Private Information Retrieval

Master's Thesis

Christian Jensen Lex

Spring 2021

Aalborg University
Mathematics

Mathematics
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Private Information Retrieval

Theme:
Private Information Retrieval

Project Period:
Spring Semester 2021

Project Group:
f21matspec_3

Participant(s):
Christian Jensen Lex

Supervisor(s):
Oliver W. Gnilke

Copies: 0

Page Numbers: 50

Date of Completion:
June 2, 2021

Abstract:

This master's thesis in mathematics aims to make an introduction to the subject of Private Information Retrieval (PIR) by introducing the main ideas of PIR as well as introducing a couple of promising PIR schemes which rates are compared with capacity achieving schemes. The PIR viewpoint taking in this thesis is that of information theoretic privacy.
The thesis presents coded distributed storage systems and how these should be considered in the PIR context. One of the primary tools in the PIR schemes of this thesis is Generalised Reed-Solomon (GRS) codes. It is shown that these have very compelling properties in the PIR setup.
A considerable amount of server-side computation must be carried out in the presented schemes. Hence, this thesis presents a scheme variant of one of the schemes that uses subfield subcodes of GRS codes (alternant codes) which reduces the computational complexity of the server response computation. Considerations towards a subfield subcode version of the second scheme are also presented.

Contents

Preface

This thesis is written by a tenth semester mathematics student at Aalborg University during the spring semester of 2021. The main theme of the project is *private information retrieval* with a coding theoretic perspective. Prerequisite knowledge required for reading the project is basic knowledge of abstract algebra, linear algebra, and coding theory.
References are denoted by a number, e.g. [1], corresponding to a number in the bibliography. Definitions, theorems, propositions, lemmas, corollaries and examples are numbered consecutively according to the chapter they are contained in. Figures and tables are numbered in a similar manner.

I would like to thank my supervisor Oliver W. Gnilke for his help and supervision during the writing of this thesis as well as the opportunity to co-author a paper with Oliver that some of this thesis builds upon and which can be found in the appendix of the thesis. His guidance has been most helpful when my focus has been steered in an unfruitful direction.

Aalborg University, June 2, 2021

Christian Jensen Lex
clex16@student.aau.dk

Reader's Guide

In this project the following notation is used:

- $\mathbb{N}$ denotes the natural numbers including 0. $\mathbb{N}_{>i}$ denotes the natural numbers larger than i.
- $\mathbb{Z}$ denotes the integers.
- $\mathbb{Q}$ denotes the rational numbers.
- $\mathbb{R}$ denotes the real numbers.
- $\mathbb{C}$ denotes the complex numbers.
- $\mathbb{F}_q$ denotes the finite field of size q.
- $\langle r \rangle$ denotes the ideal generated by r for $r \in R$ where R is a ring.
- All rings in this project are commutative and contain multiplicative identity 1.
- Vectors are denoted by boldface, e.g., $\boldsymbol{v} \in \mathbb{R}^n$.
- $\langle \boldsymbol{v}, \boldsymbol{w} \rangle$ denotes the dot product between vectors $\boldsymbol{v}$ and $\boldsymbol{w}$.
- $B_r(\boldsymbol{v})$ denotes the ball of radius r centered in the point $\boldsymbol{v}$.
- $\mathbf{1}$ denotes an appropriately length vector of all ones.
- For any $n \in \mathbb{N}_{>0}$ we denote the set $\{1, \ldots, n\}$ by $[n]$.
- G_C denotes a generator matrix for the linear code C.
- d_C denotes the minimum distance of the code C

Introduction

Suppose that some user wishes to access certain files on a database without giving any hints to the database as to what file was retrieved. An example of someone interested in such manner of accessing files could be a citizen of a country where access to parts of the internet is illegal or at least frowned upon.

In case there is only a single server present it is not hard to convince oneself that the only feasible choice for the user is to download the entire content of the server. Otherwise, the server would get at least some hint towards what file was requested. It is far from practical to download all files on a server since in any real world scenario a server would contain far to much data do download. Hence, this thesis considers storage systems where files are distributed onto several servers that does not communicate freely among each other. The user sends a query to each server that does not reveal the identity of the requested file, and the servers replies with a response that allows the user to reconstruct the requested file. This process loosely describes the concept of Private Information Retrieval (PIR).

PIR schemes were first described in [3] and can be divided into two categories: Information theoretic PIR and computational PIR, and [3] described PIR as the former where the latter was first introduced in [9]. Information theoretic PIR requires that no information about the requested file can be recovered from the query by a single server even with an unbounded computing capacity. Computational PIR focuses on making information about the requested file computationally infeasible to recover. A respective comparison from other more well-known parts of cryptography could be the information theoretically secure one time pad encryption against the computationally secure public key cryptosystems such as RSA or elliptic curve cryptography. This thesis focuses solely on the information theoretic schemes.

We will primarily assess the calibre of the PIR schemes by considering their PIR rate which is the ratio between information symbols downloaded and total symbols downloaded. In particular, we will ignore the cost of uploading and generating queries.

Many PIR schemes merely replicates the files onto each of the servers, but the schemes considered in this thesis uses length n error-correcting storage codes to distribute M files onto n servers. The first scheme considered is the one introduced in [6] which yields information theoretic privacy not only in the case of no communication between the servers, but also in the case that a subset of servers collude to receive indication of the file requested. The scheme does this by choosing queries by the use of a retrieval code D. By the use of Generalised Reed-Solomon (GRS) codes this scheme achieves a PIR rate of $(n - k - t + 1)/n$ where k is the dimension of the storage code and t is the number of colluding servers the scheme protects against. This scheme comes with a considerable computational cost at the servers. In the appendix a paper is found which tries to mend upon this by passing D to its subfield subcode leading to an improved response computation complexity. The idea of this paper is presented in this thesis as well.

The second scheme presented in this thesis allows for a subset of servers to either reply maliciously or be faulty without failure of file retrieval. Such servers are called

byzantine or unresponsive respectively. This scheme is that of [18] and is related to the scheme from [6]. It gives a PIR rate of $(n-k-t-2\beta-r+1)/n$ where β is the number of byzantine servers and r is the number of unresponsive servers that the scheme protects against. At last, it is considered how a subfield subcode PIR scheme can be derived from this scheme.

Private Information Retrieval

We will begin this section by presenting a simple two-server PIR scheme of [3]. This example enlightens the PIR-idea well since it is not hard to see why it is both correct and private at least when the two servers do not collude.

Example 2.1. Let $x^1, \dots, x^M \in \mathbb{F}_2$ denote the files, and let s_1 and s_2 denote two servers - the servers we wish to retrieve files from. In this scheme all the files $x^1, \dots, x^M$ are stored on both s_1 and s_2 as $\boldsymbol{x} = (x^1, \dots, x^M) \in (\mathbb{F}_2)^M$. Suppose we wish to retrieve the file $x^i, i \in [M]$. The scheme is as follows: From the space $(\mathbb{F}_2)^M$ we choose an element $\boldsymbol{d} \in (\mathbb{F}_2)^M$ uniformly at random and define $\boldsymbol{q} = \boldsymbol{d} + \boldsymbol{e}_i$. The query $\boldsymbol{d}$ is sent to the server s_1 and $\boldsymbol{q}$ is sent to s_2 to which they reply with

$$r_1 = \langle \boldsymbol{x}, \boldsymbol{d} \rangle, \text{ and} \qquad\qquad r_2 = \langle \boldsymbol{x}, \boldsymbol{q} \rangle = r_1 + x^i \tag{2.1}$$

respectively. Upon receiving r_1 and r_2 we get $r_1 + r_2 = x^i$, hence, we have recovered the correct file. Privacy follows since $\boldsymbol{q}$ as a random vector also has uniform distribution on $(\mathbb{F}_2)^M$.

It should be noted that this is merely a proof of concept since this scheme requires us to send M bits of information to each server. Hence, it would be better to ask just one of the servers to send every file - the exact case PIR tries to improve upon.

Example 2.1 clearly presents the parts of the PIR schemes that are of interest to us consists of: A process of constructing queries to the servers at random, a server side response calculation which is replied to the user and a reconstruction of the file using the responses from the servers. We will give a more precise definition of a PIR scheme inspired by these parts shortly.

In Example 2.1, every server stores a copy of the files $x^1, \dots, x^M$ and each file consists of one bit of information. From here on, we will consider a more general *distributed storage system* (DSS) strongly inspired by the DSS in [6]. This storage system is the only type of storage system we consider.

Definition 2.2. A distributed storage system $\mathscr{D}$ consists of the following parts:

i) M files $x^1, \dots, x^M \in (\mathbb{F}_{q^m})^{b \times k}$ each a matrix over $\mathbb{F}_{q^m}$.

ii) A linear $\mathbb{F}_{q^m}$ *storage code* C with parameters $[n, k, d_C]$ and generator matrix G_C.

iii) An encoding procedure

$$\begin{bmatrix} x^1 \\ \vdots \\ x^M \end{bmatrix} G_C = \begin{bmatrix} y^i \\ \vdots \\ y^M \end{bmatrix} = Y, \tag{2.2}$$

where the j^{th} column of Y is stored on the j^{th} server.

Superscripts will refer to files, subscripts to servers, and parenthesis to vector entries. E.g., $y^i = x^i G_C$ denotes the i^{th} encoded file, $\boldsymbol{y}_j^i$ the part of this encoded file on the j^{th} server, and $y_j^i(a)$ the a^{th} entry of this vector.

The scheme of Example 2.1 also uses this DSS. Indeed, the files $x^1, \ldots, x^M \in \mathbb{F}_2$ are stored using the binary $[2, 1, 2]$ repetition code C as

$$\begin{bmatrix} x^1 \\ \vdots \\ x^M \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} = \begin{bmatrix} x^1 & x^1 \\ \vdots & \vdots \\ x^M & x^M \end{bmatrix} = Y.$$

It is clear that one of these two servers can fail with the files still being recoverable. This is a special case of the following proposition.

Proposition 2.3. *The files $x^1, \ldots, x^M$ can be reconstructed from any choice of at least $n - d_C + 1$ columns of Y.*

We are now ready to give a definition of a PIR scheme.

Definition 2.4 (PIR Scheme)**.** Let $x^1, \ldots, x^M \in (\mathbb{F}_{q^m})^{b \times k}$ be files stored on n servers using an $[n, k, d_C]_{q^m}$ code C to encode the files as in (2.2). Then a PIR scheme $\mathscr{S}$ contains the following steps:

i) *Query construction.* For each file $x^j, j \in [M]$ we have a discrete random variable Q^j taking values on $\mathcal{Q}^i$ with density p_{Q^i}. To recover the i^{th} file x^i a query $q^i \in \mathcal{Q}^i$ is selected at random according to the density p_{Q^i}. The queries themselves are sets $q^i = \{\boldsymbol{q}_1^i, \ldots, \boldsymbol{q}_n^i\}$ of vectors

$$\boldsymbol{q}_j^i = (\boldsymbol{q}_j^{i1}, \ldots, \boldsymbol{q}_j^{im}), \text{ for } \boldsymbol{q}_j^{il} \in (\mathbb{F}_{q^m})^b; l = 1, \ldots, M \text{ and } j = 1, \ldots, n,$$

each realisations of a random vector Q_j^i with density $p_{Q_j^i}$. Then $\boldsymbol{q}_j^i$ is sent to the j^{th} server.

ii) *Response.* The response is determined at the server as the Euclidean inner products $r_j^i = \langle \boldsymbol{q}_j^i, \boldsymbol{y}_j \rangle$ where $\boldsymbol{y}_j$ denotes the transpose of the j^{th} column of Y. These are combined to be the response vector $\boldsymbol{r}^i = (r_1^i, \ldots, r_n^i) \in (\mathbb{F}_{q^m})^n$.

iii) *Iteration.* Steps *i*) and *ii*) are repeated until the file x^i can be reconstructed from the s response vectors $\boldsymbol{r}^i$.

iv) *Reconstruction.* The response vectors $\boldsymbol{r}^i$ are taken as input and are used to reconstruct x^i.

This definition is quite general and does not really touch upon the privacy considerations of the scheme. Sometimes a definition of correctness is given of PIR schemes, i.e., that the scheme does in fact fetch the correct file x^i. As is seen in step *iv*) of Definition 2.4 we consider PIR schemes to always be correct.

To define privacy of a PIR scheme the notion of *mutual information* (see e.g. [4]) of two random variables is convenient.

Definition 2.5 (Mutual Information)**.** Let X and Y be discrete random variables with densities p_X and p_Y which takes values on $\mathcal{X}$ and $\mathcal{Y}$ respectively. Furthermore, let $p_{X,Y}$ be the joint density of X and Y. Then

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{X,Y}(x,y) \log\left(\frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}\right) \tag{2.3}$$

is defined as the mutual information of X and Y.

Informally, the mutual information $I(X;Y)$ can be thought of as the reduction of uncertainty of X given Y and vice versa. Indeed, if X and Y are independent $p_{X,Y} = p_X p_Y$ and every term in (2.3) vanishes. Additionally, it is clear from (2.3) that $I(X;Y) \geq 0$.

Definition 2.6 (Collusion Protection)**.** A PIR scheme $\mathscr{S}$ is said to protect against t colluding servers if for every t-sized index set $T = \{j_1, \ldots, j_t\} \subseteq [n]$ we have

$$I(Q_T^i; i) = 0,$$

where Q_T^i denotes the random variable with joint density $p_{Q_T^i} = p_{Q_{j_1}^i,\ldots,Q_{j_t}^i}$ from which queries are sampled during the entire PIR scheme $\mathscr{S}$.

This means that any choice of t or fewer (even computationally unbounded) servers will get no indication as to what file x^i was requested. If a scheme $\mathscr{S}$ protects against t colluding servers we will also say that $\mathscr{S}$ is t-private. From now on, we assume that any PIR scheme is at least 1-private.

For a scheme $\mathscr{S}$ we define the *rate* of $\mathscr{S}$.

Definition 2.7 (Rate)**.** The rate of a PIR scheme is bk/ns.

The retrieval rate is the ratio between the valuable downloaded information and the total downloaded information. The rate is important when comparing PIR schemes both against other schemes and certain PIR *capacities*.

2.1 PIR Capacity

The first bound on the rate is from [1] which generalised the PIR capacity on private information retrieval with no colluding servers and repetition coded databases (i.e. PIR schemes with a repetition code as a storage code as Example 2.1) given in [16] to MDS storage codes. But first, we give a precise definition of capacity. The capacity depends on the DSS, hence in this regard, we call a PIR scheme using a DSS $\mathscr{D}$ a $\mathscr{D}$-PIR scheme.

Definition 2.8 (Capacity)**.** Consider a DSS $\mathscr{D}$. Then the PIR capacity of this DSS is defined as

$$K = \sup\{R \mid R \text{ is the rate of a } \mathscr{D}\text{-PIR scheme}\}.$$

No capacities exits for both arbitrarily coded databases and arbitrary collusion patterns, but capacities exist for repetition coded databases with collusion and coded databases with no collution; the latter of which we present here.

Theorem 2.9. *Let C be an $[n, k, d_C]_{q^m}$ storage code $(k < n)$, and let $\mathscr{D}$ be a DSS using this storage code to encode M files. Then the PIR capacity K is given by*

$$K = \frac{1 - \frac{k}{n}}{1 - \left(\frac{k}{n}\right)^M} = \frac{1}{1 + \frac{k}{n} + \cdots + \left(\frac{k}{n}\right)^{M-1}}. \tag{2.4}$$

We can see from (2.4) that the capacity K only depends on the DSS $\mathscr{D}$ insofar as it depends on the code parameters n and k of the storage code C as well as it depends on the number of encoded files M. It is also clear that K is the capacity when there is no server collusion since any scheme protects against this case by assumption.

Observe, that k/n is the code rate of C. Thus, the PIR capacity decreases when we increase the code rate and vice versa.

In [17] a capacity for t-private PIR schemes is given for $t > 1$ using a repetition coded DSS. In light of Definition 2.8 we give a more general definition of PIR capacity.

Definition 2.10 (Collusion Capacity)**.** Let $\mathscr{S}$ be a DSS. Then the t-collusion PIR capacity K_t is defined as

$$K_t = \sup\{R \mid R \text{ is the rate of a } t\text{-private } \mathscr{D}\text{-PIR scheme}\}.$$

For a given DSS K_t is not known in general. However, as stated previously, if we force C to be a repetition code we get the following capacity.

Theorem 2.11. *Let C be an $[n, 1, n]$ repetition code, and let $\mathscr{D}$ be a DSS using this storage code to encode M files. Then the t-collusion PIR capacity K_t $(t < n)$ is given by*

$$K_t = \frac{1 - \frac{t}{n}}{1 - \left(\frac{t}{n}\right)^M} = \frac{1}{1 + \frac{t}{n} + \cdots + \left(\frac{t}{n}\right)^{M-1}}.$$

As well as the capacity K is a decreasing function of the code dimension k, K_t is a decreasing function of the collution protection t. If we consider the case $t = 1$ we get exactly the situation in 2.9 with repetition coding, that is, $k = 1$. As we would expect we get $K_1 = K$. It is not an issue to assume that $t < n$ since $t = n$ would yield $K_t = 1/M$ and all files must be downloaded to ensure privacy.

The proofs of these capacities both consists of a proof of an upper bound on the rate as well as a PIR scheme that achieves this rate. Details can be found in [1] and [17] for Theorem 2.9 and Theorem 2.11 respectively.

A Private Information Retrieval Scheme

In this section we introduce the PIR scheme of [6].

3.1 Preliminaries

This scheme depends on the *star product* of a pair of codes, one of which will be the storage code C and the other a certain *retrieval code* D.

Definition 3.1 (Star Product)**.** Let C, D be length n codes over $\mathbb{F}_{q^m}$. Then the star product of C and D is defined as the subspace

$$C \star D = \text{span}\left\{(c_1 d_1, \ldots, c_n d_n) \mid (c_1, \ldots, c_n) \in C, (d_1, \ldots, d_n) \in D\right\}.$$

We will also use $\star$ to denote the entry-wise product of two vectors $\boldsymbol{u} \star \boldsymbol{v} = (u_1 v_1, \ldots, u_n v_n)$. To ensure the correctness of the scheme we will need the notion of an information set of a code.

Definition 3.2 (Information Set)**.** Let C be an $[n, k, d_C]$ code. Then a subset $K \subseteq [n]$ is called an information set if the natural projection of C onto the coordinates of C in K; $C \to C_K$ is a bijection.

3.2 The Scheme

Assume that we have some given DSS $\mathscr{D}$ with $\mathbb{F}_{q^m}$ storage code C that has parameters $[n, k, d_C]$. This is the DSS we consider throughout this scheme. We will call this scheme $\mathscr{S}_{FGHK}$ as an homage to the authors of [6].

Let $D \subseteq (\mathbb{F}_{q^m})^n$ be some other linear code which we will call the retrieval code. We adopt the notation of [6], i.e, $i \in [M]$ will denote the index of the file we are interested in retrieving, and $c := d_{C\star D} - 1$. This is the amount of symbols we can download in each iteration of the scheme. The number of rows in each file b as well as the number of iterations s of the algorithm should be considered as parameters which are adjusted according to the DSS and the PIR scheme. In this scheme, we fix

$$s = \frac{\text{lcm}(c, k)}{c}, \qquad\qquad b = \frac{\text{lcm}(c, k)}{k},$$

as well as we define a set $J := [\max\{c, k\}]$. This will be the index set of servers (after perhaps a rearrangement of servers) from which symbols are downloaded. We define $g := k/s = c/b$ as the number of symbols that is downloaded from each row of the encoded file y^i per iteration of the scheme. This ensures that after s iterations we have downloaded exactly $sg = k$ distinct symbols from each row of y^i. At last, we define

$$J_u^a \subseteq J, \qquad\qquad u \in [s],\ a \in [b],$$

which is the server index set in the u^{th} iteration such that $y_j^i(a)$ is downloaded from the servers $j \in J_u^i$.

We will now consider the steps of this PIR scheme. The steps are presented as in Definition 2.4.

