68$ 12 codes with $\widehat{k}=69$ 4 codes with $\widehat{k}=70$ 4 codes with $\widehat{k}=72$ 4 codes with $\widehat{k}=73$ 4 codes with $\widehat{k}=74$ 6 codes with $\widehat{k}=75$ 8 codes with $\widehat{k}=76$ 32 codes with $\widehat{k}=78$ 4 codes with $\widehat{k}=79$ 25 codes with $\widehat{k}=80$ 16 codes with $\widehat{k}=81$ 26 codes with $\widehat{k}=82$ 4 codes with $\widehat{k}=83$ 88 codes with $\widehat{k}=84$
29	87	2 codes with $\widehat{k}=87$	30	90	2 codes with $\widehat{k}=66$ 3 codes with $\widehat{k}=72$ 9 codes with $\widehat{k}=74$ 6 codes with $\widehat{k}=78$ 1 code with $\widehat{k}=84$ 2 codes with $\widehat{k}=86$ 8 codes with $\widehat{k}=87$ 4 codes with $\widehat{k}=88$ 4 codes with $\widehat{k}=89$ 95 codes with $\widehat{k}=90$

Recall that in the test described in the previous section we found only 2 random codes (out of 10000 attempts), of dimension 10, length 30 and dual distance greater than 2, whose product is not the whole space; here we see that 8 cyclic codes of dimension 10, length 30 and dual distance greater than 2 (out of 11) have product smaller than the whole space. Intuitively, this should say that cyclic codes have a small product dimension.

On the other hand, there are also cases in which all cyclic codes have full product rank, for example for $k = 17, 29$, and cases in which all cyclic codes have almost full product rank, for example for $k = 7, 11, 19, 23$.

Now recall corollary 2.5.8, which tells us that if a code C has dual distance $d^\perp \geq \widehat{t} + 2$ and its product $\widehat{C}$ has distance $\widehat{d} \geq \widehat{t} + 2$ (for some $\widehat{t} \in \mathbb{N}$) then the LSSS $\Sigma(C)$ has 1-strong multiplication, so it is natural to ask some information about dual distance and product distance of the cyclic codes we are dealing with. We have that the two $[15, 5]_2$ codes have dual distance $d^\perp = 4$ and product distance $\widehat{d} = 3$, so they give LSSS's with 1-strong multiplication; these codes are generated by the polynomials

$$X^{10}+X^9+X^8+X^6+X^5+X^2+1 \quad \text{and} \quad X^{10}+X^8+X^5+X^4+X^2+X+1.$$

Unfortunately, dual distance and product distance do not improve for larger length and dimension. Precisely, we find out that

- the two $[30, 10]_2$ codes with $\widehat{k} = 22$,
- the two $[45, 15]_2$ codes with $\widehat{k} = 33$,
- the two $[60, 20]_2$ codes with $\widehat{k} = 44$,

have again $d^\perp = 4$ and $\widehat{d} = 3$, and these are the only binary cyclic codes of dimension $k \leq 20$ and $n = 3k$ such that $d^\perp > 2$ and $\widehat{d} > 2$. We also note that these codes are generated by the polynomials

$$X^{10i}+X^{9i}+X^{8i}+X^{6i}+X^{5i}+X^{2i}+1 \text{ and } X^{10i}+X^{8i}+X^{5i}+X^{4i}+X^{2i}+X^i+1$$

for $i = 2, 3, 4$. This is not casual: these codes are constructed from the two $[15, 5]_2$ codes. For example, let C be the $[15, 5]_2$ code generated by $X^{10} + X^9 + X^8 + X^6 + X^5 + X^2 + 1$, then the $[30, 10]_2$ code C' generated by $X^{20} + X^{18} + X^{16} + X^{12} + X^{10} + X^4 + 1$ is the linear span of all vectors of the form

$$(c_0, 0, c_1, 0, \ldots, c_{n-1}, 0) \quad \text{and} \quad (0, c_0, 0, c_1, \ldots, 0, c_{n-1})$$

with $(c_0, c_1, \ldots, c_{n-1}) \in C$. This preserves distance and dual distance. Moreover, proposition 4.2.2 tells us that also $\widehat{C'}$ can be obtained from $\widehat{C}$ in the same way, hence product distance and product dual distance are preserved as well.

5

Lower bound on the product dimension

From here on, q will be a fixed prime power.

Our purpose is to prove the following result.

Theorem 5.0.3. *For all $\varepsilon > 0$, for all $t \in \mathbb{N}$, for all $[n,k]_q$ codes with dual distance $d^\perp \geq 2t+1$, we have*

$$\widehat{k} \geq k + \left(\frac{1}{2} - \varepsilon\right) t \log_q^2(n-k) + o\left(\log_q^2(n-k)\right).$$

First note that for $t = 0$ this result is trivial, hence we may assume $t \geq 1$. So all the codes we are dealing with have dual distance $d^\perp \geq 2t+1 > 2$.

The following example shows that there exist $[n,k]_q$ codes with $d^\perp = 3$ and $\widehat{k} = k + \frac{1}{2}\log_q^2(n-k) + O\left(\log_q(n-k)\right)$, therefore the lower bound of the theorem, at least in the case when $d^\perp = 3$, is asymptotically the best possible. However, for higher dual distance, it can be improved; this is essentially due to the fact that the Hamming bound, which is used in the proof, is loose for large dual distance.

Example 5.0.4. Let $h \in \mathbb{N}$, $N := \frac{q^h-1}{q-1}$, consider the simplex $[N,h]_q$ code C' (see example 2.2.5). C' has dual distance $d^\perp = 3$. We have seen that $\dim \widehat{C'} = \frac{h(h+1)}{2}$.

Let $k := q^h - 1$, let $G' \in M_{k,N}(\mathbb{F}_q)$ be the matrix whose rows are all the non zero codewords of C'. Let $n := k + N$, let $G \in M_{k,n}(\mathbb{F}_q)$ be the matrix whose first k columns form a $k \times k$ identity matrix and whose last N columns form G'.

Let C be the $[n,k]_q$ code generated by G. We have

$$\begin{aligned}\dim \widehat{C} &= \dim C + \dim \widehat{C'} = k + \frac{h(h+1)}{2} = \\ &= k + \frac{\log_q(N(q-1)+1)(\log_q(N(q-1)+1)+1)}{2} = \\ &= k + \frac{1}{2}\log_q^2(n-k) + O\left(\log_q(n-k)\right).\end{aligned}$$

In order to prove theorem 5.0.3, we adopt the following strategy:

1. we arrange the generator matrix of the code in a way which leads us to easily find linearly independent vectors of the product code (section 5.1);
2. using elementary algebraic tools, we bound from below the number of such vectors (section 5.2);
3. using elementary analytic tools, we show that the bound found at the previous step agrees, at least for sufficiently large values of $n-k$, with the statement of the theorem (section 5.3).

5.1 Arrangement lemmas

Let C be an $[n,k]_q$ code with dual distance $d^\perp > 2$. Let $G \in M_{k,n}(\mathbb{F}_q)$ be a generator matrix for C, written in standard form as

$$G = \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & G' \\ & & 1 & \end{array}\right)$$

with $G' \in M_{k,n-k}(\mathbb{F}_q)$. From here on, we put $N := n-k$; thus $G' \in M_{k,N}(\mathbb{F}_q)$.

We want to arrange G in a special way. The allowed operations are:

- swap columns (corresponding to a coordinate renumbering);
- perform elementary operations on the rows (corresponding to a basis change).

The first idea is to reorder G (actually the columns of G', i.e. the last N columns of G) as follows.

1. Initialisation: let $i := 1, G_i := G'$;
2. if $m_1 + \cdots + m_{i-1} < N$ then

   2.1 (basis change) choose as first row of G_i a non zero row;

2.2 (coordinate renumbering) let the first row of G_i be

$$(\underbrace{* \ \cdots \ *}_{m_i} \mid \underbrace{0 \ \cdots\cdots \ 0}_{N-(m_1+\cdots+m_i)});$$

2.3 let $i := i+1$, $G_i \in M_{k-(i-1),N-(m_1+\cdots+m_{i-1})}(\mathbb{F}_q)$ be the matrix formed by the last $k-(i-1)$ rows and $N-(m_1+\cdots+m_{i-1})$ columns of G_{i-1}, repeat step 2;

3. else put $\ell := i-1$ and output the rearranged G.

Note that $d^\perp > 2$ ensures that G' has no zero columns, hence step 2.1 is justified and eventually $m_1+\cdots+m_i = N$ and the algorithm stops.

After this arrangement, G' is written as

$$G' = \begin{pmatrix} * & \cdots & * & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ ? & \cdots & ? & * & \cdots & * & 0 & \cdots & \cdots & 0 \\ ? & \cdots & \cdots & ? & \cdots & ? & \ddots & 0 & \cdots & 0 \\ \vdots & & & & & & \ddots & * & \cdots & * \\ ? & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & ? & \cdots & ? \end{pmatrix},$$

where the stars mean a non-zero element of $\mathbb{F}_q$ and the question marks mean any element of $\mathbb{F}_q$.

We have proved the following arrangement lemma.

Lemma 5.1.1. *Let C be an $[n,k]_q$-code with dual distance $d^\perp > 2$. Then we can write the generator matrix of C as*

$$G = \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & G' \\ & & 1 & \end{array}\right),$$

where $G' = (G'_{i,j}) \in M_{k,N}(\mathbb{F}_q)$ has the following property: there exists a (finite) sequence $m_1, \ldots, m_\ell \in \mathbb{N}_>$ with $m_1+\cdots+m_\ell = N$ such that, for all $i = 1, \ldots, \ell$:

(i) for all $j = m_1+\cdots+m_{i-1}+1, \ldots, m_1+\cdots+m_i$, $G'_{i,j} \neq 0$;

(ii) for all $j > m_1+\cdots+m_i$, $G'_{i,j} = 0$.