Query Construction: A query is constructed by choosing Mb codewords $\boldsymbol{d}^{l,a} = (d^{l,a}(1), \ldots, d^{l,a}(n))$ of D uniformly at random where $l \in [M]$ and $a \in [b]$. For each $j \in [n]$ we define a vector $\boldsymbol{d}_j$ by

$$\boldsymbol{d}_j^l = \left(d^{l,1}(j), \ldots, d^{l,b}(j)\right) \in (\mathbb{F}_{q^m})^b, \qquad \boldsymbol{d}_j = \left(\boldsymbol{d}_j^1, \ldots, \boldsymbol{d}_j^M\right) \in (\mathbb{F}_{q^m})^{Mb}.$$

A subset $J_1 = [c] \subseteq J$ is defined and is partitioned according to the rows of the files:

$$J_1^1 = [g]\,, J_1^2 = [2g] \setminus [g], \ldots, J_1^b = [gb = c] \setminus [(b-1)g]. \tag{3.1}$$

Remember, we wish to fetch the i^{th} file. Hence, the j^{th} query $\boldsymbol{q}_j^i$ is defined by

$$\boldsymbol{q}_j^i = \begin{cases} \boldsymbol{d}_j + \boldsymbol{e}_{b(i-1)+a} & \text{if } j \in J_1^a, \\ \boldsymbol{d}_j & \text{if } j \notin J_1. \end{cases} \tag{3.2}$$

Response Computation: The j^{th} entry of the response vector is then determined as

$$r_j^i = \langle \boldsymbol{q}_j^i, \boldsymbol{y}_j \rangle. \tag{3.3}$$

Iteration: Query construction and response computation are iterated s times by applying a length g cyclic shift to the partition (3.1) within J. That is, if the a^{th} index set for the $u-1^{\text{th}}$ iteration is $J_{u-1}^a = \{j_1, \ldots, j_g\}$ then

$$J_u^a = \{j_1 + g \pmod{|J|}, \ldots, j_g + g \pmod{|J|}\}.$$

Then a new superset J_u is defined as $J_u = J_u^1 \cup \cdots \cup J_u^b$ and the u^{th} queries can be defined using this set as in (3.2) by

$$\boldsymbol{q}_j^i = \begin{cases} \boldsymbol{d}_j + \boldsymbol{e}_{b(i-1)+a} & \text{if } j \in J_u^a, \\ \boldsymbol{d}_j & \text{if } j \notin J_u. \end{cases}$$

Reconstruction: A parity check matrix H for $C \star D$ is found and data from the u^{th} iteration is reconstructed as

$$\boldsymbol{r}^i H^T = \left(\boldsymbol{0}_{(u-1)c}, y_{(u-1)c+1}^i(1), \ldots, y_{uc}^i(b), \boldsymbol{0}_{n-uc}\right) H^T, \tag{3.4}$$

where $\boldsymbol{0}_l$ denotes the length l zero vector. Since $c = d_{C\star D} - 1$ any choice of c columns of H are linearly independent. Hence, $y_{(u-1)c+1}^i(1), \ldots, y_{uc}^i(b)$ can be recovered from the matrix-vector product (3.4).

It is still not entirely clear whether this is a PIR scheme in the sense that it retrieves the correct file x^i.

Lemma 3.3. *The reconstruction process on the data from the u^{th} iteration in the scheme recovers the c symbols $y_{(u-1)c+1}^i(1), \ldots, y_{uc}^i(b)$, i.e., the equality (3.4) is correct.*

Proof. To ease the notation we consider the case $u = 1$, since the argument will generalise easily to the case $u > 1$. Hence, consider the set J_1 and the partition (3.1). Suppose that $j \notin J_1$. Then

$$r_j^i = \langle \boldsymbol{q}_j^i, \boldsymbol{y_j} \rangle = \langle \boldsymbol{d}_j, \boldsymbol{y}_j \rangle = \sum_{l=1}^{M} \langle \boldsymbol{d}_j^l, \boldsymbol{y}_j^l \rangle = \sum_{l=1}^{M} \sum_{a=1}^{b} d^{l,a}(j) y_j^l(a).$$

Similarly, in case that $j \in J_1^{a_0}$ for some $a_0 \in [b]$ we get

$$r_j^i = \langle \boldsymbol{q}_j^i, \boldsymbol{y_j} \rangle = \langle \boldsymbol{d}_j + \boldsymbol{e}_{b(i-1)+a_0}, \boldsymbol{y}_j \rangle = \sum_{l=1}^{M} \sum_{a=1}^{b} d^{l,a}(j) y_j^l(a) + y_j^i(a_0).$$

We adjoin these to the total response vector $\boldsymbol{r}^i$ for the 1st iteration as

$$\begin{aligned} \boldsymbol{r}^i &= \left(r_1^i, \ldots, r_n^i\right) \\ &= \underbrace{\sum_{l=1}^{M} \sum_{a=1}^{b} \left(d^{l,a}(1) y_1^l(a), \ldots, d^{l,a}(n) y_n^l(a)\right)}_{\in C \star D} \\ &\quad + \left(y_1^i(1), \ldots, y_g^i(1), \ldots, y_{g(b-1)}^i(b), \ldots, y_c^i(b), \mathbf{0}_{n-c}\right). \end{aligned}$$

Now, by multiplying $\boldsymbol{r}^i$ with H^T for H a parity check matrix for $C \star D$ we get

$$\boldsymbol{r}^i H^T = \mathbf{0}_n + \left(y_1^i(1), \ldots, y_g^i(1), \ldots, y_{g(b-1)}^i(b), \ldots, y_c^i(b), \mathbf{0}_{n-c}\right) H^T,$$

and as mentioned previously, since $c = d_{C\star D} - 1$ any choice of c columns are linearly independent, so the c symbols $y_1^i(1), \ldots, y_g^i(1), \ldots, y_{g(b-1)}^i(b), \ldots, y_c^i(b)$ can be recovered. ■

Theorem 3.4. *Let C be an $[n, k, d_C]$ storage code, and let D be a retrieval code. If either*

i) $d_{C\star D} - 1 \leq k$, or

ii) $J \subseteq [n]$ exists of size $\max\{d_{C\star D} - 1, k\}$ such that every subset of size k is an information set of C,

then the scheme $\mathscr{S}_{FGHK}$ is a PIR scheme with rate $(d_{C\star D} - 1)/n$.

Proof. By Lemma 3.3 after s iterations of the scheme we have received k symbols from each row $\boldsymbol{y}^{i,a}$ of y^i. We need to ensure that the entire file can be recovered from these symbols. In case *i)* holds, choose for $J \subseteq [n]$ any k-sized information set. Since the symbols of y^i will come from the columns in J, x^i can be recovered.

In case *ii)* holds, for each row $\boldsymbol{y}^{i,a}$ we have k symbols from the columns of some index set $K_a \subseteq J$ of size k, which by assumption is an information set. Thus, x^i can be recovered in this case as well.

By Definition 2.7 the rate of this scheme is

$$\frac{bk}{ns} = \frac{k\frac{\mathrm{lcm}(c,k)}{k}}{n\frac{\mathrm{lcm}(c,k)}{c}} = \frac{d_{C\star D}-1}{n}.$$

■

Theorem 3.5. *The PIR scheme $\mathscr{S}_{FGHK}$ is $d_{D^\perp}-1$-private*

Proof. Let $T = \{j_1, \ldots, j_t\} \subseteq [n]$ be a size $t \leq d_{D^\perp} - 1$ index set as in Definition 2.6. Since $t \leq d_{D^\perp} - 1$ every choice of t columns of the generator matrix G_D are linearly independent and D restricted to T, D_T, is all of $(\mathbb{F}_{q^m})^t$. The queries in the scheme are constructed by choosing the vectors $\boldsymbol{d}^{l,a} = \big(d^{l,a}(1), \ldots, d^{l,a}(n)\big)$ uniformly at random from D and by these constructing the matrix

$$M_d = \begin{bmatrix} d^{1,1}(1) & \ldots & d^{1,b}(1) & \ldots & d^{M,1}(1) & \ldots & d^{M,b}(1) \\ d^{1,1}(2) & \ldots & d^{1,b}(2) & \ldots & d^{M,1}(2) & \ldots & d^{M,b}(2) \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ d^{1,1}(n) & \ldots & d^{1,b}(n) & \ldots & d^{M,1}(n) & \ldots & d^{M,b}(n) \end{bmatrix} \tag{3.5}$$

The j^{th} row of M_d defines the vector $\boldsymbol{d}_j \in (\mathbb{F}_{q^m})^{Mb}$. Since $D_{\{j\}}$ forms all of $\mathbb{F}_{q^m}$ and the elements $\boldsymbol{d}^{l,a}$ are chosen uniformly at random from D, the entry $d^{l,a}(j)$ must have uniform distribution on $\mathbb{F}_{q^m}$. Hence, $\boldsymbol{d}_j$ is uniformly distributed on $(\mathbb{F}_{q^m})^{Mb}$. Using that D_T is all of $(\mathbb{F}_{q^m})^t$ the joint distribution of $\{\boldsymbol{d}_j \mid j \in T\}$ is the uniform distribution on $\big((\mathbb{F}_{q^m})^{Mb}\big)^t$. Using these uniformly distributed vectors, the queries are constructed as

$$\{\boldsymbol{q}^i_{j_1}, \ldots, \boldsymbol{q}^i_{j_t}\} = \{\boldsymbol{d}_j + \boldsymbol{e}_{f(i,j)} \mid j \in T \cap J\} \cup \{\boldsymbol{d}_j \mid j \in T \backslash J\},$$

where $f(i,j)$ indexes $\boldsymbol{e}_{f(i,j)}$ appropriately as in (3.2). If $\boldsymbol{q}^i_j = \boldsymbol{d}_j + \boldsymbol{e}_{f(i,j)}$ then $\boldsymbol{q}^i_j$ will also be uniformly distributed since the uniform distribution is invariant under translation. Hence, the distribution of $\{\boldsymbol{q}^i_{j_1}, \ldots, \boldsymbol{q}^i_{j_t}\}$ is a uniform distribution and is thus independent of i. Thus, during one iteration we have $I(\boldsymbol{q}^i_{j_1}, \ldots, \boldsymbol{q}^i_{j_t}, i) = 0$. The vectors $\boldsymbol{d}^{l,a}$ chosen uniformly at random from D for one iteration is chosen independent from each other iteration. If we by Q^i_T denote the random variable with joint distribution of all the queries sent to servers in T during the iterations of the scheme. This is uniform on $\big((\mathbb{F}_{q^m})^{Mb}\big)^{ts}$. Hence, we get $I(Q^i_T, i) = 0$. ■

In the case that we have $|T| = t > d_{D^\perp} - 1$, potentially, the columns of G_D indexed by T are linearly dependent, and D_T will be a proper subspace of $(\mathbb{F}_{q^m})^t$. Restricting a column of (3.5) to T and adding $\boldsymbol{e}_{f(i,j)}$ will therefore in some cases lie outside D_T, hence, for the mutual information we have $I(Q^i_T, i) > 0$.

3.3 PIR Using Generalised Reed-Solomon Codes

We have now described the scheme, but said nothing on how the codes C and D should be chosen such that this scheme is as efficient as possible. In [18] it is described how

MDS codes can be chosen for the storage code C to good effect. We will also consider MDS coded databases - in particular databases that use *generalised Reed-Solomon codes* (GRS codes) as storage codes.

Definition 3.6 (GRS Code)**.** Let $\mathbb{F}_{q^m}[X]^k$ denote the polynomials over $\mathbb{F}_{q^m}$ of degree less than k. Furthermore, let $\boldsymbol{v} \in (\mathbb{F}_{q^m}^*)^n$, $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m})^n$ for $\alpha_i \neq \alpha_j$ when $i \neq j$, and $k \leq n$. Then a generalised Reed-Solomon code is defined as

$$\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}) = \left\{ (v_1 f(\alpha_1), \ldots, v_n f(\alpha_n)) \;\middle|\; f \in \mathbb{F}_{q^m}[X]^k \right\}.$$

It is clear that $n \leq q^m$. We will call the vector $\boldsymbol{\alpha}$ the *support* of the code, and we will denote codewords of a GRS code by the polynomial which generates it. More precisely, for $f \in \mathbb{F}_{q^m}[X]^k$ we denote a codeword $\boldsymbol{f} \in \mathrm{GRS}_k(\boldsymbol{v}, \boldsymbol{\alpha})$ associated through the evaluation homomorphism $\mathrm{ev}_{\boldsymbol{v},\boldsymbol{\alpha}} : \mathbb{F}_{q^m}[X]^k \to (\mathbb{F}_{q^m})^n$ given by

$$\mathrm{ev}_{\boldsymbol{\alpha},\boldsymbol{v}}(f) = (v_1 f(\alpha_1), \ldots, v_n f(\alpha_n)). \tag{3.6}$$

Proposition 3.7. *Generalised Reed-Solomon codes are linear MDS codes with parameters* $[n, k, n-k+1]$.

Proof. Linearity follows immediately since for $f, g \in \mathbb{F}_{q^m}[X]^k$ and $a, b \in \mathbb{F}_{q^m}$ we have

$$\begin{aligned} &a(v_1 f(\alpha_1), \ldots, v_n f(\alpha_n)) + b(v_1 g(\alpha_1), \ldots, v_n g(\alpha_n)) \\ =&(v_1(a+b)[f(\alpha_1) + g(\alpha_1)], \ldots, v_n(a+b)[f(\alpha_n) + g(\alpha_n)]), \end{aligned}$$

and $(a+b)[f(X)+g(X)] = (a+b)(f+g)(X) \in \mathbb{F}_{q^m}[X]^k$. The dimension follows directly from the evaluation homomorphism (3.6) since the dimension of $\mathbb{F}_{q^m}[X]^k$ is k. Two distinct codewords $\boldsymbol{f}, \boldsymbol{g} \in \mathrm{GRS}_k(\boldsymbol{v}, \boldsymbol{\alpha})$ can agree in at most $k-1$ positions since an interpolation of $f-g$ in k or more zeroes yields $f-g=0$ by the degree of $f-g$. Hence, the minimum distance is given by $n-k+1$. ∎

Regarding Theorem 3.5 GRS codes have a very convenient property, namely that they are closed under taking the dual.

Proposition 3.8. *For a length n GRS code $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ we have*

$$\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})^\perp = \mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \tilde{\boldsymbol{v}}),$$

where

$$\tilde{\boldsymbol{v}} = \left(\frac{1}{v_1 \prod_{i \neq 1}(\alpha_1 - \alpha_i)}, \ldots, \frac{1}{v_n \prod_{i \neq n}(\alpha_n - \alpha_i)} \right). \tag{3.7}$$

Proof. Let $\boldsymbol{f} \in \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ and let $\boldsymbol{g} \in \mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \tilde{\boldsymbol{v}})$ for $\tilde{\boldsymbol{v}}$ as in (3.7). By Lagrange interpolating the points $(f(\alpha_1)g(\alpha_1), \ldots, f(\alpha_n)g(\alpha_n))$ we can recover fg since it is of degree at most $n-k+k-2 = n-2$. Hence,

$$fg = \sum_{i=1}^{n} f(\alpha_i) g(\alpha_i) \prod_{j \neq i} \frac{X - \alpha_j}{\alpha_i - \alpha_j}. \tag{3.8}$$

As we have just argued, the coefficient of X^{n-1} of fg must be zero. Combining this with (3.8) yields for the X^{n-1}th coefficient:

$$\begin{aligned}0 &= \sum_{i=1}^{n} f(\alpha_i)\frac{g(\alpha_i)}{\prod_{j\neq i}(\alpha_i-\alpha_j)}\\ &= \sum_{i=1}^{n} v_i f(\alpha_i)\frac{g(\alpha_i)}{v_i\prod_{j\neq i}(\alpha_i-\alpha_j)}\\ &= \sum_{i=1}^{n} v_i f(\alpha_i)\tilde{v_i} g(\alpha_i)\\ &= \langle \boldsymbol{f}, \boldsymbol{g}\rangle.\end{aligned}$$

Since all pairs of elements $\boldsymbol{f} \in \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}), \boldsymbol{g} \in \mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \tilde{\boldsymbol{v}})$ are orthogonal and the dimensions add up to n, the two codes must be dual. ■

The idea in the scheme presented in this section will be to choose C and D in $\mathscr{S}_{FGHK}$ both as GRS codes. Hence, we must consider the star product of two GRS codes.

Proposition 3.9. *Let* $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u})$ *and* $\mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v})$ *be two length* n *GRS codes with the same support. Then*

$$\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u}) \star \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v}) = \mathrm{GRS}_{\min\{k+l-1,n\}}(\boldsymbol{\alpha}, \boldsymbol{u}\star\boldsymbol{v}).$$

Proof. The inclusion $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u}) \star \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v}) \subseteq \mathrm{GRS}_{\min\{k+l-1,n\}}(\boldsymbol{\alpha}, \boldsymbol{u}\star\boldsymbol{v})$ follows by considering two codewords $\boldsymbol{f} \in \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u})$ and $\boldsymbol{g} \in \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v})$. Then

$$\boldsymbol{f}\star\boldsymbol{g} = (u_1v_1(fg)(\alpha_1), \ldots, u_nv_n(fg)(\alpha_n)) \tag{3.9}$$

In case $k+l-1 \leq n$ the polynomial fg will be in $\mathbb{F}_{q^m}[X]^n$ and $\boldsymbol{f}\star\boldsymbol{g} \in \mathrm{GRS}_{\min\{k+l-1,n\}}(\boldsymbol{\alpha}, \boldsymbol{u}\star\boldsymbol{v})$ by (3.9). In case $k+l-1 > n$ the polynomial fg will not be in $\mathbb{F}_{q^m}[X]^n$, but a unique polynomial in $\mathbb{F}_{q^m}[X]^n$ passing through the points of $\mathrm{ev}_{\boldsymbol{\alpha},\mathbf{1}}(fg)$ exists. Hence, $\boldsymbol{f}\star\boldsymbol{g} \in \mathrm{GRS}_{\min\{k+l-1,n\}}(\boldsymbol{\alpha}, \boldsymbol{u}\star\boldsymbol{v})$ in this case as well.

For the other inclusion WLOG we assume that $\kappa := k+l-1 \leq n$ and we consider the canonical basis for $\mathbb{F}_{q^m}[X]^\kappa$ given by $1, X, \ldots, X^{\kappa-1}$. Under the evaluation mapping (3.6) this yields a basis for $\mathrm{GRS}_{k+l-1}(\boldsymbol{\alpha}, \boldsymbol{u}\star\boldsymbol{v})$ given by

$$\begin{aligned}\mathrm{ev}_{\boldsymbol{\alpha},\boldsymbol{u}\star\boldsymbol{v}}(1) &= (u_1v_1, \ldots, u_nv_n),\\ \mathrm{ev}_{\boldsymbol{\alpha},\boldsymbol{u}\star\boldsymbol{v}}(X) &= (u_1v_1\alpha_1, \ldots, u_nv_n\alpha_n),\\ &\vdots\\ \mathrm{ev}_{\boldsymbol{\alpha},\boldsymbol{u}\star\boldsymbol{v}}(X^{\kappa-1}) &= (u_1v_1\alpha_1^{\kappa-1}, \ldots, u_nv_n\alpha_n^{\kappa-1}).\end{aligned}$$

Any such basis element can be written as

$$(u_1v_1\alpha_1^\mu, \ldots, u_nv_n\alpha_n^\mu) = (u_1\alpha_1^a, \ldots, u_n\alpha_n^a) \star (v_1\alpha_1^b, \ldots, v_n\alpha_n^b),$$

for $\mu < k + l - 1$, $a < k$, $b < l$, and $a + b = \mu$. Hence, $(u_1 v_1 \alpha_1^\mu, \ldots, u_n v_n \alpha_n^\mu) \in \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u}) \star \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v})$ and

$$\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u}) \star \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v}) \subseteq \mathrm{GRS}_{\min\{k+l-1,n\}}(\boldsymbol{\alpha}, \boldsymbol{u} \star \boldsymbol{v}).$$

■

By choosing the storage code C of our DSS as a GRS code GRS_k we can find a retrieval codes that yields collusion protection for $1 \leq t \leq n - k$ servers for the scheme $\mathscr{S}_{FGHK}$.