Proof. Clear from the construction above. □

This arrangement naturally defines ℓ matrices $A_i \in M_{k-i+1,m_i}(\mathbb{F}_q)$ whose top row has no zero entries. These matrices give a natural decomposition of (a subspace of) $\widehat{C}$: if we take, for all $i = 1, \ldots, \ell$, the product of another non zero row of A_i and its first row then we obtain a family of linearly independent vectors of $\widehat{C}$, which generates a subspace having intersection 0 with (the copy contained in $\widehat{C}$ of) C.

In particular, as $d^\perp > 2$, A_i always has another non zero row, so all these matrices give a positive contribution. Moreover, we can find a linearly independent set of $\operatorname{rk} A_i - 1$ linearly independent rows of A_i which does not contain its first row. Thus, the decomposition above and lemma 2.2.2 yield

$$\widehat{k} \geq k + \sum_{i=1}^{\ell} \max\{\operatorname{rk} A_i - 1, 1\}. \tag{5.1.1}$$

We will see that $\operatorname{rk} A_i \geq f(m_i)$, where f is an increasing function. This means that we are interested in conditions on the m_i's. In order to do this, we use the Plotkin bound (theorem 2.1.10).

For $h \in \mathbb{N}$, we put

$$\alpha := \alpha(h) := 1 - \frac{q^h(q-1)}{q^{h+1}-1} = \frac{q^h - 1}{q^{h+1}-1};$$

$(\alpha(h))$ defines an increasing sequence converging to $\frac{1}{q}$.

Again, we start from a generator matrix G of an $[n,k]_q$ code C with $d^\perp > 2$ written in standard form. We arrange it as follows.

1. Initialisation: let $i := 1, G_i := G'$;
2. if $\operatorname{rk} G_i \geq h+1$ then
   - 2.1 (basis change) choose as first row of G_i a row of weight
   $$0 < m_i \leq (1-\alpha)(N - (m_1 + \cdots + m_{i-1})),$$
   - 2.2 (coordinate renumbering) let the first row of G_i be
   $$(\underbrace{* \ \cdots \ *}_{m_i} \mid \underbrace{0 \ \cdots\cdots \ 0}_{N-(m_1+\cdots+m_i)}),$$
   - 2.3 let $i := i+1$, $G_i \in M_{k-(i-1),N-(m_1+\cdots+m_{i-1})}(\mathbb{F}_q)$ be the matrix formed by the last $k-(i-1)$ rows and $N-(m_1+\cdots+m_{i-1})$ columns of G_{i-1}, repeat step 2;
3. else put $\ell' := \ell'(h) := i-1$ and use the previous algorithm on G_i.

Note that the Plotkin bound says exactly that, for an $[N-(m_1+\cdots+m_{i-1}), h+1, d]_q$ code, we have

$$d \leq (1-\alpha)(N - (m_1 + \cdots + m_{i-1})),$$

i.e. that step 2.1 is justified.

After this arrangement, we may assume that G is of the form

$$G = \left(\begin{array}{ccc|c} 1 & & ? & \\ & \ddots & & G' \\ 0 & & 1 & \end{array}\right),$$

where the question mark stands for any element of $\mathbb{F}_q$, with the constraint that, on the left part of the matrix, we have at most $h+1$ non zero entries per row, including the one on the diagonal, and only on the first ℓ' rows. G' will be again of the form

$$G' = \begin{pmatrix} * & \cdots & * & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ ? & \cdots & ? & * & \cdots & * & 0 & \cdots & \cdots & 0 \\ ? & \cdots & \cdots & ? & \cdots & ? & \ddots & 0 & \cdots & 0 \\ \vdots & & & & & & \ddots & * & \cdots & * \\ ? & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & ? & \cdots & ? \end{pmatrix},$$

but with bounds on the m_i's.

Hence we have a second arrangement lemma.

Lemma 5.1.2. *Let C be an $[n,k]_q$-code with dual distance $d^\perp > 2$; let $h \in \mathbb{N}$, $\alpha := \frac{q^h-1}{q^{h+1}-1}$. Then we can write the generator matrix of C as*

$$G = \left(\begin{array}{ccc|c} 1 & & ? & \\ & \ddots & & G' \\ 0 & & 1 & \end{array}\right),$$

where the question mark stands for any element of $\mathbb{F}_q$, with the constraint that we have at most $h+1$ non zero entries per row (including the one on the diagonal), and $G' = (G'_{i,j}) \in M_{k,N}(\mathbb{F}_q)$ has the following property: there exists a (finite) sequence $m_1, \ldots, m_\ell \in \mathbb{N}_>$ with $m_1 + \cdots + m_\ell = N$ such that, for all $i = 1, \ldots, \ell$:

(i) for all $j = m_1 + \cdots + m_{i-1} + 1, \ldots, m_1 + \cdots + m_i$, $G'_{i,j} \neq 0$;

(ii) for all $j > m_1 + \cdots + m_i$, $G'_{i,j} = 0$.

Moreover, there exists an index $\ell' < \ell$ such that:

(iii) the submatrix of the left part of G obtained by removing its first ℓ' rows and ℓ' columns is an $(k-\ell') \times (k-\ell')$ identity matrix;

(iv) the submatrix of G formed by its last $m_{\ell'+1} + \cdots + m_\ell$ columns has rank $< h+1$;

(v) for all $i = 1, \ldots, \ell'$,

$$m_i \leq (1-\alpha)(N - (m_1 + \cdots + m_{i-1})).$$

Proof. Clear from the construction above. □

We proof a simple arithmetic property of the m_i's.

Lemma 5.1.3. *Let $N \in \mathbb{N}_>$, $\alpha \in]0,1[$, $m_1, \ldots, m_{\ell'} \in \mathbb{N}_>$ such that, for all $i = 1, \ldots, \ell'$,*

$$m_i \leq (1-\alpha)(N - (m_1 + \cdots + m_{i-1})).$$

Then

$$\sum_{i=1}^{\ell'} m_i \leq (1-\alpha^{\ell'})N.$$

Proof. We argue by induction on ℓ'. The case of $\ell' = 1$ is true by hypothesis, so assume that $\ell' > 1$ and

$$\sum_{i=1}^{\ell'-1} m_i \leq (1-\alpha^{\ell'-1})N.$$

We have

$$\begin{aligned}\sum_{i=1}^{\ell'} m_i &= \sum_{i=1}^{\ell'-1} m_i + m_{\ell'} \leq \sum_{i=1}^{\ell'-1} m_i + (1-\alpha)\left(N - \sum_{i=1}^{\ell'-1} m_i\right) = \\ &= N - \alpha N + \alpha \sum_{i=1}^{\ell'-1} m_i \leq N - \alpha N + \alpha(1-\alpha^{\ell'-1})N = \\ &= N - \alpha^{\ell'} N = (1-\alpha^{\ell'})N.\end{aligned}$$

□

Again, we can write a lower bound for $\widehat{k}$ as the one of (5.1.1). Again, all matrices give a positive contribution. But in this case, we have to take into account the new non zero entries in the left part of G: for example, if the i-th entry of the first row of G is non zero then the product of the first row of G and its i-th row may belong to (the copy contained in $\widehat{C}$ of) C; hence, as there may be at most h such entries, except the one on the diagonal, we have

$$\widehat{k} \geq k + \sum_{i=1}^{\ell'} \max\{\operatorname{rk} A_i - 1 - h, 1\} + \sum_{i=\ell'+1}^{\ell} \max\{\operatorname{rk} A_i - 1, 1\}. \quad (5.1.2)$$

The last arrangement step is a lemma which allows us to assume that the m_i's form a decreasing sequence.

Lemma 5.1.4. *Let $M \in \mathbb{N}_>, \beta \in]0,1[$; let $m, m' \in \mathbb{N}_>$ be such that $m' \leq \beta(M-m)$. If $m < m'$ then*

1. *$m' \leq \beta M$,*
2. *$m \leq \beta(M - m')$.*

Proof. Let $s := m' - m \in \mathbb{N}_>$.

The first claim is trivial, since (as $\beta > 0$) $m' \leq \beta(M-m) \leq \beta M$.

The second claim is equivalent to $m + \beta m' \leq \beta M$ and we have

$$\begin{aligned} m + \beta m' &= m' + \beta m' - s \leq \beta(M-m) + \beta m' - s = \\ &= \beta M + \beta(m'-m) - s = \beta M - (1-\beta)s \leq \\ &\leq \beta M \end{aligned}$$

as $\beta < 1, s > 0$. □

If, in the sequence $m_1, \ldots, m_{\ell'} \in \mathbb{N}_>$ such that, for all $i = 1, \ldots, \ell'$,

$$m_i \leq (1-\alpha)(N - (m_1 + \cdots + m_{i-1})),$$

we have, for some index i, $m_i < m_{i+1}$ then applying this lemma with $M := N - (m_1 + \cdots + m_{i-1}), \beta := 1 - \alpha, m := m_i, m' := m_{i+1}$ we get a decreasing sequence with the same property. This arithmetic property of the m_i's will be useful later.

5.2 Algebraic step

Let $t \in \mathbb{N}_>$ and let C be an $[n,k]_q$ code with dual distance $d^\perp \geq 2t+1$. As $d^\perp > 2$, the arrangement lemmas in the previous section give a natural way to bound the product dimension $\widehat{k}$, by constructing ℓ submatrices $A_i \in M_{k-i+1,m_i}(\mathbb{F}_q)$ of a generator matrix of C such that inequalities as (5.1.1) and (5.1.2) hold. We estimate the rank of these matrices as follows.