Theorem 3.10. *For a DSS $\mathscr{D}$ with a length n storage code $C = \mathrm{GRS}_k$ a GRS retrieval code D exists such that the rate of $\mathscr{S}_{FGHK}$ using $\mathscr{D}$ and D is*

$$\frac{n - (k + t - 1)}{n}$$

and such that $\mathscr{S}_{FGHK}$ is t-private for all t satisfying $1 \leq t \leq n - k$.

Proof. Let $D = \mathrm{GRS}_t(\boldsymbol{\alpha}, \boldsymbol{u})$ (the choice of $\boldsymbol{u} \in (\mathbb{F}_{q^m}^*)^n$ is not important) be the retrieval code of the scheme. Then by Proposition 3.9 we have

$$C \star D = \mathrm{GRS}_{k+t-1}(\boldsymbol{\alpha}, \boldsymbol{v} \star \boldsymbol{u}).$$

This code has minimum distance $(n - (k + t - 2))$ by Proposition 3.7. Furthermore,

$$D^\perp = \mathrm{GRS}_{n-t}(\boldsymbol{\alpha}, \tilde{\boldsymbol{u}})$$

where $\tilde{\boldsymbol{u}}$ is as in (3.7). Again by Proposition 3.7, $D^\perp$ has minimum distance $t + 1$. Now, by applying Theorems 3.4 and 3.5 we get rate $(n - (k + t - 1))/n$ and t-privacy for the scheme. Note that Theorem 3.4 applies since $C^\perp$ has minimum distance $k + 1$. Hence, any choice of k servers of $\mathscr{D}$ will result in an information set of C. Indeed, G_C is a parity check matrix for $C^\perp$ and any choice of k columns of G_C must be linearly independent. ■

In this scheme we are interested in making the minimum distance of $C \star D$ as large as possible since this results in an improved rate. The following proposition is a Singleton-like bound on star products of codes as presented in [14].

Proposition 3.11. *Let C, D be codes over $\mathbb{F}_{q^m}$ with parameters $[n, k_C, d_C]$ and $[n, k_D, d_D]$ respectively. Then the minimum distance of $C \star D$ satisfies*

$$d_{C \star D} \leq \max\{1, n - k_C - k_D + 2\}. \tag{3.10}$$

Proof. By $\mathrm{Supp}(C)$ we denote the index set $\mathrm{Supp}(C) = \{i \mid \exists \boldsymbol{c} \in C : c_i \neq 0\}$. We assume that $\mathrm{Supp}(C) = \mathrm{Supp}(D) = [n]$. If not, we can consider the codes $C_{\mathrm{Supp}(C) \cap \mathrm{Supp}(D)}$ and $C_{\mathrm{Supp}(C) \cap \mathrm{Supp}(D)}$. This will clearly not change the minimum distance on the LHS of (3.10) and the dimensions k_C and k_D on the RHS of (3.10) will only decrease.

By considering a minimum weight word $\boldsymbol{d} \in D$ and any word $\boldsymbol{c} \in C$ with $\mathrm{Supp}(\boldsymbol{c}) \cap \mathrm{Supp}(\boldsymbol{d}) \neq \emptyset$ we get $1 \leq w(\boldsymbol{c} \star \boldsymbol{d}) \leq d_D$. Hence, $d_{C\star D} \leq d_D$ and by the Singleton bound we have

$$k_D \leq n - d_{C\star D} + 1. \tag{3.11}$$

Now, let $\tilde{C} = (C \star D)^{\perp}$, and let $\nu = \min\{n, k_D + d_{\tilde{C}^{\perp}} - 2\}$. Observe that $\nu - k_D + 1 \leq d_{\tilde{C}^{\perp}} - 1$ so any choice of $\nu - k_D + 1$ columns of a generator matrix of $\tilde{C}$ will be linearly independent. This means that for any size $\nu - k_D$ index set J of $\tilde{C}$ there is a codeword $\boldsymbol{v} \in \tilde{C}$ with $j_0 \notin J$ such that $v_{j_0} \neq 0$ and $v_j = 0$ for all $j \in J$. Choose a systematic generator matrix for D as $G = [I_{k_D} \mid A]$, and for $i \in [k_D]$ consider the i^{th} row $\boldsymbol{w}$ of G. Let $J = [\nu] \backslash [k_D]$ and pick a codeword $\boldsymbol{v} \in \tilde{C}$ that is nonzero in the i^{th} entry and zero in J. Then $\boldsymbol{z} = \boldsymbol{w} \star \boldsymbol{v}$ is nonzero in the i^{th} entry and zero in J. Doing this for each $i \in [k_D]$ yields k_D linearly independent elements of $D \star \tilde{C}$.

Now, suppose that $i \in [\nu] \backslash [k_D]$ and let i' be the index of a row $\boldsymbol{w}$ of G that is nonzero in this entry. We let $\boldsymbol{v} \in \tilde{C}$ be nonzero in the i^{th} entry and zero over the index set $J = \{i'\} \cup ([\nu] \backslash ([k_D] \cup \{i\}))$. Then $\boldsymbol{z} = \boldsymbol{w} \star \boldsymbol{v}$ form $\nu - k_D$ linearly independent vectors in $D \star \tilde{C}$ also independent of previously defined vectors. This gives us in total ν linearly independent vectors in $D \star \tilde{C}$ and

$$\dim(D \star \tilde{C}) \geq \nu = \min\{n, k_D + d_{\tilde{C}^{\perp}} - 2\}. \tag{3.12}$$

For the code $(D \star \tilde{C})$ we have

$$(D \star \tilde{C})^{\perp} = (D \star (C \star D)^{\perp})^{\perp} = C. \tag{3.13}$$

Indeed, consider $\boldsymbol{c} \in C$ and $\boldsymbol{v} \in D \star \tilde{C}$. By definition $\boldsymbol{v}$ is some linear combination $\boldsymbol{v} = \sum_i a_i(\boldsymbol{d}_i \star \tilde{\boldsymbol{c}}_i)$ where $a_i \in \mathbb{F}_{q^m}, \boldsymbol{d}_i \in D$, and $\tilde{\boldsymbol{c}}_i \in \tilde{C}$. Then we get

$$\begin{aligned}
\langle \boldsymbol{c}, \boldsymbol{v} \rangle &= \langle \boldsymbol{c}, \sum_i a_i(\boldsymbol{d}_i \star \tilde{\boldsymbol{c}}_i) \rangle = \sum_i a_i \langle \boldsymbol{c}, \boldsymbol{d}_i \star \tilde{\boldsymbol{c}}_i \rangle \\
&= \sum_i a_i \langle \mathbf{1}, \boldsymbol{c} \star (\boldsymbol{d}_i \star \tilde{\boldsymbol{c}}_i) \rangle = \sum_i a_i \langle \mathbf{1}, (\boldsymbol{c} \star \boldsymbol{d}_i) \star \tilde{\boldsymbol{c}}_i \rangle \\
&= \sum_i a_i \langle (\boldsymbol{c} \star \boldsymbol{d}_i), \tilde{\boldsymbol{c}}_i \rangle = \mathbf{0},
\end{aligned}$$

since $\tilde{\boldsymbol{c}}_i \in \tilde{C} = (C \star D)^{\perp}$. Equations (3.12) and (3.13) are now combined to yield

$$k_C = n - \dim(C^{\perp}) = n - \dim(D \star \underbrace{(C \star D)^{\perp}}_{=\tilde{C}}) \leq n - k_D - d_{C\star D} + 2.$$

Rearranging terms then gives

$$d_{C\star D} \leq n - k_C - k_D + 2.$$

■

A more general result for the star product of any number of codes is also given in [14], however we only consider products of pairs. By Proposition 3.9 we have the product of GRS codes as

$$\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u}) \star \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v}) = \mathrm{GRS}_{\min\{n,k+l-1\}}(\boldsymbol{\alpha}, \boldsymbol{u} \star \boldsymbol{v}).$$

Hence, by the MDS property of GRS codes, $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u}) \star \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v})$ has minimum distance $d = \max\{1, n-k-l+2\}$ and we see by Proposition 3.11 that the star product of GRS codes meet this multiplicative Singleton bound. This means that the choice of GRS codes as storage code C and retrieval code D in some sense maximises the rate since the rate is an increasing function of $d_{C\star D}$.

It is actually the case that the only codes with dimension $k > 1$ that meet this multiplicative Singleton-bound are GRS codes. This is Theorem 14 (i) of [13].

We will fix $t = 1$ for varying k and vice versa to compare this scheme with the PIR capacities of Section 2.1.

Proposition 3.12. *Let $C = \mathrm{GRS}_1(\boldsymbol{\alpha}, \boldsymbol{u})$ be a length n storage code and let $D = \mathrm{GRS}_{1\leq t\leq n-1}(\boldsymbol{\alpha}, \boldsymbol{v})$ be a length n retrieval code applied in the scheme $\mathscr{S}_{FGHK}$. Then the rate of $\mathscr{S}_{FGHK}$ tends toward the t-collusion PIR capacity $(1-t/n)/(1-(t/n)^M)$ as the number of files M tends to ∞.*

Proof. The code C is equivalent to the $[n, 1, n]$ repetition code, and the rate of this scheme is by Theorem 3.10 given by

$$\frac{n-t}{n} = 1 - \frac{t}{n}.$$

As $M \to \infty$ we get for the capacity

$$\lim_{M\to\infty} \frac{1-\frac{t}{n}}{1-\left(\frac{t}{n}\right)^M} = 1 - \frac{t}{n}$$

■

Proposition 3.13. *Let $C = \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u})$ be a length n storage code and let $D = \mathrm{GRS}_1(\boldsymbol{\alpha}, \boldsymbol{v})$ be a length n retrieval code applied in the scheme $\mathscr{S}_{FGHK}$. Then the rate of $\mathscr{S}_{FGHK}$ tends toward the collusion free PIR capacity $(1-k/n)/(1-(k/n)^M)$ as the number of files M tends to ∞.*

Proof. By Theorem 3.10 the rate of this scheme is

$$\frac{n-k}{n} = 1 - \frac{k}{n}.$$

As $M \to \infty$ we get the capacity

$$\lim_{M\to\infty} \frac{1-\frac{k}{n}}{1-\left(\frac{k}{n}\right)^M} = 1 - \frac{k}{n}.$$

■

This shows that this GRS scheme is good in the cases $t = 1$ and $k = 1$ at least when we have a lot of files. One of the main benefits for this scheme is its application in the intermediate cases where both $k > 1$ and $t > 1$ however a general t-collusion PIR capacity is unfortunately not known. It should also be noted that the size of the field of computation must satisfy $q^m \geq n$, which we will elaborate upon later.

Example 3.14. We will now consider an example of the PIR scheme $\mathscr{S}_{FGHK}$. The scheme parameters of this examples can be seen in Table 1.

n	M	k	t	b	s	c	g	C	D	$C \star D$
7	4	2	3	3	2	3	1	$\mathrm{GRS}_2(\boldsymbol{\alpha}, \mathbf{1})$	$\mathrm{GRS}_3(\boldsymbol{\alpha}, \mathbf{1})$	$\mathrm{GRS}_4(\boldsymbol{\alpha}, \mathbf{1})$

Table 1: Scheme parameters for our example of the scheme $\mathscr{S}_{FGHK}$ using GRS codes.

In particular, we wish to encode 4 files $x^1, \ldots, x^4$ of size 3×2 onto 7 servers $s_1, \ldots, s_7$ using the Reed-Solomon code $\mathrm{GRS}_2(\boldsymbol{\alpha}, \mathbf{1})$ with $\boldsymbol{\alpha} = (0, 1, 2, 3, 4, 5, 6)$. We will consider codes over the finite field $\mathbb{F}_7$. Usually the number of files would be far larger than the number of servers, but for both ease of computation and clarity we consider only a few files. The files will be

$$x^1 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, x^2 = \begin{bmatrix} 2 & 4 \\ 0 & 2 \\ 4 & 0 \end{bmatrix}, x^3 = \begin{bmatrix} 1 & 3 \\ 5 & 0 \\ 2 & 4 \end{bmatrix}, x^4 = \begin{bmatrix} 2 & 5 \\ 3 & 6 \\ 0 & 1 \end{bmatrix}. \tag{3.14}$$

The rows of the canonical generator matrix G of a GRS code $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ are given by the image of the canonical basis of $\mathbb{F}_{q^m}[X]^k$ under the evaluation homomorphism (3.6):

$$G = \begin{bmatrix} v_1 & \ldots & v_n \\ v_1\alpha_1 & \ldots & v_n\alpha_n \\ \vdots & \ddots & \vdots \\ v_1\alpha_1^{k-1} & \ldots & v_n\alpha_n^{k-1} \end{bmatrix}. \tag{3.15}$$

Hence, the canonical generator matrices for C and D are given by respectively

$$G_C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}, G_D = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 1 & 4 & 2 & 2 & 4 & 1 \end{bmatrix}. \tag{3.16}$$

A systematic generator matrix $\tilde{G}$ for a GRS code $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ is given by

$$\tilde{G} = \begin{bmatrix} v_1 f_1(\alpha_1) & \ldots & v_n f_1(\alpha_n) \\ \vdots & \ddots & \vdots \\ v_1 f_k(\alpha_1) & \ldots & v_n f_k(\alpha_n) \end{bmatrix}, \tag{3.17}$$

for

$$f_i(X) = \frac{1}{v_i} \prod_{j \in [k] \setminus \{i\}} \frac{X - \alpha_j}{\alpha_i - \alpha_j}.$$

Exactly as in the Lagrange interpolation formula we get $f_i(\alpha_i) = v_i^{-1}$ and $f_i(\alpha_j) = 0$ for $j \in [k]\backslash\{i\}$, and this is a systematic generator matrix for the GRS code. That it is a generator matrix can be seen by the fact that $\{\prod_{j\in[k]\backslash\{i\}}(X-\alpha_j)/(\alpha_i-\alpha_j)\}_{i=1}^{k}$ forms a basis for $\mathbb{F}_{q^m}[X]^k$ and then applying the evaluation homomorphism. The systematic generator matrices for C and D are respectively given by

$$\tilde{G}_C = \begin{bmatrix} 1 & 0 & 6 & 5 & 4 & 3 & 2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}, \tilde{G}_D = \begin{bmatrix} 1 & 0 & 0 & 1 & 3 & 6 & 3 \\ 0 & 1 & 0 & 4 & 6 & 6 & 4 \\ 0 & 0 & 1 & 3 & 6 & 3 & 1 \end{bmatrix}$$

For our PIR scheme we will choose the systematic generator matrix $\tilde{G}_C$ for encoding and the canonical generator matrix G_D for the retrieval code generator matrix.

By G_C we encode the files $x^1, \ldots, x^4$ as

$$Y = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 6 & 5 & 4 & 3 & 2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 2 & 4 \\ 0 & 2 \\ 4 & 0 \\ 1 & 3 \\ 5 & 0 \\ 2 & 4 \\ 2 & 5 \\ 3 & 6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 6 & 5 & 4 & 3 & 2 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 0 \\ 3 & 4 & 5 & 6 & 0 & 1 & 2 \\ 5 & 6 & 0 & 1 & 2 & 3 & 4 \end{bmatrix} \} \; y^1 = x^1 G_C$$
$$\begin{bmatrix} 2 & 4 & 6 & 1 & 3 & 5 & 0 \\ 0 & 2 & 4 & 6 & 1 & 3 & 5 \\ 4 & 0 & 3 & 6 & 2 & 5 & 1 \end{bmatrix} \} \; y^2 = x^2 G_C$$
$$\begin{bmatrix} 1 & 3 & 5 & 0 & 2 & 4 & 6 \\ 5 & 0 & 2 & 4 & 6 & 1 & 3 \\ 2 & 4 & 6 & 1 & 3 & 5 & 0 \end{bmatrix} \} \; y^3 = x^3 G_C$$
$$\begin{bmatrix} 2 & 5 & 1 & 4 & 0 & 3 & 6 \\ 3 & 6 & 2 & 5 & 1 & 4 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix} \} \; y^4 = x^4 G_C$$
$$\boldsymbol{y}_1 \; \boldsymbol{y}_2 \; \boldsymbol{y}_3 \; \boldsymbol{y}_4 \; \boldsymbol{y}_5 \; \boldsymbol{y}_6 \; \boldsymbol{y}_7$$

From Y we can clearly see what parts of the files x^i are encoded on the servers s_j.

We will now consider the retrieval of a file, say x^3 in a private manner. The $mb = 12$ codewords $\boldsymbol{d}^{l,a}$ are chosen uniformly at random from D by choosing vectors $\boldsymbol{v}^{l,a} \in \mathbb{F}_7^3$ and defining $\boldsymbol{d}^{l,a} = \boldsymbol{v}^{l,a} G_D$. Hence, we get by collecting these vectors as rows of a matrix

V:

$$V^T = \begin{array}{c} \begin{matrix} \boldsymbol{d}^{1,1} & \boldsymbol{d}^{1,2} & \boldsymbol{d}^{1,3} & \boldsymbol{d}^{2,1} & \boldsymbol{d}^{2,2} & \boldsymbol{d}^{2,3} & \boldsymbol{d}^{3,1} & \boldsymbol{d}^{3,2} & \boldsymbol{d}^{3,3} & \boldsymbol{d}^{4,1} & \boldsymbol{d}^{4,2} & \boldsymbol{d}^{4,3} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 5 & 3 & 3 & 1 & 3 & 4 & 0 & 1 & 4 & 1 \\ 6 & 0 & 4 & 2 & 2 & 5 & 4 & 0 & 0 & 0 & 3 & 4 \\ 5 & 5 & 2 & 2 & 0 & 2 & 6 & 0 & 4 & 1 & 4 & 6 \\ 4 & 1 & 6 & 3 & 4 & 6 & 2 & 4 & 5 & 4 & 0 & 0 \\ 3 & 2 & 2 & 5 & 0 & 3 & 6 & 5 & 3 & 2 & 5 & 0 \\ 2 & 1 & 4 & 1 & 2 & 0 & 4 & 3 & 5 & 2 & 5 & 6 \\ 1 & 5 & 5 & 5 & 3 & 4 & 3 & 5 & 4 & 4 & 0 & 4 \end{bmatrix} \end{array}. \tag{3.18}$$

We will construct queries $\boldsymbol{q}_j^3$ from the rows $\boldsymbol{d}_j$ of $V^T, j \in [7]$. To construct these queries the index set $J = \{1, 2, 3\}$ and the subsets for the first and second iteration are respectively

$$J_1^1 = \{1\}, J_1^2 = \{2\}, J_1^3 = \{3\} \subseteq J,$$
$$J_2^1 = \{2\}, J_2^2 = \{3\}, J_2^3 = \{1\} \subseteq J.$$

Using these sets the queries are constructed as in (3.2) and in the first iteration of the scheme we get $\boldsymbol{q}_j^3 = \boldsymbol{d}_j$ for $j \neq 1, 2, 3$ and

$$\boldsymbol{q}_1^3 = \boldsymbol{d}_1 + \boldsymbol{e}_7, \boldsymbol{q}_2^3 = \boldsymbol{d}_2 + \boldsymbol{e}_8, \boldsymbol{q}_3^3 = \boldsymbol{d}_3 + \boldsymbol{e}_9.$$

Similarly, for a new set of $\boldsymbol{d}_j$ we get during the second iteration $\boldsymbol{q}_j^3 = \boldsymbol{d}_j$ for $j \neq 1, 2, 3$ and

$$\boldsymbol{q}_1^3 = \boldsymbol{d}_1 + \boldsymbol{e}_8, \boldsymbol{q}_2^3 = \boldsymbol{d}_2 + \boldsymbol{e}_9, \boldsymbol{q}_3^3 = \boldsymbol{d}_3 + \boldsymbol{e}_7.$$

During response computation this will ensure that $k = 2$ symbols are downloaded from every row of the encoded file y^3. Consider Figure 1 where it is shown what symbols are downloaded in each iteration. The red symbols are downloaded in iteration 1 using J_1 and the blue symbols are downloaded in iteration 2 using J_2.

$$\left.\begin{bmatrix} \mathbf{1} & \mathbf{3} & 5 & 0 & 2 & 4 & 6 \\ 5 & \mathbf{0} & \mathbf{2} & 4 & 6 & 1 & 3 \\ \mathbf{2} & 4 & \mathbf{6} & 1 & 3 & 5 & 0 \end{bmatrix}\right\} y^3 = x^3 G_C$$
$$\boldsymbol{y}_1^3\ \boldsymbol{y}_2^3\ \boldsymbol{y}_3^3\ \boldsymbol{y}_4^3\ \boldsymbol{y}_5^3\ \boldsymbol{y}_6^3\ \boldsymbol{y}_7^3$$

Figure 1: Downloading of encoded symbols.