Lemma 5.2.1. *Let $A \in M_{k',m}(\mathbb{F}_q)$ be a matrix whose columns are non zero and d'-wise linearly independent, put $t' := \lfloor \frac{d'}{2} \rfloor$. Then*

$$\operatorname{rk} A \geq \log_q \sum_{i=0}^{t'} \binom{m}{i} (q-1)^i.$$

Proof. This is a corollary of the Hamming bound (theorem 2.1.11) applied to the dual of the $[m, \operatorname{rk} A]_q$ code generated by the rows of A, which is an $[m, m - \operatorname{rk} A, d^\perp]_q$ code with $d^\perp > d'$. □

Corollary 5.2.2. *Let $A \in M_{k',m}(\mathbb{F}_q)$ be a matrix whose columns are non zero and pairwise linearly independent, let $h \in \mathbb{N}$. If $m \geq q^h$ then $\operatorname{rk} A \geq h+1$.*

Proof. Applying the previous lemma with $d' = 2$, we get

$$\operatorname{rk} A \geq \log_q(1 + m(q-1)) \geq \log_q(1 + q^h(q-1)) > h + \log_q(q-1) > h,$$

hence $\operatorname{rk} A \geq h+1$. □

Corollary 5.2.3. *Let $A \in M_{k',m}(\mathbb{F}_q)$ be a matrix whose columns are non zero and d'-wise linearly independent, put $t' := \lfloor \frac{d'}{2} \rfloor$.*

1. *If $t' \leq \frac{m}{2}$ then $\operatorname{rk} A \geq t' \log_q m - t' \log_q 2 - \log_q t'!$.*
2. *If $t' > \frac{m}{2}$ then $\operatorname{rk} A \geq m$.*

Proof. Clearly

$$\sum_{i=0}^{t'} \binom{m}{i} (q-1)^i \geq \binom{m}{t'} = \frac{m(m-1)\cdots(m-t'+1)}{t'!} \geq \frac{(m-t+1)^{t'}}{t'!},$$

hence if $t' \leq \frac{m}{2}$ then

$$\sum_{i=0}^{t'} \binom{m}{i} (q-1)^i \geq \frac{(\frac{m}{2})^{t'}}{t'!}$$

and by lemma 5.2.1 $\operatorname{rk} A \geq t' \log_q m - t' \log_q 2 - \log_q t'!$.

As to claim 2, if $t' > \frac{m}{2}$ then $d' > m$, hence $\operatorname{rk} A \geq m$. □

For convenience, from here on we put $\tau := t \log_q 2 + \log_q t!$ and

$$f_t(x) := \begin{cases} t \log_q x - \tau, & \text{if } x \geq 2t \\ x & \text{if } x < 2t \end{cases},$$

so the two claims of the previous lemma can be replaced by $\operatorname{rk} A \geq f_{t'}(m)$.

For any $d' < d^\perp$ the A_i's satisfy the hypothesis of this corollary, hence (5.1.1) yields

$$\widehat{k} \geq k + \sum_{i=1}^{\ell} \max\{f_t(m_i) - 1, 1\}$$

and (5.1.2) yields

$$\begin{aligned} \widehat{k} &\geq k + \sum_{i=1}^{\ell'} \max\{f_t(m_i) - 1 - h, 1\} + \sum_{i=\ell'+1}^{\ell} \max\{f_t(m_i) - 1, 1\} \geq \\ &\geq k + \sum_{i=1}^{\ell'} \max\{f_t(m_i) - 1 - h, 1\}. \end{aligned} \tag{5.2.1}$$

Put $N := n - k$. For $h \in \mathbb{N}$, we put

$$\alpha := \alpha(h) := \frac{q^h - 1}{q^{h+1} - 1}, \quad \gamma := \gamma(h) := \frac{1}{\log_q \frac{1}{\alpha}} - \frac{1}{h},$$
$$r := r(h) := \lfloor \gamma \log_q N \rfloor + 1;$$

note that these define increasing sequences and

$$\alpha(h) \nearrow \frac{1}{q}, \quad \gamma(h) \nearrow 1, \quad r(h) \nearrow \lfloor \log_q N \rfloor + 1.$$

From here on, assume $r \leq \ell'$. Let $\delta \in]0,1[$; let

$$r' := \min\{i \in \mathbb{N} : m_{i+1} < (\alpha^i N)^{1-\delta} \text{ or } i = \ell'\}$$

i.e. $m_i \geq (\alpha^{i-1}N)^{1-\delta}$ for all $i = 1, \ldots, r'$ and $m_{r'+1} < (\alpha^{r'}N)^{1-\delta}$. Moreover, assume that, for all $i = 1, \ldots, r'$, $t \leq \frac{m_i}{2}$. Hence, (5.2.1) yields

$$\begin{aligned}\widehat{k} &\geq k + \sum_{i=1}^{r'} \max\{t \log_q m_i - \tau - 1 - h, 1\} + \sum_{i=r'+1}^{\ell'} \max\{f_t(m_i) - 1 - h, 1\} \\ &\geq k + \sum_{i=1}^{r'} \max\{t \log_q m_i - \tau - 1 - h, 1\} + (\ell' - r'). \end{aligned} \tag{5.2.2}$$

We have broken the sum in (5.2.1) into two summands: a main part and a tail. The idea is that if r' is "large enough" then the main part is large enough to achieve the result stated in theorem 5.0.3, whilst if r' is "small" then we argue by considering the tail.