The queries q_j^3 is sent to the corresponding servers s_j and the responses are calculated as in (3.3). They are adjoined into the response vector $\boldsymbol{r}^i$:

$$\boldsymbol{r}^{\boldsymbol{i}} = (3, 2, 5, 5, 4, 1, 1).$$

We find a parity check matrix H for $C \star D$ as a generator matrix for its dual $(C \star D)^{\perp}$. This is not a problem to find since we know exactly what code this is by Propositions 3.8 and 3.9. Namely, since $C \star D = \mathrm{GRS}_4(\boldsymbol{\alpha}, \mathbf{1})$ we have

$$(C \star D)^{\perp} = \mathrm{GRS}_3(\boldsymbol{\alpha}, \boldsymbol{z})$$

for $z_i = 1/\prod_{j \neq i}(\alpha_i - \alpha_j)$. We can easily evaluate this since for any finite field of size q we have $f(X) = X^q - X = \prod_{\alpha \in \mathbb{F}_q}(X - \alpha)$ and by taking the formal derivative:

$$f'(X) = -1 = \sum_{\beta \in \mathbb{F}_q} \prod_{\alpha \neq \beta} (X - \alpha). \tag{3.19}$$

Evaluating the RHS in any element α_0 of $\mathbb{F}_q$ will only leave the term $\prod_{\alpha \neq \alpha_0}(\alpha_0 - \alpha)$, whence $z_i = -1$ for all $i \in [7]$. Hence, this code is equivalent to $\mathrm{GRS}_3(\boldsymbol{\alpha}, \mathbf{1})$ and we choose G_D as in (3.16) as the generator matrix for $(C \star D)^{\perp}$. By using G_D as a parity check matrix H for $C \star D$ we receive

$$\boldsymbol{r}^i H^T = (0, 5, 3) = (y_1^3(1), y_2^3(2), y_2^3(3), \mathbf{0}_4) H^T,$$

by Lemma 3.3. Hence, we can recover $y_1^3(1), y_2^3(2), y_2^3(3)$ by the Gauss elimination algorithm as

$$\left[\begin{array}{ccccccc|c} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 5 \\ 0 & 1 & 4 & 2 & 2 & 4 & 1 & 3 \end{array}\right] \sim \left[\begin{array}{ccccccc|c} 1 & 0 & 0 & 1 & 3 & 6 & 3 & 1 \\ 0 & 1 & 0 & 4 & 6 & 6 & 4 & 0 \\ 0 & 0 & 1 & 3 & 6 & 3 & 1 & 6 \end{array}\right].$$

Thus, $y_1^3(1) = 1, y_2^3(2) = 0$, and $y_3^3(3) = 6$ which agrees with the first iteration downloaded symbols of Figure 1. The symbols in the second iteration are downloaded in a completely analogous manner.

Suppose we have downloaded the symbols $(1, 3)$, $(0, 2)$ and $(2, 6)$ of respectively row $1, 2$ and 3 of y^3. Since $k = 2$ we can recover the rows of x^3 by choosing the appropriate submatrices of $\tilde{G}_C$ and solving for x^3 as

$$\begin{aligned} \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 2 \end{array}\right] &\Rightarrow \text{Row 1 of } x^3 \text{ equals } (1, 2), \\ \left[\begin{array}{cc|c} 0 & 1 & 0 \\ 6 & 2 & 2 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 0 & 5 \\ 0 & 1 & 0 \end{array}\right] &\Rightarrow \text{Row 2 of } x^3 \text{ equals } (5, 0), \\ \left[\begin{array}{cc|c} 1 & 0 & 2 \\ 6 & 2 & 6 \end{array}\right] \sim \left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 4 \end{array}\right] &\Rightarrow \text{Row 3 of } x^3 \text{ equals } (2, 4). \end{aligned}$$

By (3.14) we can see that x^3 has been recovered correctly and we know that it has been 3-privately recovered.

If we have a lot of files in our DSS the PIR scheme $\mathscr{S}_{FGHK}$ will have a large computational cost at the servers. This is particularly the case when many servers are also present since we demand the size of the finite field $q^m \geq n$, and the computational complexity is an increasing function of both q^m and n. We will consider a new scheme that tries to mend on this issue.

PIR Using Subfield Subcodes

The preliminaries of this section are inspired by [15].

The first thing we will present is a definition of a *subfield subcode*.

Definition 4.1. Let C be a length n code over a finite field $\mathbb{F}_{q^m}$. Then the subfield subcode $C|_{\mathbb{F}_q}$ of C is defined as

$$C|_{\mathbb{F}_q} = C \cap \mathbb{F}_q^n.$$

It is clear from this definition that passing to the subfield subcode does not reduce the minimum distance of a code which will be important for our analysis. It does however reduce the dimension of the code in general.

There are other ways to construct codes on finite fields $\mathbb{F}_q$ using codes over extension fields $\mathbb{F}_{q^m}$. One way to do this is by the *norm* and *trace* functions.

Definition 4.2. Let L/K be a finite field extension of degree m, and let $\alpha \in L$ define a K-linear mapping $\mu_\alpha : L \to L$ given by $\mu_\alpha(z) = \alpha z$. Then the trace of α, $\mathrm{Tr}_{L/K}(\alpha)$, is defined as

$$\mathrm{Tr}_{L/K}(\alpha) := \mathrm{Tr}(\mu_\alpha).$$

Similarly, the norm of α, $\mathrm{N}_{L/K}(\alpha)$, is defined as

$$\mathrm{N}_{L/K}(\alpha) = \det(\mu_\alpha).$$

We will write $\mathrm{Tr}(\alpha)$ and $\mathrm{N}(\alpha)$ when the field extension is clear from contexts. By choosing a basis $\beta_1, \ldots, \beta_m \in L$ for L/K (the extension is finite), write

$$\mu_\alpha(\beta_i) = \sum_{j=1}^{m} a_{ij}\beta_j,$$

for $a_{ij} \in K$. Hence, with respect to this basis, the trace and norm are given by

$$\mathrm{Tr}(\alpha) = \sum_{i=1}^{m} a_{ii}, \qquad \mathrm{N}(\alpha) = \det(a_{ij})_{1 \le i,j \le m}.$$

We are of course mostly interested in extension fields of the type $\mathbb{F}_{q^m}/\mathbb{F}_q$. In this case, the trace and norm are given by

$$\mathrm{Tr}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) = \alpha + \alpha^q + \cdots + \alpha^{q^{m-1}}, \qquad \mathrm{N}_{\mathbb{F}_{q^m}/\mathbb{F}_q}(\alpha) = \alpha^{1+q+\cdots+q^{m-1}}.$$

This can be seen by considering the fact that the roots of the characteristic polynomial $\chi_{\alpha,\mathbb{F}_{q^m}/\mathbb{F}_q}$ of μ_α are exactly the elements of the Galois group $\mathrm{Aut}(\mathbb{F}_{q^m}/\mathbb{F}_q)$ applied on α. The Galois group of $\mathbb{F}_{q^m}/\mathbb{F}_q$ is the group generated by the Frobenius automorphism $\alpha \mapsto \alpha^q$. For details see e.g. Theorem 2.22 of [10].

Definition 4.3. For an element $\boldsymbol{c} \in (\mathbb{F}_{q^m})^n$ we define the *coordinate-wise trace mapping* $\mathrm{Tr} : (\mathbb{F}_{q^m})^n \to \mathbb{F}_q^n$ by

$$\mathrm{Tr}(\boldsymbol{c}) = (\mathrm{Tr}(c_1), \ldots, \mathrm{Tr}(c_n)) .$$

Now for a code $C \subseteq (\mathbb{F}_{q^m})^n$ the *trace code* of C, $\mathrm{Tr}(C)$, is defined as

$$\mathrm{Tr}(C) = \{\mathrm{Tr}(\boldsymbol{c}) \mid \boldsymbol{c} \in C\} \subseteq \mathbb{F}_q^n.$$

Subfield subcodes and trace codes are quite closely related. Under the assumption that $q \nmid m$ then for a code $C \subseteq (\mathbb{F}_{q^m})^n$ we have $C|_{\mathbb{F}_q} \subseteq \mathrm{Tr}(C)$. Indeed, for any $\boldsymbol{c} \in C|_{\mathbb{F}_q}$ we have $\mathrm{Tr}(\boldsymbol{c}) = (mc_1, \ldots, mc_n) = m\boldsymbol{c} \in C|_{\mathbb{F}_q}$. An important theorem due to Delsarte [5] relates subfield subcodes and trace codes.

Theorem 4.4 (Delsarte's Theorem)**.** *For a code $C \subseteq (\mathbb{F}_{q^m})^n$ we have*

$$(C|_{\mathbb{F}_q})^\perp = \mathrm{Tr}(C^\perp).$$

Proof. The first inclusion $\mathrm{Tr}(C^\perp) \subseteq (C|_{\mathbb{F}_q})^\perp$ follows readily since for $\boldsymbol{c} = (c_1, \ldots, c_n) \in C|_{\mathbb{F}_q}$ and $\boldsymbol{v} = (v_1, \ldots, v_n) \in C^\perp$ we have

$$\langle \boldsymbol{c}, \mathrm{Tr}(\boldsymbol{v}) \rangle = \sum_{i=1}^{n} c_i \mathrm{Tr}(v_i) = \mathrm{Tr}(\langle \boldsymbol{c}, \boldsymbol{v} \rangle) = 0.$$

To prove the inclusion $(C|_{\mathbb{F}_q})^\perp \subseteq \mathrm{Tr}(C^\perp)$ we assume that the equivalent statement $\mathrm{Tr}(C^\perp)^\perp \subseteq C|_{\mathbb{F}_q}$ is false and seek a contradiction. Hence, take elements $\boldsymbol{c} \in \mathrm{Tr}(C^\perp)^\perp \setminus C|_{\mathbb{F}_q}$ and $\boldsymbol{v} \in (C|_{\mathbb{F}_q})^\perp$ such that $\langle \boldsymbol{c}, \boldsymbol{v} \rangle \neq 0$. An element $\gamma \in \mathbb{F}_{q^m}$ is chosen such that $\mathrm{Tr}(\gamma \langle \boldsymbol{c}, \boldsymbol{v} \rangle) \neq 0$. Then

$$0 \neq \mathrm{Tr}(\gamma \langle \boldsymbol{c}, \boldsymbol{v} \rangle) = \langle \boldsymbol{c}, \mathrm{Tr}(\gamma \boldsymbol{v}) \rangle. \tag{4.1}$$

Since $\gamma \boldsymbol{v}$ is obviously in $C^\perp$ and $\boldsymbol{c} \in \mathrm{Tr}(C^\perp)^\perp$ we have $\mathrm{Tr}(\gamma \boldsymbol{v}) \in \mathrm{Tr}(C^\perp)$ and

$$\langle \boldsymbol{c}, \mathrm{Tr}(\gamma \boldsymbol{v}) \rangle = 0,$$

which by (4.1) is a contradiction. Thus, $\mathrm{Tr}(C^\perp)^\perp \subseteq C|_{\mathbb{F}_q}$ or equivalently $(C|_{\mathbb{F}_q})^\perp \subseteq \mathrm{Tr}(C^\perp)$ is true. ∎

We can combine Delsarte's theorem with Proposition 3.8 to get the following relation between subfield subcodes and trace codes of GRS codes.

Corollary 4.5. *For a length n GRS code over $\mathbb{F}_{q^m}$ with support $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m})^n$ and scaling vector $\boldsymbol{v} \in (\mathbb{F}_{q^m}^*)^n$ we have*

$$\left(\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})|_{\mathbb{F}_q} \right)^\perp = \mathrm{Tr}\left(\mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \tilde{\boldsymbol{v}}) \right),$$

where

$$\tilde{v}_i = \frac{1}{v_i \prod_{j \neq i} (\alpha_i - \alpha_j)}. \tag{4.2}$$

Proof. By Theorem 4.4 we have that

$$\left(\mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v})|_{\mathbb{F}_q}\right)^{\perp} = \mathrm{Tr}\left((\mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v}))^{\perp}\right),$$

and by Proposition 3.8 we have that

$$(\mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v}))^{\perp} = \mathrm{GRS}_{n-k}(\boldsymbol{\alpha},\tilde{\boldsymbol{v}}),$$

with $\tilde{\boldsymbol{v}}$ as in (4.2) completing the proof. ∎

This relation can be visualised in Figure 2

$$\begin{array}{ccc} \mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v}) & \xleftarrow{\quad\text{Dual}\quad} & \mathrm{GRS}_{n-k}(\boldsymbol{\alpha},\tilde{\boldsymbol{v}}) \\ \Big\downarrow \cap\mathbb{F}_q^n & & \mathrm{Tr}\Big\downarrow \\ \mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v})|_{\mathbb{F}_q} & \xleftarrow{\quad\text{Dual}\quad} & \mathrm{Tr}(\mathrm{GRS}_{n-k}(\boldsymbol{\alpha},\tilde{\boldsymbol{v}})) \end{array}$$

Figure 2: A subfield subcode and trace code relation for GRS codes.

We wish to apply this relation to the scheme $\mathscr{S}_{FGHK}$. In this new setting we will still consider the same DSS with the storage code as a GRS code $C = \mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v})$ over $\mathbb{F}_{q^m}$ storing M files on n servers. Throughout the rest of the section, the DSS $\mathscr{D}$ in question will thus be such a DSS. This scheme will however take the retrieval code D as a subfield subcode of a GRS code $\mathrm{GRS}_t(\boldsymbol{\alpha},\boldsymbol{u})$, and we will denote this scheme by $\mathscr{S}|_{\mathbb{F}_q}$. We have some of the tools to state what kind of rate and protection such a PIR scheme can expect to achieve.

Theorem 4.6. *The PIR scheme $\mathscr{S}|_{\mathbb{F}_q}$ using DSS $\mathscr{D}$ and retrieval code $D|_{\mathbb{F}_q} = \mathrm{GRS}_t(\boldsymbol{\alpha},\boldsymbol{u})|_{\mathbb{F}_q}$ has rate*

$$R \geq \frac{n-k-t+1}{n}$$

and is $d_{\mathrm{Tr}(D^{\perp})} - 1$-private.

Proof. The rate follows by the inclusion

$$C \star D \subseteq \mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v}) \star \mathrm{GRS}_t(\boldsymbol{\alpha},\boldsymbol{u}),$$

and the fact that $d_{\mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v})\star\mathrm{GRS}_t(\boldsymbol{\alpha},\boldsymbol{u})} = n-k-t+2$. Then by Theorem 3.4 the rate is

$$R = (d_{C\star D}-1)/n \geq (d_{\mathrm{GRS}_k(\boldsymbol{\alpha},\boldsymbol{v})\star\mathrm{GRS}_t(\boldsymbol{\alpha},\boldsymbol{u})}-1)/n = (n-k-t+1)/n.$$

The privacy follows immediately by Theorem 3.5 and Delsarte's theorem. ∎

What we hope to achieve by this scheme is an improved server-side computational cost by reducing the size of the field of operations. We do however pay with a potential reduction in privacy as we have for a code C over $\mathbb{F}_{q^m}$:

$$\dim C \leq \dim \mathrm{Tr}(C) \leq m \dim C, \tag{4.3}$$

by the linearity of the trace mapping. Since $D^\perp$ is MDS and the trace mapping potentially increases the dimension at least a similar reduction in privacy can be expected by the Singleton bound.

This issue can also be addressed from the viewpoint yielded by the parity check matrix of a subfield subcode of a GRS code. For a length n GRS code $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ by Proposition 3.8 and (3.16) we have a parity check matrix H for $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ given by

$$H = \begin{bmatrix} \tilde{v}_1 & \dots & \tilde{v}_n \\ \alpha_1 \tilde{v}_1 & \dots & \alpha_n \tilde{v}_n \\ \vdots & \ddots & \vdots \\ \alpha_1^{n-k-1} \tilde{v}_1 & \dots & \alpha_n^{n-k-1} \tilde{v}_n \end{bmatrix},$$

where $\tilde{\boldsymbol{v}}$ is given as in (3.7). Thus, the subfield subcode $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})|_{\mathbb{F}_q}$ consists of all vectors $\boldsymbol{v} \in (\mathbb{F}_q)^n$ satisfying $\boldsymbol{v} H^T = \boldsymbol{0}$. We can consider H as a parity check matrix for $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})|_{\mathbb{F}_q}$ by representing the entries of H as elements of $(\mathbb{F}_q)^m$ instead of elements of $\mathbb{F}_{q^m}$ using some appropriate basis. This yields us a size $n \times m(n-k)$ parity check matrix for $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})|_{\mathbb{F}_q}$. Hence, if we have some retrieval code $D|_{\mathbb{F}_q} = \mathrm{GRS}_t(\boldsymbol{\alpha}, \boldsymbol{v})|_{\mathbb{F}_q}$ then to ensure that $(D|_{\mathbb{F}_q})^\perp = \mathrm{Tr}(D^\perp)$ has dimension strictly smaller than n we must enforce the restriction $n/m > n - t$. If a lot of rows of H are linearly dependent $n - t$ can be quite a lot larger than n/m. Hence, if we can choose GRS codes where this is satisfied we can get codes that does not decrease collusion protection significantly.

On the other hand, let us argue why this restriction could seem reasonable. Suppose that the rows of H are chosen uniformly and independently at random from $(\mathbb{F}_q)^n$. Then the probability that the rows of H span $(\mathbb{F}_q)^n$ is given as

$$\prod_{j=0}^{n-1} \left(1 - \frac{q^j}{q^{m(n-t)}}\right)$$

assuming that $m(n-t) \geq n$. As an example take $\mathbb{F}_{q^m} = \mathbb{F}_{16}$ and $n = 16$. Taking $D = \mathrm{GRS}_{10}(\boldsymbol{v}, \boldsymbol{\alpha})$ yields a probability that $D|_{\mathbb{F}_q}$ spans $(\mathbb{F}_q)^n$ of 0.996 under the assumption that the parity check matrix has uniformly distributed rows. This assumption is obviously flawed, but it helps to enlighten upon the "strictness" of the restriction $n - t < n/m$. To ensure that the scheme $\mathscr{S}|_{\mathbb{F}_q}$ has nonzero rate we have $n - t \geq k$, combining to

$$k \leq n - t < \frac{n}{m},$$

which especially for small n and small prime fields $\mathbb{F}_q$ can be an issue. It should be noted that subfield subcodes of GRS codes are also known as *alternant codes*.