We start by considering the case of $r' \geq r$. First note that

$$\begin{aligned}\sum_{i=1}^{r'} \max\{t \log_q m_i - \tau - 1 - h, 1\} &\geq \sum_{i=1}^{r} \max\{t \log_q m_i - \tau - 1 - h, 1\} \geq \\ &\geq \sum_{i=1}^{r} (t \log_q m_i - \tau - 1 - h) = \\ &= t \sum_{i=1}^{r} \log_q m_i - r(\tau + 1 + h). \end{aligned} \tag{5.2.3}$$

By definition of r', we have

$$\begin{aligned}\sum_{i=1}^{r} \log_q m_i \geq \sum_{i=1}^{r} \log_q (\alpha^{i-1}N)^{1-\delta} &= \frac{1-\delta}{\log_{\frac{1}{\alpha}} q} \sum_{i=1}^{r} \left(\log_{\frac{1}{\alpha}} N - (i-1)\right) = \\ &= \frac{1-\delta}{\log_{\frac{1}{\alpha}} q} \left(r \log_{\frac{1}{\alpha}} N - \frac{r(r-1)}{2}\right). \end{aligned}$$

As $\gamma \log_{\frac{1}{\alpha}} N \leq r \leq \log_{\frac{1}{\alpha}} N + 1$, we have

$$\begin{aligned}\sum_{i=1}^{r} \log_q m_i &\geq \frac{1-\delta}{\log_{\frac{1}{\alpha}} q} \left(\gamma \log^2_{\frac{1}{\alpha}} N - \frac{\log_{\frac{1}{\alpha}} N (\log_{\frac{1}{\alpha}} N + 1)}{2}\right) = \\ &= \frac{1-\delta}{\log_{\frac{1}{\alpha}} q} \left(\gamma - \frac{1}{2}\right) \log^2_{\frac{1}{\alpha}} N - \frac{1-\delta}{2 \log_{\frac{1}{\alpha}} q} \log_{\frac{1}{\alpha}} N = \\ &= \frac{1-\delta}{\log_q \frac{1}{\alpha}} \left(\gamma - \frac{1}{2}\right) \log^2_q N - \frac{1-\delta}{2} \log_q N. \end{aligned} \tag{5.2.4}$$

Putting together (5.2.2), (5.2.3) and (5.2.4) we get

$$\widehat{k} \geq k + \frac{1-\delta}{\log_q \frac{1}{\alpha}} \left(\gamma - \frac{1}{2}\right) t \log_q^2 N - \frac{1-\delta}{2} t \log_q N - r(\tau + 1 + h). \quad (5.2.5)$$

Now we consider the case of $r' < r$. By lemma 5.1.4 we may assume that the m_i's form a decreasing sequence, hence

$$\sum_{i=r'+1}^{\ell'} m_i \leq (\ell' - r') m_{r'+1} < (\ell' - r')(\alpha^{r'} N)^{1-\delta}.$$

Moreover

$$\sum_{i=\ell'+1}^{\ell} m_i < q^h,$$

otherwise by corollary 5.2.2 the matrix formed by the last $\sum_{i=\ell'+1}^{\ell} m_i$ columns of G would have rank at least $h+1$, which is against the definition of ℓ' (see lemma 5.1.2). On the other hand, by lemma 5.1.3,

$$\sum_{i=r'+1}^{\ell} m_i = \sum_{i=1}^{\ell} m_i - \sum_{i=1}^{r'} m_i \geq N - (1 - \alpha^{r'})N = \alpha^{r'} N.$$

Putting everything together we obtain

$$\ell' - r' \geq \frac{\alpha^{r'} N - q^h}{(\alpha^{r'} N)^{1-\delta}} = (\alpha^{r'} N)^{\delta} \left(1 - \frac{q^h}{\alpha^{r'} N}\right). \quad (5.2.6)$$

As $r' < r \leq \gamma \log_q N + 1 = \log_{\frac{1}{\alpha}} N - \frac{1}{h} \log_{\frac{1}{\alpha}} N \log_q \frac{1}{\alpha} + 1$, we have that

$$\alpha^{r'} > \alpha N^{-1} N^{\frac{1}{h} \log_q \frac{1}{\alpha}},$$

hence (5.2.6) yields

$$\ell' - r' \geq \alpha^{\delta} N^{\frac{\delta}{h} \log_q \frac{1}{\alpha}} - \frac{q^h}{\alpha^{1-\delta} N^{\frac{1-\delta}{h} \log_q \frac{1}{\alpha}}}. \quad (5.2.7)$$

Putting together (5.2.2) and (5.2.7) we get

$$\widehat{k} \geq k + \alpha^{\delta} N^{\frac{\delta}{h} \log_q \frac{1}{\alpha}} - \frac{q^h}{\alpha^{1-\delta} N^{\frac{1-\delta}{h} \log_q \frac{1}{\alpha}}}. \quad (5.2.8)$$

5.3 Analytic step

As in the previous section, let $t \in \mathbb{N}_>$ and $\tau := t \log_q 2 + \log_q t!$. Let C be an $[n,k]_q$ code with dual distance $d^\perp \geq 2t+1$ and generator matrix G arranged as in lemma 5.1.2 (so $m_1, \ldots, m_\ell$ and ℓ' are defined); put $N := n - k$.

In the previous section we have defined, for $h \in \mathbb{N}$, three sequences

$$\alpha(h) := \frac{q^h - 1}{q^{h+1} - 1}, \quad \gamma(h) := \frac{1}{\log_q \frac{1}{\alpha}} - \frac{1}{h}, \quad r(h) := \lfloor \gamma \log_q N \rfloor + 1,$$

such that

$$\alpha(h) \nearrow \frac{1}{q}, \quad \gamma(h) \nearrow 1, \quad r(h) \nearrow \lfloor \log_q N \rfloor + 1.$$

In this section, we choose as $h := h(N) \in \mathbb{N}$ an increasing function such that $h(N) \to \infty$ and $h^2 \in o(\log_q N)$; to fix ideas, one may think $h := \lfloor (\log_q N)^{\frac{1}{4}} \rfloor$ or $h := \lfloor \log_q \log_q N \rfloor$. So now α, γ and r are actually functions of N. Again, let $\delta \in]0, 1[$ and let

$$r' := \min\{i \in \mathbb{N} : m_{i+1} < (\alpha^i N)^{1-\delta} \text{ or } i = \ell'\}.$$

Recall that all computations performed in the previous section make sense only under the two assumptions

1. $r \leq \ell'$,
2. for all $i = 1, \ldots, r'$, $t \leq \frac{m_i}{2}$.

By definition of r', the second assumption is clearly true for sufficiently large N. The following lemma justifies the first.

Lemma 5.3.1. *Notation as above. Then, for sufficiently large N, $r \leq \ell'$.*

Proof. By definition of ℓ', if the matrix G_r formed by the last $m_r + \cdots + m_\ell$ columns of G has rank greater than $h + 1$ then $\ell' \geq r$. By corollary 5.2.2, $\operatorname{rk} G_r \geq \lfloor \log_q(N - (m_1 + \cdots + m_{r-1})) \rfloor + 1$, hence if $\lfloor \log_q(N - (m_1 + \cdots + m_{r-1})) \rfloor \geq h$ then $\ell' \geq r$. As $r \leq \gamma \log_q N + 1$ we have

$$\begin{aligned}
\lfloor \log_q(N - (m_1 + \cdots + m_{r-1})) \rfloor &\geq \log_q(N - (m_1 + \cdots + m_{r-1})) - 1 \geq \\
&\geq \log_q(N - (1 - \alpha^{r-1})N) - 1 = \\
&= (r-1) \log_q \alpha + \log_q N - 1 \geq \\
&\geq \gamma \log_q N \log_q \alpha + \log_q N - 1 = \\
&= (\gamma \log_q \alpha + 1) \log_q N - 1 = \\
&= -\frac{1}{h} \log_q \alpha \log_q N - 1.
\end{aligned}$$

Now note that, as $h^2 \in o(\log_q N)$,

$$\frac{-\frac{1}{h} \log_q \alpha \log_q N - 1}{h} \to +\infty,$$

hence for sufficiently large N

$$\lfloor \log_q(N - (m_1 + \cdots + m_{r-1})) \rfloor \geq -\frac{1}{h} \log_q \alpha \log_q N - 1 \geq h.$$

□

This means that, for sufficiently large N, everything we said in the previous section is true. In particular, the product dimension of C satisfies either (5.2.5) or (5.2.8). We recall them for convenience:

$$\widehat{k} \geq k + \frac{1-\delta}{\log_q \frac{1}{\alpha}} \left(\gamma - \frac{1}{2}\right) t \log_q^2 N - \frac{1-\delta}{2} t \log_q N - r(\tau + 1 + h), \tag{5.2.5}$$

$$\widehat{k} \geq k + \alpha^\delta N^{\frac{\delta}{h} \log_q \frac{1}{\alpha}} - \frac{q^h}{\alpha^{1-\delta} N^{\frac{1-\delta}{h} \log_q \frac{1}{\alpha}}}. \tag{5.2.8}$$

We claim that both of them imply theorem 5.0.3.

In particular, $\delta \in]0,1[$ is arbitrary, hence we can choose $\delta := \delta(N) := \frac{1}{g}$, where $g := g(N)$ is an increasing function such that $g(N) \to \infty$ and $gh \log_q \log_q N \in o(\log_q N)$; to fix ideas, one may think $g := \frac{\sqrt{\log_q N}}{h \log_q \log_q N}$.