Example 4.7. A certain class of subfield subcodes with good parameters exists: the *(classical) Goppa codes*. These are codes defined as follows. Choose a support $L = (\alpha_1, \ldots, \alpha_n) \in (\mathbb{F}_{q^m})^n$ where $\alpha_i \neq \alpha_j$ for $i \neq j$, and a polynomial $g \in \mathbb{F}_{q^m}[X]$ of degree τ which has no roots among the elements of L ($\tau < n/m$). Then the Goppa code $\Gamma_q(L, g)$ is defined as the code

$$\Gamma_q(L, g) = \left\{ \boldsymbol{c} \in \mathbb{F}_q^n \;\middle|\; \sum_{i=1}^{n} \frac{c_i}{X - \alpha_i} \equiv 0 \pmod{g} \right\}. \tag{4.4}$$

At first glance, this code seems to have not much in common with GRS codes. To see that it does (as is done in [2]) we first define the code

$$\Gamma_{q^m}(L, g) = \left\{ \boldsymbol{c} \in (\mathbb{F}_{q^m})^n \;\middle|\; \sum_{i=1}^{n} \frac{c_i}{X - \alpha_i} \equiv 0 \pmod{g} \right\}.$$

Let f be any polynomial $f \in \mathbb{F}_{q^m}[X]^n$ where $f \in \langle g \rangle$. Such an f can be recovered by Lagrange interpolation on f evaluated over the elements of L. Hence,

$$f(X) = \sum_{i=1}^{n} f(\alpha_i) \prod_{j \neq i} \frac{X - \alpha_j}{\alpha_i - \alpha_j}.$$

We define the polynomial $h(X) = \prod_{i=1}^{n}(X - \alpha_i)$. By Leibniz's rule this has the formal derivative

$$h'(X) = \sum_{i=1}^{n} \prod_{j \neq i} (X - \alpha_j).$$

Hence,

$$\begin{aligned} \frac{f(X)}{h(X)} &= \frac{\sum_{i=1}^{n} f(\alpha_i) \prod_{j \neq i} \frac{X - \alpha_j}{\alpha_i - \alpha_j}}{h(X)} \\ &= \sum_{i=1}^{n} \frac{f(\alpha_i)}{(X - \alpha_i)} \prod_{j \neq i} \frac{1}{(\alpha_i - \alpha_j)} \\ &= \sum_{i=1}^{n} \frac{f(\alpha_i)}{(X - \alpha_i) h'(\alpha_i)}. \end{aligned} \tag{4.5}$$

Since $f \in \langle g \rangle$ the polynomial $\sum_{i=1}^{n} f(\alpha_i)/((X - \alpha_i)h'(\alpha_i))$ is zero in the ring $\mathbb{F}_{q^m}[X]/\langle g \rangle$, the codeword

$$\left(\frac{f(\alpha_1)}{\prod_{j \neq 1}(\alpha_1 - \alpha_j)}, \ldots, \frac{f(\alpha_n)}{\prod_{j \neq n}(\alpha_n - \alpha_j)} \right) \tag{4.6}$$

must be in $\Gamma_{q^m}(L, g)$. Write $f = \varphi g$. Then φ is a polynomial of $\mathbb{F}_{q^m}[X]^{n-\tau}$, and (4.6) is a codeword of $\mathrm{GRS}_{n-\tau}(L, \boldsymbol{v})$ for $v_i = g(\alpha_i)/h'(\alpha_i)$, and we get $\mathrm{GRS}_{n-\tau}(L, \boldsymbol{v}) \subseteq \Gamma_{q^m}(L, g)$.

For the other implication we choose a codeword $\boldsymbol{c} \in \Gamma_{q^m}(L, g)$ and define

$$f(X) = \sum_{i=1}^{n} \frac{c_i \prod_{j=1}^{n}(X - \alpha_j)}{X - \alpha_i}.$$

Since

$$\sum_{i=1}^{n} \frac{c_i}{X - \alpha_i} = 0$$

in $\mathbb{F}_{q^m}[X]/\langle g \rangle$ we must have $f \in \langle g \rangle$. The degree of f is strictly smaller than n and by Lagrange interpolation we get $c_i = f(\alpha_i)/(\prod_{j \neq i}(\alpha_i - \alpha_j))$ as in (4.5). Thus

$$\boldsymbol{c} = \left(\frac{f(\alpha_1)}{\prod_{j \neq 1}(\alpha_1 - \alpha_j)}, \ldots, \frac{f(\alpha_n)}{\prod_{j \neq n}(\alpha_n - \alpha_j)} \right),$$

and we can conclude that

$$\begin{aligned}
\Gamma_{q^m}(L, g) &= \left\{ \left(\frac{f(\alpha_1)}{\prod_{j \neq 1}(\alpha_1 - \alpha_j)}, \ldots, \frac{f(\alpha_n)}{\prod_{j \neq n}(\alpha_n - \alpha_j)} \right) \;\middle|\; f \in \mathbb{F}_{q^m}[X]^n, f \in \langle g \rangle \right\} \\
&= \left\{ \left(\frac{g(\alpha_1)}{\prod_{j \neq 1}(\alpha_1 - \alpha_j)} \tilde{f}(\alpha_1), \ldots, \frac{g(\alpha_n)}{\prod_{j \neq n}(\alpha_n - \alpha_j)} \tilde{f}(\alpha_n) \right) \;\middle|\; \tilde{f} \in \mathbb{F}_{q^m}[X]^{n-\tau} \right\} \\
&= \mathrm{GRS}_{n-\tau}(L, \boldsymbol{v})
\end{aligned}$$

for $v_i = g(\alpha_i)/\prod_{j \neq i}(\alpha_i - \alpha_j)$. Clearly, we get $\Gamma_q(L, g) = \Gamma_{q^m}(L, g)|_{\mathbb{F}_q}$ and we see how Goppa codes are subfield subcodes of GRS codes. It is known that Goppa codes meet the Gilbert-Varshamov bound asymptotically and have in general good parameters (see e.g. [11]). Hence such codes would seem as a good choice for the retrieval code $D|_{\mathbb{F}_q}$ of $\mathscr{S}|_{\mathbb{F}_q}$.

Let us consider some more concrete examples of such a subfield subcode scheme. We will consider the scheme $\mathscr{S}_{FGHK}$ for a certain choice of retrieval code D and compare it to its corresponding subfield subcode scheme $\mathscr{S}|_{\mathbb{F}_q}$.

Example 4.8. As a first example we choose the storage code C_1 as the $[n, 1, n]$ repetition code $C_1 = \mathrm{GRS}_1(\boldsymbol{\alpha}_1, \mathbf{1})$ for $\boldsymbol{\alpha}_1$ as all of $\mathbb{F}_n$ (n is a prime power) in any order. The retrieval code is chosen as the Reed-Solomon code $D_1 = \mathrm{GRS}_{n-1}(\boldsymbol{\alpha}_1, \mathbf{1})$ which has the $[n, 1, n]$ repetition code as its dual. This yields a scheme $\mathscr{S}_{FGHK}$ with rate and collusion protection respectively as

$$R_1 = \frac{d_{C_1 \star D_1} - 1}{n} = \frac{1}{n}, \qquad\qquad t_1 = d_{D_1^\perp} - 1 = n - 1.$$

By passing to the subfield subcode scheme $\mathscr{S}|_{\mathbb{F}_2}$ we get retrieval code $D_1|_{\mathbb{F}_2}$ which has the length n matrix $H = [1 \ \cdots \ 1]$ as its parity check matrix just as D_1. Therefore, the

code $D_1|_{\mathbb{F}_2}$ has dimension $n-1$ and minimum distance 2. Furthermore, since C_1 is the repetition code we have for the star product code $C_1 \star D_1|_{\mathbb{F}_2} = D_1|_{\mathbb{F}_2}$. Hence, we get rate and collution protection for the scheme $\mathscr{S}|_{\mathbb{F}_2}$ respectively as

$$R_1' = \frac{d_{C_1 \star D_1|_{\mathbb{F}_2}} - 1}{n} = \frac{1}{n}, \qquad t_1' = d_{(D_1|_{\mathbb{F}_2})^\perp} - 1 = n - 1.$$

Thus, we can pass to the subfield subcode scheme for free, but this scheme does not allow for any proper storage coding.

The next example we will consider is slightly more interesting since it allows for proper storage coding.

Example 4.9. Choose for $k \leq 3$ the storage code as the length 8 code $C_2 = \mathrm{GRS}_k(\boldsymbol{\alpha}_2, \boldsymbol{v}_2)$ where $\boldsymbol{v}_2$ is any element of $(\mathbb{F}_8^*)^8$ and $\boldsymbol{\alpha}_2$ is all of $\mathbb{F}_8$. Furthermore, we choose the retrieval code as $D_2 = \mathrm{GRS}_5(\boldsymbol{\alpha}_2, \mathbf{1})$. This yields a GRS scheme $\mathscr{S}_{FGHK}$ with rate and collusion protection as

$$R_2 = \frac{4-k}{8}, \qquad t_2 = 5.$$

The corresponding subfield subcode scheme has retrieval code $D_2|_{\mathbb{F}_2}$ with generator matrix (checked using SageMath)

$$G_2|_{\mathbb{F}_2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{4.7}$$

This is the parity check matrix of the $[8, 4, 4]$ extended binary Hamming code which is self-dual. Hence, $D_2|_{\mathbb{F}_2}$ must be the $[8, 4, 4]$ extended binary Hamming code. The star product code $C_2 \star D_2|_{\mathbb{F}_2}$ has minimum distance bounded below by the minimum distance of $C_2 \star D_2$, and computations show that these minimum distances are in fact equal. Hence, we get rate and collusion protection for the subfield subcode scheme as

$$R_2' = \frac{4-k}{8}, \qquad t_2' = 3.$$

Example 4.10. As a last example we will consider a subfield subcode scheme with a trinary retrieval code. To this end, we choose the storage code $C_3 = \mathrm{GRS}_k(\boldsymbol{\alpha}_3, \boldsymbol{v}_3)$ for some $\boldsymbol{v}_3 \in (\mathbb{F}_9^*)^9$, $\boldsymbol{\alpha}_3 \in (\mathbb{F}_9)^9$, and $k \leq 5$. We choose the retrieval code to be $D_3 = \mathrm{GRS}_4(\boldsymbol{\alpha}_3, \mathbf{1})$. This yields a scheme $\mathscr{S}_{FGHK}$ with rate and protection

$$R_3 = \frac{6-k}{9}, \qquad t_3 = 4.$$

The subfield subcode $D_3|_{\mathbb{F}_3}$ is the $[9, 3, 6]$-code with the generator matrix

$$G_3|_{\mathbb{F}_3} = \begin{bmatrix} 1 & 0 & 0 & 2 & 1 & 2 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 & 1 & 2 & 2 & 1 \\ 0 & 0 & 1 & 2 & 1 & 1 & 0 & 2 & 2 \end{bmatrix},$$

which has a $[9, 6, 3]$-code as its dual. As in the previous example the codes $C_3 \star D_3$ and $C_3 \star D_3|_{\mathbb{F}_3}$ has equal minimum distance again checked by computation. Hence, we get rate and collusion protection for the subfield subcode scheme as

$$R_3' = \frac{6-k}{9}, \qquad t_3' = 2.$$

Note, that neither of the subfield subcode schemes of Examples 4.9 nor 4.10 satisfy the assumption $n - t < n/m$.

If instead we must ensure some t-privacy for our scheme we can turn to a different approach. We choose $C = \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ as before, but take the retrieval code as $(D|_{\mathbb{F}_q})^\perp = \mathrm{Tr}(D^\perp)$ for $D = \mathrm{GRS}_{n-t}(\boldsymbol{\alpha}, \boldsymbol{v})$. This ensues that the scheme has privacy $d_{D|_{\mathbb{F}_q}} - 1 \geq t$, but we do not have any immediate lower bounds on $d_{C\star(D|_{\mathbb{F}_q})^\perp}$. This is the case since the trace code $(D|_{\mathbb{F}_q})^\perp = \mathrm{Tr}(\mathrm{GRS}_{n-t}(\boldsymbol{\alpha}, \boldsymbol{v}))$ in general is not contained in $\mathrm{GRS}_{n-t}(\boldsymbol{\alpha}, \boldsymbol{v})$, hence

$$C \star \mathrm{Tr}(D^\perp) \not\subseteq C \star D^\perp$$

in general.

4.1 Computational Complexity

To assess the advantages of the scheme $\mathscr{S}|_{\mathbb{F}_q}$ compared to $\mathscr{S}_{FGHK}$ using GRS codes an analysis on the complexity of the file retrieval will be made on both schemes. It is during file retrieval where the majority of the computational cost of the scheme $\mathscr{S}_{FGHK}$ lies.

We will start by presenting the complexity of finite field arithmetic. For a reference, see e.g. [7] and [12].

Lemma 4.11. *Let $\mathbb{F}_p$ be a prime field. Then addition and subtraction in $\mathbb{F}_p$ has bit complexity $\mathcal{O}(\log(p))$, and multiplication and division in $\mathbb{F}_p$ has bit complexity $\mathcal{O}(\log(p)^2)$.*

Lemma 4.12. *Let $\mathbb{F}_{p^m}$ be a finite field for p a prime. Then addition and subtraction in $\mathbb{F}_{p^m}$ takes $\mathcal{O}(m)$ operations in $\mathbb{F}_p$, and multiplication and division in $\mathbb{F}_{p^m}$ takes $\mathcal{O}(m^2)$ operations in $\mathbb{F}_p$.*

It should be noted that the multiplication algorithm yielding complexity $\mathcal{O}(m^2 \log(p)^2)$ is naïve multiplication. More efficient algorithms exists for multiplication of both large polynomials and integers (see e.g. [7]). However for small fields naïve multiplication will be fastest. Hence, we will consider the complexities from Lemma 4.12.

Proposition 4.13. *The response computation of $\mathscr{S}_{FGHK}$ during one iteration requires*

$$n(Mb-1)\mathcal{O}(m\log(q)) + nMb\mathcal{O}(m^2\log(q)^2) = nMb\mathcal{O}(m^2\log(q)^2)$$

bit operations.

Proof. During a single iteration of response calculation n response entries of the response vector $\boldsymbol{r}^i$ are calculated as

$$r_j^i = \langle \boldsymbol{q}_j^i, \boldsymbol{y}_j \rangle$$

as in (3.3). This dot product consists of $Mb-1$ additions in $\mathbb{F}_{q^m}$ and Mb multiplications in $\mathbb{F}_{q^m}$. Hence, the additions has complexity $n(Mb-1)\mathcal{O}(m\log(q))$ and the multiplications has complexity $nMb\mathcal{O}(m^2\log(q)^2)$ concluding our proof. ■

Proposition 4.14. *The response computation of $\mathscr{S}|_{\mathbb{F}_q}$ during one iteration requires*

$$n(Mb-1)\mathcal{O}(m\log(q)) + nMb\mathcal{O}(m\log(q)^2) = nMb\mathcal{O}(m\log(q)^2)$$

bit operations.

Proof. As in the previous proof n response entries of the response vector $\boldsymbol{r}^i$ are calculated as

$$r_j^i = \langle \boldsymbol{q}_j^i, \boldsymbol{y}_j \rangle.$$

The $n(Mb-1)$ additions in $\mathbb{F}_{q^m}$ has complexity $n(Mb-1)\mathcal{O}(m\log(q))$, but the Mb multiplications in the inner product are between an element of $\mathbb{F}_q$ and an element of $\mathbb{F}_{q^m}$. Such a multiplication has complexity $\mathcal{O}(m\log(q)^2)$. Hence the multiplications has complexity $nMb\mathcal{O}(m\log(q)^2)$. ■

A particularly large improvement is made when $q=2$. Then only the $n(Mb-1)$ additions in $\mathbb{F}_{2^m}$ remains yielding a bit complexity of $n(Mb-1)\mathcal{O}(m)$ for each iteration of the response computation.

A Scheme With Byzantine and Unresponsive Servers

In this section we will describe a generalisation of the scheme $\mathscr{S}_{FGHK}$ that takes account for both a certain amount of servers that are unresponsive to the sent queries as well as a subset of servers with malicious intent, i.e., servers that reply with an incorrect response. The scheme described is that of [19] and we will denote it by $\mathscr{S}_{TGKFH}$ as an homage to the authors of [19].

The files in this scheme are stored using a DSS $\mathscr{D}$ as described in Definition 2.2. Any server that does not respond to a query we call *unresponsive*, and a server responding with an incorrect response we call *byzantine*. During each iteration of the PIR scheme it is assumed that β byzantine servers responding with any element of $\mathbb{F}_{q^m}$ as well as r unresponsive servers are present. The response of an unresponsive server is replaced with an erasure symbol ?.

Although this is a generalisation of the scheme $\mathscr{S}_{FGHK}$ using GRS codes we will change our viewpoint slightly. We will furthermore only consider Reed-Solomon storage codes $\mathrm{RS}_k(\boldsymbol{\alpha}) := \mathrm{GRS}_k(\boldsymbol{\alpha}, \mathbf{1})$. We consider again the evaluation homomorphism $\mathrm{ev}_{\boldsymbol{\alpha}} := \mathrm{ev}_{\boldsymbol{\alpha},\mathbf{1}}$ from (3.6). As in Example 3.14 we can choose a basis for $\mathrm{RS}_k(\boldsymbol{\alpha}) = \mathrm{Im}(\mathrm{ev}_{\boldsymbol{\alpha}})$ as

$$\mathrm{ev}_{\boldsymbol{\alpha}}(1), \mathrm{ev}_{\boldsymbol{\alpha}}(X), \ldots, \mathrm{ev}_{\boldsymbol{\alpha}}(X^{k-1}).$$

Note that $\mathbb{F}_{q^m}[X]^k$ and $\mathrm{RS}_k(\boldsymbol{\alpha})$ are in fact isomorphic and

$$\mathrm{ev}_{\boldsymbol{\alpha}}^{-1} : \mathrm{RS}_k(\boldsymbol{\alpha}) \to \mathbb{F}_{q^m}[X]^k$$

is given by Lagrange interpolation.

For a file x^i, we denote the a^{th} row of x^i by $\boldsymbol{x}_a^i = (x_{a,0}^i, \ldots, x_{a,k-1}^i) \in (\mathbb{F}_{q^m})^k$. Using the canonical generator matrix G_C of (3.15) for $C = \mathrm{RS}_k(\boldsymbol{\alpha})$ to encode $\boldsymbol{x}_a^i$ ensues

$$\boldsymbol{x}_a^i G_C = \mathrm{ev}_{\boldsymbol{\alpha}}(f_a^i) \tag{5.1}$$

where $f_a^i \in \mathbb{F}_{q^m}[X]^k$ is given by

$$f_a^i(X) = x_{a,0}^i + x_{a,1}^i X + \cdots + x_{a,k-1}^i X^{k-1}. \tag{5.2}$$

Thus, for an encoded file $y^i = x^i G_C$ we have

$$y^i = x^i G_C = \begin{bmatrix} f_1^i(\alpha_1) & \ldots & f_1^i(\alpha_n) \\ \vdots & \ddots & \vdots \\ f_b^i(\alpha_1) & \ldots & f_b^i(\alpha_n) \end{bmatrix}.$$

The representation of a file as coefficients of a polynomial as in (5.2) will be central for this scheme. Let us now present our scheme according to the four steps of Definition 2.4. First, an auxiliary constant $\rho := n - (k + t + 2\beta + r - 1)$ is defined. This is the amount of information symbols retrieved in each iteration of the scheme which we will elaborate upon soon. As in the scheme $\mathscr{S}_{FGHK}$ we would like to retrieve exactly one

file during the s iterations of the scheme. Hence, we enforce that $b = \operatorname{lcm}(\rho, k)/k$ and $s = \operatorname{lcm}(\rho, k)/\rho$ ensuring that $bk = s\rho$.

Query Construction: Let $\sigma \in [s]$ denote the current iteration index and let $D := \mathrm{RS}_t(\boldsymbol{\alpha})$ be our retrieval code. We choose Mb codewords $\boldsymbol{d}_a^{l,(\sigma)} \in D$, for $l \in [M]$ and $a \in [b]$ independently and uniformly at random and define

$$d_a^{l,(\sigma)}(X) := \mathrm{ev}_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{d}_a^{l,(\sigma)}) \in \mathbb{F}_{q^m}[X]^t. \tag{5.3}$$

We define

$$e_j(X) := \begin{cases} X^j & \text{if } j \geq t, \\ 0 & \text{if } j < t. \end{cases}$$

Suppose that we are interested in the retrieval of the file x^i. Then we define

$$q_a^{l,(\sigma)}(X) := \begin{cases} d_a^{l,(\sigma)}(X) + e_{\sigma\rho+(1-a)k+t-1}(X) & \text{if } l = i, \\ d_a^{l,(\sigma)}(X) & \text{if } l \neq i. \end{cases} \tag{5.4}$$

The evaluations of these polynomials are sent to the servers. More precisely, the evaluations

$$\mathrm{ev}_{\boldsymbol{\alpha}}(q_a^{l,(\sigma)}) = \left(q_a^{l,(\sigma)}(\alpha_1), \ldots, q_a^{l,(\sigma)}(\alpha_n)\right)$$

are collected in a matrix

$$\begin{bmatrix} q_1^{1,(\sigma)}(\alpha_1) & \ldots & q_b^{1,(\sigma)}(\alpha_1) & \ldots & q_1^{M,(\sigma)}(\alpha_1) & \ldots & q_b^{M,(\sigma)}(\alpha_1) \\ q_1^{1,(\sigma)}(\alpha_2) & \ldots & q_b^{1,(\sigma)}(\alpha_2) & \ldots & q_1^{M,(\sigma)}(\alpha_2) & \ldots & q_b^{M,(\sigma)}(\alpha_2) \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ q_1^{1,(\sigma)}(\alpha_n) & \ldots & q_b^{1,(\sigma)}(\alpha_n) & \ldots & q_1^{M,(\sigma)}(\alpha_n) & \ldots & q_b^{M,(\sigma)}(\alpha_n) \end{bmatrix} \begin{matrix} \boldsymbol{q}_1^{(\sigma)} \\ \boldsymbol{q}_2^{(\sigma)} \\ \vdots \\ \boldsymbol{q}_n^{(\sigma)} \end{matrix} \tag{5.5}$$

as in (3.18) and the row $\boldsymbol{q}_j^{(\sigma)} \in (\mathbb{F}_{q^m})^{Mb}$ is sent to the j^{th} server for each $j \in [n]$.

Response Computation and Iteration: The j^{th} server computes

$$r_j^{(\sigma)} = \langle \boldsymbol{q}_j^{(\sigma)}, \boldsymbol{y}_j \rangle \tag{5.6}$$

and replies according to its nature. In particular, if the server is byzantine it chooses an element $x \in \mathbb{F}_{q^m}$ uniformly at random and replies with this element. If the server is unresponsive it will not reply. If the server is neither byzantine nor unresponsive it replies with $r_j^{(\sigma)}$.

If $\sigma < s$ we increase σ by 1 and **Query Construction** and **Response Computation** are repeated. Otherwise we pass to file reconstruction.

Reconstruction: The reconstruction of the file entries from the first iteration and the remaining iterations are slightly different and we consider them separately. Hence, assume that $\sigma = 1$ and we have received the total response vector $\tilde{\boldsymbol{r}}^{(1)}$ which has the j^{th} server response as its j^{th} entry. The *true response* $\boldsymbol{r}^{(1)} = \left(\langle \boldsymbol{q}_1^{(1)}, \boldsymbol{y}_1 \rangle, \ldots, \langle \boldsymbol{q}_n^{(1)}, \boldsymbol{y}_n \rangle\right)$ is

now recovered from $\tilde{\boldsymbol{r}}^{(1)}$. The *response polynomial* $r^{(1)}$ is defined as $r^{(1)} = \mathrm{ev}_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{r}^{(1)})$, and it is decomposed as

$$r^{(1)}(X) = \underbrace{g^{(1)}(X)}_{\deg < k+t-1} + X^{k+t-1}h^{i,(1)}(X).$$

Now, suppose that we have found $h^{i,(\varsigma)}$ for $\varsigma = 1, \ldots, \sigma-1$ for $\sigma > 1$. In the σ^{th} iteration we receive the response vector $\tilde{\boldsymbol{r}}^{(\sigma)}$ and define

$$\tilde{\boldsymbol{r}}'^{(\sigma)} = \tilde{\boldsymbol{r}}^{(\sigma)} - \sum_{\varsigma=1}^{\sigma} \mathrm{ev}_{\boldsymbol{\alpha}}(X^{k+t-1+\rho(\sigma-\varsigma)}h^{i,(\varsigma)}).$$

$\tilde{\boldsymbol{r}}'^{(\sigma)}$ is an element of $\mathrm{RS}_{\rho+k+t-1}$, hence errors and erasures can be corrected yielding

$$\boldsymbol{r}'^{(\sigma)} = \boldsymbol{r}^{(\sigma)} - \sum_{\varsigma=1}^{\sigma} \mathrm{ev}_{\boldsymbol{\alpha}}(X^{k+t-1+\rho(\sigma-\varsigma)}h^{i,(\varsigma)}), \tag{5.7}$$

where $\boldsymbol{r}^{(\sigma)}$ is the true response for the σ^{th} iteration. The polynomial $h^{i,(\sigma)}$ is now defined as

$$\mathrm{ev}_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{r}'^{(\sigma)}) = r'^{(\sigma)}(X) = \underbrace{g^{(\sigma)}(X)}_{\deg < k+t-1} + X^{k+t-1}h^{i,(\sigma)}.$$

This procedure is repeated for all $\sigma = 2, \ldots, s$ until we can recover the file x^i by the identity

$$\sum_{a=1}^{b} X^{k(b-a)} f_a^i(X) = \sum_{\sigma=1}^{s} X^{\rho(s-\sigma)} h^{i,(\sigma)}, \tag{5.8}$$

where the polynomials f_a^i are as in (5.1) and (5.2) since $\deg(f_a^i) < k$ and $\deg(h^{i,(\sigma)}) < \rho$.

It is still not entirely clear why this scheme works, and that it is in fact t-private. We will prove this by a series of lemmata.

In the following lemma we will assume that we have neither byzantine nor unresponsive servers in our DSS, but that this fact is unknown to the user, i.e., the scheme should still offer protection against β byzantine and r unresponsive servers.

Lemma 5.1. *If we have no byzantine or unresponsive servers, the identity* (5.8) *is correct, i.e., $\mathscr{S}_{TGKFH}$ is a PIR-scheme.*

Proof. Suppose that we are interested in the retrieval of the i^{th} file, i.e., the queries are defined as in (5.4). In the σ^{th} iteration of the scheme the response in case of no byzantine nor unresponsive servers the received response from server j is given as in (5.6) by

$$\begin{aligned} r_j^{(\sigma)} &= \langle \boldsymbol{q}_j^{(\sigma)}, \boldsymbol{y}_j \rangle \\ &= \sum_{l=1}^{M} \sum_{a=1}^{b} q_a^{l,(\sigma)}(\alpha_j) f_a^l(\alpha_j) \\ &= \sum_{l=1}^{M} \sum_{a=1}^{b} d_a^{l,(\sigma)}(\alpha_j) f_a^l(\alpha_j) + \sum_{a=1}^{b} e_{\sigma\rho-ak+k+t-1}(\alpha_j) f_a^i(\alpha_j). \end{aligned} \tag{5.9}$$

To identify when $e_{\sigma\rho-ak+k+t-1} \not\equiv 0$ consider

$$\sigma\rho - ak + k + t - 1 \geq t \Leftrightarrow \frac{\sigma\rho}{k} \geq a - 1 + \frac{1}{k},$$

and we get that $e_{\sigma\rho-ak+k+t-1} \not\equiv 0$ exactly when $\lceil \sigma\rho/k \rceil \geq a$. Hence, from (5.9) we get

$$r_j^{(\sigma)} = \sum_{l=1}^{M} \sum_{a=1}^{b} d_a^{l,(\sigma)}(\alpha_j) f_a^l(\alpha_j) + \sum_{a=1}^{\lceil \sigma\rho/k \rceil} \alpha_j^{\sigma\rho-ak+k+t-1} f_a^i(\alpha_j).$$

Thus, the true response vector $\boldsymbol{r}^{(\sigma)}$ can be expressed as the evaluation $\boldsymbol{r}^{(\sigma)} = \mathrm{ev}_{\boldsymbol{\alpha}}(r^{(\sigma)})$ where

$$r^{(\sigma)}(X) = \underbrace{\sum_{l=1}^{M} \sum_{a=1}^{b} d_a^{l,(\sigma)}(X) f_a^l(X)}_{=g^{(\sigma)}(X)} + \sum_{a=1}^{\lceil \sigma\rho/k \rceil} X^{\sigma\rho-ak+k+t-1} f_a^i(X). \tag{5.10}$$

The codeword $\mathrm{ev}_{\boldsymbol{\alpha}}(g^{(\sigma)})$ is contained in $C \star D$ exactly as the corresponding codeword in the scheme $\mathscr{S}_{FGHK}$, hence, $\deg(g^{(\sigma)}) < k + t - 1$ by Proposition 3.9.

The reconstruction algorithm is split into two parts, $\sigma = 1$ and $\sigma > 1$. For the former part, suppose that $\sigma = 1$. Then by (5.10) the response polynomial $r^{(1)}$ is given by

$$\begin{aligned} r^{(1)}(X) &= g^{(1)}(X) + \sum_{a=1}^{\lceil \rho/k \rceil} X^{\rho-ak+k+t-1} f_a^i(X) \\ &= g^{(1)}(X) + X^{k+t-1+\rho-\lceil \rho/k \rceil k} f_{\lceil \rho/k \rceil}(X) + X^{k+t-1} \sum_{a=1}^{\lceil \rho/k \rceil - 1} X^{\rho-ak} f_a^i(X) \end{aligned} \tag{5.11}$$

We wish to split expression into terms of degree $< k+t-1$ and terms of degree $\geq k+t-1$. The first term $g^{(1)}$ has degree $< k + t - 1$ and the last term has degree $\geq k + t - 1$. For the middle term we get

$$\begin{aligned} X^{k+t-1+\rho-\lceil \rho/k \rceil k} f_{\lceil \rho/k \rceil}(X) &= X^{k+t-1+\rho-\lceil \rho/k \rceil k} \sum_{\kappa=0}^{k-1} x^i_{\lceil \rho/k \rceil, \kappa} X^\kappa \\ = \underbrace{X^{k+t-1+\rho-\lceil \rho/k \rceil k} \sum_{\kappa=0}^{\lceil \rho/k \rceil k - \rho - 1} x^i_{\lceil \rho/k \rceil, \kappa} X^\kappa}_{=\gamma^{(1)}(X),\ \deg(\gamma^{(1)}(X)) < k+t-1} &+ \underbrace{X^{k+t-1+\rho-\lceil \rho/k \rceil k} \sum_{\kappa=\lceil \rho/k \rceil k - \rho}^{k-1} x^i_{\lceil \rho/k \rceil, \kappa} X^\kappa}_{\deg \geq k+t-1}, \end{aligned} \tag{5.12}$$

using that the polynomials f_a^i are defined as in (5.2). By combining (5.11) and (5.12) it is now straight-forward to decompose $r^{(1)}$ into terms of degree $< k + t - 1$ and terms of

degree $\geq k+t-1$:

$$r^{(1)}(X) = \underbrace{g^{(1)}(X) + \gamma^{(1)}(X)}_{\deg < k+t-1} + X^{k+t-1} \underbrace{\left(\sum_{\kappa=\lceil \rho/k \rceil k-\rho}^{k-1} x^i_{\lceil \rho/k \rceil, \kappa} X^{\rho - \lceil \rho/k \rceil k + \kappa} + \sum_{a=1}^{\lceil \rho/k \rceil - 1} X^{\rho - ak} f^i_a(X) \right)}_{= h^{i,(1)}(X)}. \tag{5.13}$$

Suppose now that $\sigma > 1$, and that we have found the polynomials $h^{i,(\varsigma)}$ for $\varsigma = 1, \ldots, \sigma - 1$. Then $h^{i,(\sigma)}$ with $\deg(h^{i,(\sigma)}) < \rho$ is defined as the polynomial satisfying

$$r^{(\sigma)}(X) = \underbrace{g^{(\sigma)}(X)}_{\deg < k+t-1} + X^{k+t-1} \sum_{\varsigma=1}^{\sigma} X^{\rho(\sigma-\varsigma)} h^{i,(\varsigma)}(X), \tag{5.14}$$

hence, $h^{i,(\sigma)}$ is determined as

$$X^{k+t-1} h^{i,(\sigma)}(X) = r^{(\sigma)}(X) - g^{(\sigma)}(X) - X^{k+t-1} \sum_{\varsigma=1}^{\sigma-1} X^{\rho(\sigma-\varsigma)} h^{i,(\varsigma)}(X).$$

In the last iteration of the scheme we receive the response $\boldsymbol{r}^{(s)}$ with corresponding response polynomial $r^{(s)}$. This polynomial is by (5.10) and (5.14) given as

$$\begin{aligned} r^{(s)}(X) &= g^{(s)}(X) + X^{k+t-1} \sum_{a=1}^{b} X^{k(b-a)} f^i_a(X) \\ &= g^{(s)}(X) + X^{k+t-1} \sum_{\sigma=1}^{s} X^{\rho(s-\sigma)} h^{i,(\sigma)}(X), \end{aligned}$$

whence the identity (5.8) follows. ■

In (5.13) we can clearly see what entries are downloaded from each row in the first iteration. From the rows $a = 1, \ldots, \lceil \rho/k \rceil - 1$ we download the entire row as the coefficients of the polynomials f^i_a, and from the $\lceil \rho/k \rceil^{\text{th}}$ row we download the elements $x^i_{\lceil \rho/k \rceil, \kappa}$ for $\kappa = \lceil \rho/k \rceil k - \rho, \ldots, k-1$, yielding in total

$$k(\lceil \rho/k \rceil - 1) + k - (\lceil \rho/k \rceil k - \rho) = \rho$$

downloaded symbols. This shows how elements are downloaded from different rows of a file during an iteration of the scheme in case we do not download the entirety of a file. Note that it is the polynomial $g^{(\sigma)}$ that encapsulates the randomness added in each iteration.

Lemma 5.2. *During each iteration $\sigma = 1, \ldots, s$ of the scheme $\mathscr{S}_{TGHFK}$ the presence of β byzantine servers and r unresponsive servers can be handled. More precisely, if the response vector $\tilde{\boldsymbol{r}}^{(\sigma)}$ containing the j^{th} server response in its j^{th} entry only differ from the true response $\boldsymbol{r}^{(\sigma)} = \left(\langle \boldsymbol{q}_1^{(\sigma)}, \boldsymbol{y}_1 \rangle, \ldots, \langle \boldsymbol{q}_n^{(\sigma)}, \boldsymbol{y}_n \rangle\right)$ by at most r erasures and β errors then the polynomial $h^{i,(\sigma)}$ can be recovered from $\tilde{\boldsymbol{r}}^{(\sigma)}$.*

Proof. Suppose that $\sigma = 1$. The true response $\boldsymbol{r}^{(1)}$ has corresponding response polynomial

$$\mathrm{ev}_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{r}^{(1)}) = r^{(1)}(X) = \underbrace{g^{(1)}(X)}_{\deg < k+t-1} + X^{k+t-1} \underbrace{h^{i,(1)}(X)}_{\deg < \rho},$$

hence, $\boldsymbol{r}^{(1)}$ is contained in $\mathrm{RS}_{\rho+k+t-1}(\boldsymbol{\alpha})$ which has minimum distance

$$n - \rho - k - t + 2 = 2\beta + r + 1,$$

and β errors and r erasures can be corrected.

Now, suppose that $\sigma > 1$. Here, the true response $\boldsymbol{r}^{(\sigma)}$ has corresponding response polynomial

$$r^{(\sigma)}(X) = \underbrace{g^{(\sigma)}(X)}_{\deg < k+t-1} + X^{k+t-1} \sum_{\varsigma=1}^{\sigma} X^{\rho(\sigma-\varsigma)} \underbrace{h^{i,(\varsigma)}(X)}_{\deg < \rho},$$

but for $\varsigma = 1, \ldots, \sigma - 1$ the polynomials $h^{i,(\varsigma)}$ are known and their evaluations can be subtracted from $\boldsymbol{r}^{(\sigma)}$ as in (5.7) yielding $\boldsymbol{r}'^{(\sigma)}$. The codeword $\boldsymbol{r}'^{(\sigma)}$ is an element of the same code $\mathrm{RS}_{\rho+k+t-1}(\boldsymbol{\alpha})$ as in the first iteration. Thus, by the same argument, β errors and r erasures can be corrected in the σ^{th} iteration. ∎

Lemma 5.3. *The scheme $\mathscr{S}_{TGHFK}$ protects against t colluding servers.*

Proof. This follows as the proof of Theorem 3.5. ∎

Theorem 5.4. *The scheme $\mathscr{S}_{TGHFK}$, using storage code $C = \mathrm{RS}_k(\boldsymbol{\alpha})$ to store M files on n servers, with the retrieval code $D = \mathrm{RS}_t(\boldsymbol{\alpha})$ is a t-private PIR scheme that in each iteration protects against the presence of β byzantine and r unresponsive servers when $n > k + t + 2\beta + r - 1$. This scheme has rate*

$$R = \frac{bk}{s(n-r)} = \frac{\rho}{n-r} = \frac{n - (k + t + 2\beta + r - 1)}{n - r}.$$

Proof. By Lemma 5.1 and 5.2 the scheme $\mathscr{S}_{TGKFH}$ is a PIR-scheme that in each iteration offers protection against β byzantine and r unresponsive servers. By Lemma (5.3) the scheme is furthermore t-private. The rate follows by Definition 2.7. ∎

We have previously claimed that this is a generalisation of the scheme $\mathscr{S}_{FGHK}$. This is the case inasmuch as the two schemes yield the same rate if $\beta, r = 0$, however there is a difference in how we define the queries. In the scheme $\mathscr{S}_{FGHK}$ we add an error to the retrieval code codeword yielding the queries, and in $\mathscr{S}_{TGKFH}$ we add an element of a higher dimension RS code which is not in the retrieval code to the retrieval code codeword ensuring that the queries are still elements of an RS code of the same length. We will consider an example on how these schemes differ during file retrieval.

Example 5.5. Suppose that we have some DSS with $k = 8$ distributing n files on M servers. For comparisons sake we assume that there are no byzantine nor unresponsive servers. Let us first consider how the scheme $\mathscr{S}_{FGHK}$ retrieves the i^{th} file. Let $b = 3$, $c = \rho = 6$, and let $J = [8]$ be the index set from which symbols are downloaded from the encoded file $y^i = x^i G_C$. In Figure 3 we can see how symbols are downloaded from this scheme.

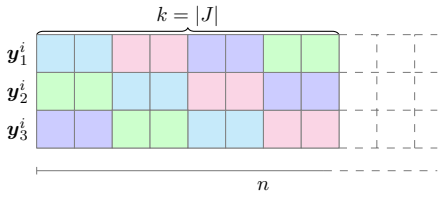

Figure 3: How symbols are downloaded from the rows of the encoded file y^i in the scheme $\mathscr{S}_{FGHK}$. The entries ▭ are downloaded in iteration 1, ▭ in iteration 2, ▭ in iteration 3, and ▭ in iteration 4.

In Figure 4 we can see how the scheme $\mathscr{S}_{TGKFH}$ with the exact same parameters downloads the i^{th} file x^i as the coefficients of the polynomials f_a^i.

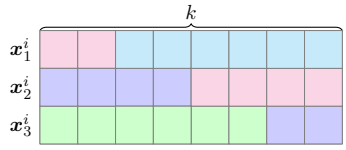

Figure 4: How symbols are downloaded from the rows of the file x^i in the scheme $\mathscr{S}_{TGKFH}$. The entries ▭ are downloaded in iteration 1, ▭ in iteration 2, ▭ in iteration 3, and ▭ in iteration 4.

Figure 4 also enlightens on how the polynomials f_a^i are represented as polynomials $h^{i,(\sigma)}$ as in (5.8). In Figure 5 we can see exactly how this is done.

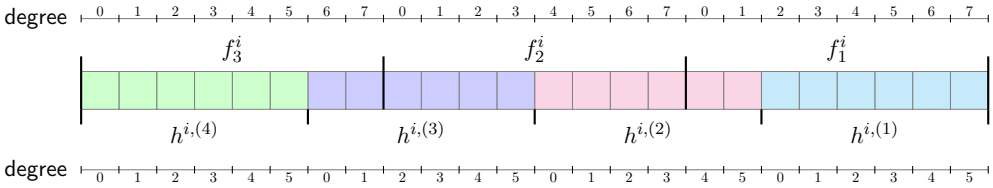

Figure 5: How the rows of x^i as the polynomials f_1^i, f_2^i, and f_3^i are decomposed as the polynomials $h^{i,(\sigma)}$. The colours correspond to the downloaded symbols in the four iterations as in Figure 4.

We have seen now how the schemes $\mathscr{S}_{FGHK}$ and $\mathscr{S}_{TGHFK}$ differ. Figure 6 presents the general idea of the scheme $\mathscr{S}_{TGKFH}$ in the case that $k > \rho$.