Note that

$$\frac{1-\delta}{\log_q \frac{1}{\alpha}} \left(\gamma - \frac{1}{2}\right) \nearrow \frac{1}{2},$$

hence, for all $\varepsilon > 0$ and sufficiently large N,

$$\frac{1-\delta}{\log_q \frac{1}{\alpha}} \left(\gamma - \frac{1}{2}\right) \geq \frac{1}{2} - \varepsilon.$$

Moreover, as $r \in O(\log_q N)$ and $h^2 \in o(\log_q N)$,

$$-t\frac{1-\delta}{2} \log_q N - r(\tau + 1 + h) \in o(\log_q^2 N).$$

This proves that (5.2.5) implies theorem 5.0.3.

Now we claim that:

$$\frac{2\alpha^\delta N^{\frac{\delta}{h} \log_q \frac{1}{\alpha}}}{\log_q^2 N} \to +\infty, \tag{5.3.1}$$

$$\frac{q^h}{\alpha^{1-\delta} N^{\frac{1-\delta}{h} \log_q \frac{1}{\alpha}}} \to 0. \tag{5.3.2}$$

In particular, (5.3.1) implies that, for sufficiently large N, $\alpha^\delta N^{\frac{\delta}{h} \log_q \frac{1}{\alpha}} \geq \frac{1}{2} t \log_q^2 N$, hence if these claims are true then (5.2.8) implies theorem 5.0.3. We have

$$\lim_{N\to\infty} \frac{2\alpha^\delta N^{\frac{\delta}{h} \log_q \frac{1}{\alpha}}}{\log_q^2 N} = 2 \lim_{N\to\infty} \frac{N^{\frac{\delta}{h}}}{\log_q^2 N} = 2 \lim_{N\to\infty} \frac{q^{\frac{\delta}{h} \log_q N}}{q^{2 \log_q \log_q N}} =$$

$$= 2 \lim_{N\to\infty} q^{\frac{\delta}{h} \log_q N - 2 \log_q \log_q N}$$

and, as $gh \log_q \log_q N \in o(\log_q N)$,

$$\lim_{N\to\infty} \left(\frac{\delta}{h} \log_q N - 2 \log_q \log_q N\right) = \lim_{N\to\infty} \frac{\log_q N - 2gh \log_q \log_q N}{gh} = +\infty.$$

This implies (5.3.1). As to (5.3.2),

$$\lim_{N\to\infty} \frac{q^h}{\alpha^{1-\delta} N^{\frac{1-\delta}{h}\log_q \frac{1}{\alpha}}} = q \lim_{N\to\infty} \frac{q^h}{N^{\frac{1}{h}}} = q \lim_{N\to\infty} \frac{q^h}{q^{\frac{1}{h}\log_q N}} =$$
$$= q \lim_{N\to\infty} q^{h-\frac{1}{h}\log_q N}$$

and

$$\lim_{N\to\infty} \left(h - \frac{1}{h}\log_q N \right) = \lim_{N\to\infty} \frac{h^2 - \log_q N}{h} = -\infty.$$

This concludes the proof of theorem 5.0.3.

Bibliography

[1] Cascudo I.; Chen H.; Cramer R.; Xing C. (2009). Asymptotically Good Ideal Linear Secret Sharing with Strong Multiplication over *any* Fixed Finite Field. *CRYPTO*, pp. 466–486.

[2] Chen H.; Cramer R. (2006). Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computation over Small Fields. *CRYPTO*, pp. 521–536.

[3] Cover T.; Thomas J. (2006). *Elements of Information Theory*. Wiley-Interscience, second edition.

[4] Cramer R.; Damgaard I.; Maurer U. (2000). General Secure Multi-Party Computation from any Linear Secret-Sharing Scheme. *EUROCRYPT*.

[5] Garcia A.; Stichtenoth H.; Bassa A.; Beelen P. (2012). Towers of function fields over non-prime finite fields. http://arxiv.org/abs/1202.5922.

[6] Ihara Y. (1981). Some remarks on the number of rational points of algebraic curves over finite fields. *J. Fac. Sci. Tokyo*, **28**(3), 721–724.

[7] Ishai Y.; Kushilevitz E.; Ostrovsky R.; Prabhakaran M.; Sahai A.; Wullschleger J. (2011). Constant-Rate Oblivious Transfer from Noisy Channels. *CRYPTO*, pp. 667–684.

[8] MacWilliams F.; Sloane N. (1977). *The Theory of Error-Correcting Codes*. North-Holland.

[9] Massey J. (1993). Minimal Codewords and Secret Sharing. In *Sixth Joint Swedish-Russian Workshop on Information Theory*, pp. 276–279.

[10] Randriambololona H. (2012). Asymptotically good binary linear codes with asymptotically good self-intersection spans. http://arxiv.org/abs/1204.3057.

[11] Serre J.-P. (1985). Rational points on curves over finite fields notes of lectures at Harvard University.

[12] Shoup V. (2008). *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, second edition.

[13] Stichtenoth H. (1993). *Algebraic function fields and codes*. Springer, Heidelberg.

[14] van Lint J. (1999). *Introduction to Coding Theory*. Springer, Heidelberg.

[15] Yao A. (1982). Protocols for secure computations. In *Proceedings of the twenty-third annual IEEE Symposium on Foundations of Computer Science*, pp. 160–164.