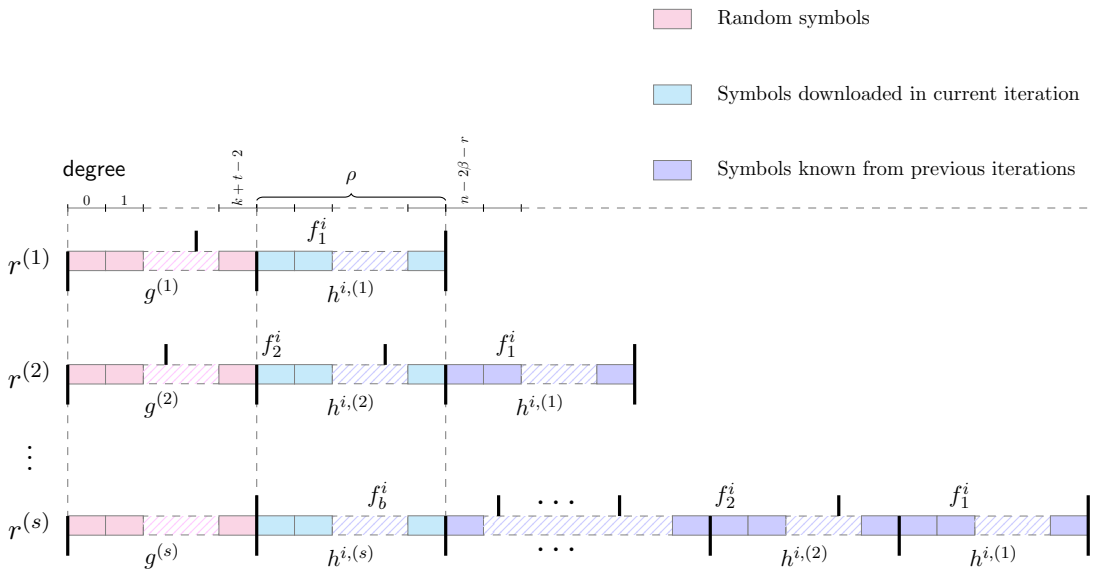

Figure 6: The scheme $\mathscr{S}_{TGKFH}$.

For further illumination of the scheme $\mathscr{S}_{TGKFH}$ let us consider an example of the scheme in action.

Example 5.6. The parameters of our example of the scheme $\mathscr{S}_{TGKFH}$ can be seen in Table 2.

n	M	k	t	r	β	b	s	ρ	C	D
11	4	3	3	2	1	2	3	2	$\mathrm{RS}_3(\boldsymbol{\alpha})$	$\mathrm{RS}_3(\boldsymbol{\alpha})$

Table 2: Scheme parameters for our example of the scheme $\mathscr{S}_{TGKFH}$.

The codes C and D will be codes over $\mathbb{F}_{11}$ and $\boldsymbol{\alpha} = (0, 1, \ldots, 10)$, and we will have 11 servers $s_1, \ldots, s_{11}$. We are interested in applying the scheme on the same four files as

in Example 3.14. That is $x^1, \ldots, x^4$ are the files of (3.14) now arranged as

$$x^1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, x^2 = \begin{bmatrix} 2 & 4 & 0 \\ 2 & 4 & 0 \end{bmatrix}, x^3 = \begin{bmatrix} 1 & 3 & 5 \\ 0 & 2 & 4 \end{bmatrix}, x^4 = \begin{bmatrix} 2 & 5 & 3 \\ 6 & 0 & 1 \end{bmatrix}.$$

Using the canonical generator matrix G_C for C as given in (3.17) to encode the files $x^1, \ldots, x^4$ we get

$$Y = XG_C = \begin{bmatrix} 1 & 6 & 6 & 1 & 2 & 9 & 0 & 8 & 0 & 9 & 2 \\ 4 & 4 & 5 & 7 & 10 & 3 & 8 & 3 & 10 & 7 & 5 \\ 2 & 6 & 10 & 3 & 7 & 0 & 4 & 8 & 1 & 5 & 9 \\ 2 & 6 & 10 & 3 & 7 & 0 & 4 & 8 & 1 & 5 & 9 \\ 1 & 9 & 5 & 0 & 5 & 9 & 1 & 3 & 4 & 4 & 3 \\ 0 & 6 & 9 & 9 & 6 & 0 & 2 & 1 & 8 & 1 & 2 \\ 2 & 10 & 2 & 0 & 4 & 3 & 8 & 8 & 3 & 4 & 0 \\ 6 & 7 & 10 & 4 & 0 & 9 & 9 & 0 & 4 & 10 & 7 \\ \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 & \boldsymbol{y}_4 & \boldsymbol{y}_5 & \boldsymbol{y}_6 & \boldsymbol{y}_7 & \boldsymbol{y}_8 & \boldsymbol{y}_9 & \boldsymbol{y}_{10} & \boldsymbol{y}_{11} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ y^1 = x^2 G_C \\ \\ y^2 = x^2 G_C \\ \\ y^3 = x^3 G_C \\ \\ y^4 = x^4 G_C \end{matrix}$$

Suppose as in Example 3.14 that we wish to retrieve the file x^3. Furthermore, suppose that servers s_3 and s_4 are unresponsive, and that s_7 is byzantine. Since we have 3 iterations in the scheme and 2 rows of each file, we will decompose the file of interest x^3 according to its file polynomials f_a^3 as in (5.2):

$$\sum_{a=1}^{b} X^{3(b-a)} f_a^3(X) = \sum_{\sigma=1}^{s} X^{\rho(s-\sigma)} h^{3,(\sigma)}(X),$$

or more explicitly

$$X^3 \underbrace{(1 + 3X + 5X^2)}_{=f_1^3} + \underbrace{(2X + 4X^2)}_{=f_2^3} = X^4 \underbrace{(3 + 5X)}_{=h^{3,(1)}} + X^2 \underbrace{(4 + X)}_{=h^{3,(2)}} + \underbrace{(2X)}_{=h^{3,(3)}} \tag{5.15}$$

We consider now the first iteration of the scheme. For query construction 8 codewords $\boldsymbol{d}_a^{l,(1)}$ are chosen uniformly and independently at random from the retrieval code D yielding 8 polynomials $d_a^{l,(1)} = \mathrm{ev}_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{d}_a^{l,(1)})$ as in (5.3). This is done by choosing elements $\boldsymbol{v}$ from $(\mathbb{F}_{11})^3$ uniformly at random and letting $d_a^{l,(1)}$ be the images of these elements under the mapping $\boldsymbol{v} = (v_0, v_1, v_2) \mapsto v_0 + v_1 X + v_2 X^2$. These polynomials are

$$\begin{aligned} d_1^{1,(1)}(X) &= 3 + 7X + 3X^2, & d_1^{3,(1)}(X) &= 10 + 9X, \\ d_2^{1,(1)}(X) &= 5 + 7X + 6X^2, & d_2^{3,(1)}(X) &= 2 + 9X + 5X^2, \\ d_1^{2,(1)}(X) &= 4 + 5X + 9X^2, & d_1^{4,(1)}(X) &= 3 + 8X + 8X^2, \\ d_2^{2,(1)}(X) &= 10 + 4X + 9X^2, & d_2^{4,(1)}(X) &= 5 + 7X + X^2. \end{aligned}$$

As in (5.4) the polynomials $q_a^{l,(1)}$ are defined as $q_a^{l,(1)} := d_a^{l,(1)}$ for $l \neq 3$ and for $l = 3$ we get

$$q_1^{3,(1)}(X) := \underbrace{10 + 9X}_{=d_1^{3,(1)}} + X^4, \qquad q_2^{3,(1)}(X) := \underbrace{2 + 9X + 5X^2}_{=d_2^{3,(1)}}.$$

The evaluations $\boldsymbol{q}_a^{l,(1)} = \mathrm{ev}_{\boldsymbol{\alpha}}(q_a^{l,(1)})$ of these polynomials comprises a matrix as in (5.5) and the j^{th} row of this matrix $\boldsymbol{q}_j^{(1)}$ is sent to the j^{th} server, which replies accordingly. The servers s_3 and s_4 does not reply at all and the server s_7 replies with the randomly chosen element 1. The remaining servers s_j respondes with $r_j^{(1)} = \langle \boldsymbol{q}_j^{(1)}, \boldsymbol{y}_j \rangle$. The true response vector and the received response are respectively

$$\begin{aligned}\boldsymbol{r}^{(1)} &= (9,7,9,2,7,1,7,0,3,10,0)\,,\\ \tilde{\boldsymbol{r}}^{(1)} &= (9,7,?,?,7,1,1,0,3,10,0)\,.\end{aligned}$$

We know that $\boldsymbol{r}^{(1)}$ is an element of $\mathrm{RS}_7(\boldsymbol{\alpha})$ and can thus correct the $r = 2$ erasures and the $\beta = 1$ error from the servers since $d_{\mathrm{RS}_7(\boldsymbol{\alpha})} = 5$. Hence, by decoding we receive $\boldsymbol{r}^{(1)}$ which interpolates to

$$\begin{aligned}r^{(1)}(X) = \mathrm{ev}_{\boldsymbol{\alpha}}^{-1}(\boldsymbol{r}^{(1)}) &= 9 + 3X^2 + 3X^3 + 3X^4 + 3X^5 + 5X^6\\ &= g^{(1)}(X) + X^5 \underbrace{(3+5X)}_{=h^{3,(1)}},\end{aligned}$$

and we can pass to the second iteration.

Eight polynomials $d_a^{l,(2)}$ are chosen at random as in the first iteration:

$$\begin{aligned} d_1^{1,(2)}(X) &= 7X, & d_1^{3,(2)}(X) &= 1 + 8X + X^2,\\ d_2^{1,(2)}(X) &= 1 + 4X + X^2, & d_2^{3,(2)}(X) &= 2 + 5X + 8X^2,\\ d_1^{2,(2)}(X) &= 2 + 3X + 7X^2, & d_1^{4,(2)}(X) &= 1 + 7X + 7X^2,\\ d_2^{2,(2)}(X) &= 6X + 4X^2, & d_2^{4,(2)}(X) &= 9 + 7X. \end{aligned}$$

As in the first iteration, we let $q_a^{l,(2)} := d_a^{l,(2)}$ for $l \neq 3$ and for $l = 3$ we let

$$q_1^{3,(2)} := \underbrace{1 + 8X + X^2}_{=d_1^{3,(2)}} + X^6, \qquad q_2^{3,(1)} := \underbrace{2 + 5X + 8X^2}_{=d_2^{3,(2)}} + X^3.$$

The evaluations of these polynomials $q_a^{l,(2)}$ are sent as queries to the servers as in the first iteration. The servers reply accordingly, and we get true response vector and received responses respectively as

$$\begin{aligned}\boldsymbol{r}^{(2)} &= (0,6,2,4,10,9,8,0,8,2,6)\,,\\ \tilde{\boldsymbol{r}}^{(2)} &= (0,6,?,?,10,9,2,0,8,2,6)\,.\end{aligned}$$

We subtract $\mathrm{ev}_{\boldsymbol{\alpha}}(X^7 \boldsymbol{h}^{3,(1)})$ from $\tilde{\boldsymbol{r}}^{(2)}$ to receive $\tilde{\boldsymbol{r}}'^{(2)} = \tilde{\boldsymbol{r}}^{(2)} - \mathrm{ev}_{\boldsymbol{\alpha}}(X^7 h^{3,(1)})$. This is a codeword of $\mathrm{RS}_7(\boldsymbol{\alpha})$ and the $r = 2$ erasures and $\beta = 1$ error can be corrected. Hence, we get

$$\boldsymbol{r}'^{(2)} = \boldsymbol{r}^2 - \mathrm{ev}_{\boldsymbol{\alpha}}(X^7 h^{3,(1)}) = (0,9,10,7,5,2,8,3,10,8,4)\,,$$

and by interpolating

$$\begin{aligned} r^{(2)}(X) &= 10X + X^3 + 4X^5 + X^6 + 3X^7 + 5X^8 \\ &= g^{(2)}(X) + X^5 \underbrace{(4+X)}_{=h^{3,(2)}} + X^7 \underbrace{(3+5X)}_{h^{3,(1)}}. \end{aligned}$$

The third iteration is done analogously and we find the retrieval polynomial

$$r^{(3)}(X) = g^{(3)}(X) + X^5 \underbrace{(2X)}_{=h^{3,(3)}} + X^7 \underbrace{(4+X)}_{=h^{3,(2)}} + X^9 \underbrace{(3+5X)}_{=h^{3,(1)}}.$$

By (5.15) the polynomials $h^{3,(1)}$,$h^{3,(2)}$, and $h^{3,(3)}$ can now be used to determine the coefficients of f_1^3 and f_2^3 and thereby the entries of x^3. In Figure 7 it is illustrated how the queries are defined in this example.

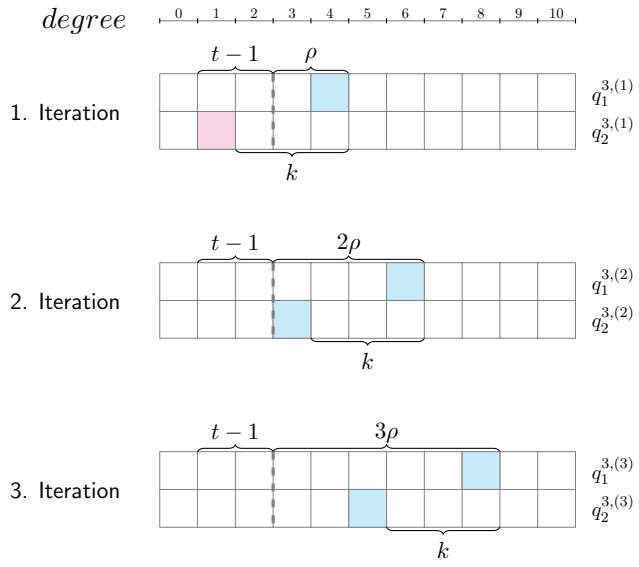

Figure 7: Illustration of how the queries $q_a^{l,(\sigma)}$ are defined in each iteration of the example. A cyan square means that the corresponding monomial is added to $d_a^{l,(\sigma)}$ yielding $q_a^{l,(\sigma)}$, and a magenta square means that nothing is added.

In Figure 8 it is seen how the query polynomials are defined in a scheme with $b = 3$ rows and $s = 2$ iterations.

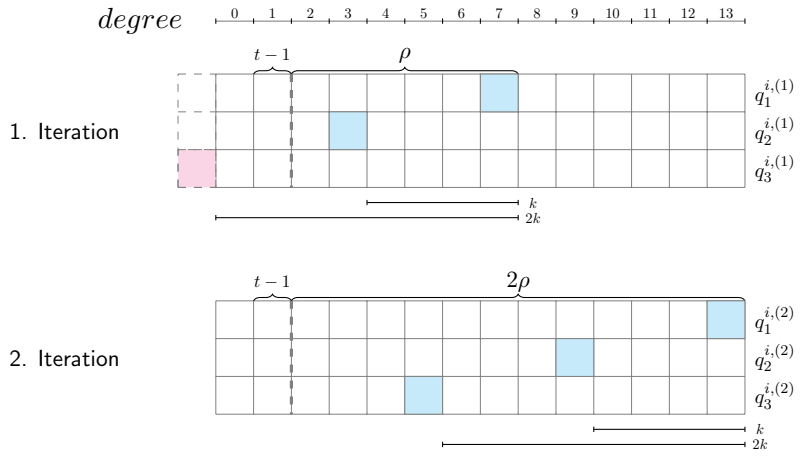

Figure 8: Illustration of how the queries $q_a^{l,(\sigma)}$ are defined in each iteration of a scheme with $k = 4, t = 2, \rho = 6, n = 14, b = 3, \beta = 1, r = 1$ and $s = 2$ fetching file i. A cyan square means that the corresponding monomial is added to $d_a^{l,(\sigma)}$ yielding $q_a^{l,(\sigma)}$, and a magenta square means that nothing is added.

5.1 Towards a Subfield Subcode Scheme

Now, we will make steps towards a subfield subcode version of the scheme $\mathscr{S}_{TGKFH}$. In particular, we wish to consider how the scheme should be altered such that we can choose the retrieval code as the subfield subcode $\mathrm{RS}_t(\boldsymbol{\alpha})|_{\mathbb{F}_q}$. In this scheme it is essential that the true response is a codeword of a length n RS code such that errors and erasures can be corrected, but in the subfield subcode case we cannot merely add the appropriate monomial as in (5.4) since this polynomial will likely not evaluate to the subfield $\mathbb{F}_q \subseteq \mathbb{F}_{q^m}$ over the evaluation points $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m})^n$. We will primarily apply the ideas of [8]. We consider the relation between the spaces $\mathbb{F}_{q^m}[X]^k$ and $\mathrm{RS}_k(\boldsymbol{\alpha})$ through the evaluation homomorphism $\mathrm{ev}_{\boldsymbol{\alpha}}$, but with slightly more structure.

Proposition 5.7. *Let $N = q^m - 1$, and let $\boldsymbol{\alpha}$ be all of $\mathbb{F}_{q^m}\backslash\{0\}$ in any order. Then we have an isomorphism of rings given by*

$$\mathrm{ev}_{\boldsymbol{\alpha}} : \mathbb{F}_{q^m}[X]/\langle X^N - 1\rangle \to \left((\mathbb{F}_{q^m})^N, \star\right).$$

Proof. We have the factorisation

$$X^{q^m-1} - 1 = \prod_{\alpha \in \mathbb{F}_{q^m}^*} (X - \alpha).$$

Hence, by the Chinese remainder theorem we get the relation

$$\begin{aligned}\mathbb{F}_{q^m}[X]/\langle X^{q^m-1} - 1\rangle &\cong \mathbb{F}_{q^m}[X]/\langle X - \alpha_1\rangle \times \cdots \times \mathbb{F}_{q^m}[X]/\langle X - \alpha_{q^m-1}\rangle \\ &\cong \mathbb{F}_{q^m} \times \cdots \times \mathbb{F}_{q^m} \\ &\cong (\mathbb{F}_{q^m})^{q^m-1}\end{aligned}$$

by the isomorphism of rings

$$\begin{aligned} f(X) \pmod{X^{q^m-1}-1} \mapsto (f(X) \pmod{X-\alpha_1}, \ldots, f(X) \pmod{X-\alpha_{q^m-1}}) \\ = (f(\alpha_1), \ldots, f(\alpha_{q^m-1})) = \mathrm{ev}_{\boldsymbol{\alpha}}(f), \end{aligned}$$

since for each ring $\mathbb{F}_{q^m}[X]/\langle X-\alpha_i\rangle$ we have the identity $X = \alpha_i$. ■

For the rest of this section, $\boldsymbol{\alpha}$ will be as in Proposition 5.7.

Inspired by the trace mapping Tr we define the *trace polynomial*.

Definition 5.8. Let $f \in \mathbb{F}_{q^m}[X]/\langle X^N-1\rangle$. Then we define the trace polynomial of f as

$$\tau(f) = f + f^q + \cdots + f^{q^{m-1}}.$$

It turns out that these trace polynomials are exactly the polynomials we seek which will be apparent soon. But first, a couple of lemmata.

Lemma 5.9. *The trace mapping* $\mathrm{Tr} : \mathbb{F}_{q^m} \to \mathbb{F}_q$ *is surjective.*

Proof. Let $\gamma \in \mathbb{F}_q$, and let $a \in \mathbb{F}_{q^m}$ be an element not in $\mathrm{Ker}(\mathrm{Tr})$. Then we have

$$\mathrm{Tr}(\gamma \mathrm{Tr}(a)^{-1} a) = \gamma \mathrm{Tr}(a)^{-1} \mathrm{Tr}(a) = \gamma,$$

hence, $\gamma \in \mathrm{Im}(\mathrm{Tr})$. ■

Lemma 5.10. *For* $f \in \mathbb{F}_{q^m}[X]/\langle X^N-1\rangle$ *it holds that*

$$\mathrm{Tr}(\mathrm{ev}_{\boldsymbol{\alpha}}(f)) = \mathrm{ev}_{\boldsymbol{\alpha}}(\tau(f)).$$

Proof. Let $f \in \mathbb{F}_{q^m}[X]/\langle X^N-1\rangle$. Then for $l = 0, \ldots, m-1$ and $i = 1, \ldots, N$ we get

$$\mathrm{ev}_{\boldsymbol{\alpha}}(f)_i^{q^l} = \left(\sum_{j=0}^{N-1} f_j \alpha_i^j\right)^{q^l} = \sum_{j=0}^{N-1} f_j^{q^l} \alpha_i^{jq^l} = \mathrm{ev}_{\boldsymbol{\alpha}}(f^{q^l})_i. \tag{5.16}$$

Thus,

$$\begin{aligned} \mathrm{ev}_{\boldsymbol{\alpha}}(\tau(f)) &= \mathrm{ev}_{\boldsymbol{\alpha}}(f + \cdots + f^{q^{m-1}}) \\ &= \mathrm{ev}_{\boldsymbol{\alpha}}(f) + \cdots + \mathrm{ev}_{\boldsymbol{\alpha}}(f^{q^{m-1}}) \\ &= \mathrm{Tr}(\mathrm{ev}_{\boldsymbol{\alpha}}(f)), \end{aligned}$$

where the last equality follows from (5.16) and the definition of the trace. ■

Proposition 5.11. *For all polynomials* $f \in \mathbb{F}_{q^m}[X]/\langle X^N-1\rangle$ *the following statements are equivalent:*

i) $f = \mathrm{Tr}(g)$ *for some* $g \in \mathbb{F}_{q^m}[X]/\langle X^N-1\rangle$,

ii) $f^q = f$, *and*

iii) for all $\beta \in \mathbb{F}_{q^m}$ *it holds that* $f(\beta) \in \mathbb{F}_q$.

Proof. For $f \in \mathbb{F}_{q^m}[X]/\langle X^N - 1\rangle$ suppose that $i)$ is satisfied, i.e., that $f = \mathrm{Tr}(g)$ for $g \in \mathbb{F}_{q^m}[X]/\langle X^N - 1\rangle$. Then

$$f^q = \mathrm{Tr}(g)^q = g^q + g^{q^2} + \cdots + g^{q^m} = g^q + g^{q^2} + \cdots + g = \mathrm{Tr}(g) = f.$$

Now, suppose that $f^q = f$. Then for any $\beta \in \mathbb{F}_{q^m}$ we have $f(\beta)^q = f(\beta)$ which implies that $f(\beta) \in \mathbb{F}_q$.

If we have $f(\beta_i) \in \mathbb{F}_q$ for all $\beta_i \in \mathbb{F}_{q^m}$ then in particular, $f(\alpha_i) \in \mathbb{F}_q$ for the entries α_i of $\boldsymbol{\alpha}$. Thus, by Lemma 5.9 there are elements $\gamma_i \in \mathbb{F}_{q^m}$ such that $f(\alpha_i) = \mathrm{Tr}(\gamma_i)$. By Lagrange interpolation of the points (α_i, γ_i) we get a polynomial $g(\alpha_i) = \gamma_i$. Then by the evaluation isomorphism

$$\mathrm{ev}_{\boldsymbol{\alpha}}(\tau(g)) = \mathrm{Tr}(\mathrm{ev}_{\boldsymbol{\alpha}}(g)) = \mathrm{ev}_{\boldsymbol{\alpha}}(f)$$

by Lemma 5.10. Hence, $f = \tau(g)$. ■

In the case that $\boldsymbol{\alpha}$ is all of $\mathbb{F}_{q^m}$ we will still have this restriction on polynomials that evaluates to the subfield over $\boldsymbol{\alpha}$. This yields us a corollary.

Corollary 5.12. *Let* $\boldsymbol{\alpha}$ *be either all of* $\mathbb{F}_{q^m}\backslash\{0\}$ *or all of* $\mathbb{F}_{q^m}$. *Then the code*

$$\mathrm{RS}_k(\boldsymbol{\alpha})|_{\mathbb{F}_q}$$

is the repetition code for $k \leq q^{m-1}$.

Proof. Let $f \in \mathbb{F}_{q^m}[X]/\langle X^N - 1\rangle$ with $\deg f < k$ that satisfies $f(\alpha_i) \in \mathbb{F}_q$ for all entries α_i of $\boldsymbol{\alpha}$. By Proposition 5.11 we have $f = \tau(h)$ for some $h \in \mathbb{F}_{q^m}[X]/\langle X^N - 1\rangle$, thus, h and f must be constant polynomials by the degree of f. By this observation, $\mathrm{RS}_k(\boldsymbol{\alpha}) = \mathrm{span}\{\mathrm{ev}_{\boldsymbol{\alpha}}(\tau(1))\} = \mathrm{span}\{\mathrm{ev}_{\boldsymbol{\alpha}}(1)\}$. ■

This corollary does also tell us something important about when our collusion protection of the subfield subcode scheme will drop to 1. This is the case every time the collusion protection of the extension field scheme satisfies $t \leq q^{m-1}$. The observations made in this subsection are far from exhaustive and some considerations are still to be made:

By an addition of a trace polynomial instead of a monomial in (5.4) leaves us downloading potentially too many (or too few) symbols in an iteration since the degree of such a polynomial must have q dividing its degree. This needs to be taken into account when choosing parameters for the scheme.

The considerations made here are only for the (almost) full support case - that is, $\boldsymbol{\alpha}$ is chosen as either all elements of $\mathbb{F}_{q^m}$ or $\mathbb{F}_{q^m} \setminus \{0\}$. Potentially, polynomials of smaller degree evaluates to the prime field if we choose the evaluation points $\boldsymbol{\alpha}$ as a smaller subset of $\mathbb{F}_{q^m}$ in a clever manner.

It should also be noted that the file of interest potentially requires some disentangling from the response polynomial since the response polynomial will likely not be as neat as in (5.9). This should however not be an issue.

At last, by generalising this scheme to be over GRS codes we can choose $D|_{\mathbb{F}_q} = \mathrm{GRS}_t(\boldsymbol{\alpha}, \boldsymbol{v})|_{\mathbb{F}_q}$ for $\boldsymbol{v} \neq \mathbf{1}$. This allows for a clever choice of $\boldsymbol{v}$ that perhaps yields a subfield subcode of dimension larger than 1 for $t \leq q^{m-1}$.

Conclusion

This thesis has presented the ideas of coded PIR as well as two promising PIR schemes $\mathscr{S}_{FGHK}$ and $\mathscr{S}_{TGKFH}$ that yields competitive PIR rates both in the case that we have no server collusion and in the case that we have repetition coding. Small examples of these schemes in action are presented to enlighten the ideas of the scheme. The second of the two schemes offers protection against servers in case that they reply maliciously or not at all. In a real world setting the possibilities of servers acting in such a manner is very much there, hence, such a generalisation is most welcome.

A subfield subcode version of the scheme $\mathscr{S}_{FGHK}$ is presented which yields a reduced computational complexity at the servers. A potential reduction in collusion protection follows this scheme, but hopefully certain retrieval codes D exists such that the minimum distance $d_{D|_{\mathbb{F}_q}^{\perp}}$ only gets reduced very slightly from $d_{D^{\perp}}$, and the rate of the scheme will not decrease when we pass to the subfield subcode scheme. Further analysis of the trace schemes where the retrieval code is chosen as a trace code instead is also possible since such a scheme is only just presented in this thesis.

Preliminary considerations are made regarding a subfield subcode scheme of the scheme $\mathscr{S}_{TGKFH}$, in particular, if we have full support or full support without 0 we must add trace polynomials of the proper degree in the query construction since these are the only polynomials that evaluates to the prime field over every element of the extension field. If such an approach is deemed infeasible then there are two other possibilities: Choose the evaluation points cleverly ensuing that a desirable polynomial evaluates to the prime field over these points or generalising the scheme to GRS retrieval codes $\mathrm{GRS}(\boldsymbol{\alpha}, \boldsymbol{v})$ and then choosing the scaling vector $\boldsymbol{v}$ in an appropriate manner.

References

[1] K. Banawan and S. Ulukus. "The Capacity of Private Information Retrieval from Coded Databases". In: *IEEE Transactions on Information Theory* 64.3 (2018), pp. 1945–1956.

[2] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. "Wild McEliece". In: *SAC 2010* 7 (2011), pp. 143–158.

[3] B. Chor et al. "Private Information Retrieval". In: *Journal of the ACM* 45.6 (1998), pp. 965–982.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. 2nd ed. John Wiley & Sons, 2006.

[5] P. Delsarte. "On Subfield Subcodes of Modified Reed-Solomon Codes". In: *IEEE Transactions on Information Theory* 21.5 (1975), pp. 575–576.

[6] R. Freij-Hollanti et al. "Private Information Retrieval from Coded Databases with Colluding Servers". In: *SIAM Journal of Applied Algebra and Geometry* 1 (2017), pp. 647–664.

[7] J. von sur Gathen and J. Gerhard. *Modern Computer Algebra*. 3rd. ed. Cambridge University Press, 2013.

[8] F. Hernando, K. Marshall, and M. E. O'Sullivan. "The Dimension of Subcode-Subfields of Shortened Generalized Reed-Solomon Codes". In: *Designs, Codes and Cryptography* 69 (2013), pp. 131–142.

[9] E. Kushilevitz and R. Ostrovsky. "Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval". In: *Proceedings 38th Annual Symposium on Foundations of Computer Science, IEEE* (1997), pp. 364–373.

[10] C. J. Lex. *Goppa Codes*. URL: `https://projekter.aau.dk/projekter/files/402522011/Goppa_Codes_Finished.pdf`.

[11] J. H. van Lint. *Introduction to Coding Theory*. 3rd ed. Springer Verlag, 1999.

[12] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. 5th printing. CRC Press, 2001.

[13] D. Mirandola and G. Zemor. "Critical Pairs for the Product Singleton Bound". In: *IEEE Transactions on Information Theory* 61.9 (2015), pp. 4928–4937.

[14] H. Randriambololona. "An Upper Bound of Singleton Type for Componentwise Products of Linear Codes". In: *IEEE Transactions on Information Theory* 59.12 (2013), pp. 7936–7939.

[15] H. Stichtenoth. *Algebraic Function Fields and Codes*. 2nd ed. Springer Verlag, 2009.

[16] H. Sun and S. A. Jafar. "The Capacity of Private Information Retrieval". In: *IEEE Transactions on Information Theory* 63.7 (2017), pp. 4075–4088.

[17] H. Sun and S. A. Jafar. "The Capacity of Robust Private Information Retrieval With Colluding Databases". In: *IEEE Transactions on Information Theory* 64.4 (2018), pp. 2361–2370.

[18] R. Tajeddine, O. Gnilke, and S. El Rouayheb. "Private Information Retrieval From MDS Coded Data in Distributed Storage Systems". In: *IEEE Transactions on Information Theory* 64.11 (2018), pp. 7081–7093.

[19] R. Tajeddine et al. *Private Information Retrieval from Coded Storage Systems with Colluding, Byzantine, and Unresponsive Servers*. 2018. URL: https://arxiv.org/abs/1806.08006.

Appendix

Low-Complexity PIR Using Subfield Subcodes

Christian J. Lex
Department of Mathematical Sciences,
Aalborg University
Aalborg, Denmark
clex16@student.aau.dk

Oliver W. Gnilke
Department of Mathematical Sciences,
Aalborg University
Aalborg, Denmark
owg@math.aau.dk

***Abstract*—A major drawback of many PIR schemes is the high computational cost at the servers. We present a scheme that uses only operations in the prime field during response generation. For binary extension fields this leads to schemes that only need XOR operations at the servers to calculate the responses. This is achieved by restricting the queries to a subfield subcode or trace code. We investigate possible parameter ranges and focus on the example of GRS codes and subfield subcodes of these.**

***Index Terms*—Private Information Retrieval, Subfield Subcodes, Trace Codes, Alternant Codes.**

I. Introduction

A lot of work on PIR considers only two parameters of interest, the download cost, or rate, and storage overhead. The effort at the server side, which can be significant is usually ignored. Two exceptions are [1], where a PIR scheme that requires no computation on the server side is presented. And [2], which introduces a new parameter called access complexity, that measures how many files a server has to access to calculate its response.

In this paper we will present the use of subfield subcodes to a private information retrieval (PIR) scheme. In particular, we consider an alteration of the PIR scheme presented in [3] which uses generalised Reed-Solomon (GRS) codes to achieve a PIR rate of $(n-(k+t-1))/n$ for n servers using a dimension k storage code. This scheme offers protection against $0 < t \leq n-k$ colluding servers. By choosing the retrieval code as a subfield subcode of a GRS code (such subfield subcodes are at times called alternant codes) instead of a GRS code a significant computational cost is saved during file retrieval. We compare the complexity of this subfield subcode scheme to the scheme in [3] using both GRS codes over any field of characteristic 2 as well as GRS codes over $\mathbb{F}_{q^m}$ where $q > 2$.

II. Private Information Retrieval

We use the setup in [3]. Let n denote the number of servers in the DSS, let μ denote the number of files, and let b denote the number of rows in each file. The files $x^1, \ldots, x^\mu \in (\mathbb{F}_{q^m})^{b\times k}$ are encoded row-wise using a linear $[n, k, d_C]_{q^m}$ code C, the *storage code*, as

$$Y = \begin{bmatrix} x^1 \\ \vdots \\ x^\mu \end{bmatrix} G_C, \tag{1}$$

where G_C is a generator matrix for C. Superscripts will refer to files, subscripts to servers, and parenthesis to vector entries. E.g., $y^i = x^i G_C$ denotes the i^{th} encoded file, $\boldsymbol{y}_j^i$ the part of this encoded file on the j^{th} server, and $y_j^i(a)$ the a^{th} entry of this vector.

Now, the PIR scheme is described. We will denote this scheme by $\mathscr{S}$. Let D be a linear length n code over $\mathbb{F}_{q^m}$. We call this code the *retrieval code*. The star product of the storage code and the retrieval code is defined as

$$C \star D = \operatorname{span}\{(c_0 d_0, \ldots, c_{n-1} d_{n-1}) \mid \boldsymbol{c} \in C, \boldsymbol{d} \in D\} \tag{2}$$

Let $c := d_{C\star D} - 1$. The set $J := \{1, \ldots, \max\{k, c\}\}$ will be the index set of servers (after perhaps a rearrangement of servers) from which symbols are downloaded. To make sure that exactly one file is downloaded after s iterations of the algorithm we enforce that $b = \operatorname{lcm}(c,k)/k$ and $s = \operatorname{lcm}(c,k)/c$.

A query is constructed by choosing μb codewords $\boldsymbol{d}^{l,a}$ of D uniformly at random where $l \in [\mu]$ and $a \in [b]$. For each $j \in [n]$ we define a vector $\boldsymbol{d}_j$ by

$$\boldsymbol{d}_j^l = \left(d^{l,1}(j), \ldots, d^{l,b}(j)\right), \ \boldsymbol{d}_j = \left(\boldsymbol{d}_j^1, \ldots, \boldsymbol{d}_j^\mu\right) \in (\mathbb{F}_{q^m})^{\mu b}. \tag{3}$$

A subset $J_1 = [d_{C\star D} - 1] \subseteq J$ is defined and is partitioned according to the rows of the files:

$$J_1^1 = [c/b], J_1^2 = [2c/b] \setminus [c/b], \ldots, J_1^b = [c] \setminus [(b-1)c/b]. \tag{4}$$

Suppose that we wish to fetch the i^{th} file. Then the j^{th} query $\boldsymbol{q}_j^i$ is defined by

$$\boldsymbol{q}_j^i = \begin{cases} \boldsymbol{d}_j + \boldsymbol{e}_{b(i-1)+a} & \text{if} j \in J_1^a, \\ \boldsymbol{d}_j & \text{if} j \notin J_1. \end{cases} \tag{5}$$

The j^{th} entry of the response vector is then determined as

$$r_j^i = \langle \boldsymbol{q}_j^i, \boldsymbol{y}_j \rangle. \tag{6}$$

This is iterated by applying a length c/b cyclic shift to the partition (4) within J. That is, if the a^{th} index set for the $u-1^{\text{th}}$ iteration is $J_{u-1}^a = \{j_1, \ldots, j_{c/b}\}$ then

$$J_u^a = \left\{j_1 + c/b \pmod{|J|}, \ldots, j_{c/b} + c/b \pmod{|J|}\right\}. \tag{7}$$

Then the queries are defined as in (5) now using the partition J_u^a and J_u as the union of the J_u^a. The data can now be reconstructed as follows: Choose a parity check matrix H for

$C \star D$. Suppose that $\boldsymbol{r}^i$ is the response vector of the first iteration. Then

$$H\boldsymbol{r}^i = H\boldsymbol{c} + H \begin{bmatrix} y_1^i(1) \\ \vdots \\ y_c^i(b) \\ 0_{(n-c)\times 1}, \end{bmatrix} \tag{8}$$

where $c \in C \star D$. Hence, $y_1^i(1), \ldots, y_c^i(b)$ can be recovered from (8) since any choice of c columns of H is a linearly independent set. This is done similarly for the other $s-1$ iterations until an entire file can be reconstructed.

We present here two theorems of [3]:

Theorem II.1. *Let C be an $[n, k, d_C]$-code and let D be some length n retrieval code. If the minimum distance of $C \star D$ denoted by $d_{C\star D}$ satisfies $d_{C\star D} \leq k$ or there exists $J \subseteq [n]$ such that every size k subset of J is an information set of C then $\mathscr{S}$ retrieves the correct file with rate $(d_{C\star D} - 1)/n$.*

Theorem II.2. *$\mathscr{S}$ protects against $d_{D^\perp} - 1$ colluding servers.*

III. Low-Complexity Variant

We will now modify the scheme $\mathscr{S}$ to reduce the complexity of calculating the responses r_j^i.

The main idea is to replace the retrieval code D with its subfield subcode $D|_{\mathbb{F}_q}$. We have the immediate inclusion $C \star D \supseteq C \star D|_{\mathbb{F}_q}$ hence we can retrieve at least as many symbols as before. The protection against collusion relies on the min. distance of a trace code.

Theorem III.1 (Delsarte [4])**.** *For a code D over $\mathbb{F}_{q^m}$,*

$$(D|_{\mathbb{F}_q})^\perp = \mathrm{Tr}(D^\perp).$$

In general min. distances of trace codes are not easy to determine and even good lower bounds are not known. But they are easy enough to determine for the examples of interest in the present application.

We collect these observations in the following theorem.

Theorem III.2. *Let C and D be a storage and a retrieval code respectively for the scheme $\mathscr{S}$. Then we can define a low complexity variant $\mathscr{S}|_{\mathbb{F}_q}$ by replacing D with $D|_{\mathbb{F}_q}$. The scheme $\mathscr{S}|_{\mathbb{F}_q}$ has at least the same rate as the scheme $\mathscr{S}$ and protects against t'-collusion, where $t' = d_{\mathrm{Tr}(D^\perp)} - 1$.*

We will keep our focus on the variant of the scheme $\mathscr{S}$ using GRS codes as storage and retrieval codes as well as its corresponding subfield subcode scheme $\mathscr{S}|_{\mathbb{F}_q}$. Hence, by $\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v})$ we denote the GRS code

$$\begin{aligned} &\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}) = \\ &\{(v_0 f(\alpha_0), \ldots, v_{n-1} f(\alpha_{n-1})) \mid f \in \mathbb{F}_{q^m}[X], \deg f < k\}, \end{aligned} \tag{9}$$

for $\boldsymbol{v} \in (\mathbb{F}_{q^m}^*)^n$ and *support* $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m})^n$ satisfying $\alpha_i \neq \alpha_j$ for $i \neq j$. The GRS codes are closed under taking duals. In particular,

$$(\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{v}))^\perp = \mathrm{GRS}_{n-k}(\boldsymbol{\alpha}, \tilde{\boldsymbol{v}}), \tag{10}$$

where

$$\tilde{\boldsymbol{v}} = \left(\frac{1}{v_0 \prod_{i\neq 0}(\alpha_0 - \alpha_i)}, \ldots, \frac{1}{v_{n-1} \prod_{i\neq n-1}(\alpha_{n-1} - \alpha_i)} \right). \tag{11}$$

The star product as defined in (2) for GRS codes C and D satisfies

$$\begin{aligned} &\mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u}) \star \mathrm{GRS}_l(\boldsymbol{\alpha}, \boldsymbol{v}) = \\ &\qquad \mathrm{GRS}_{\min\{n, k+l-1\}}(\boldsymbol{\alpha}, (u_0 v_0, \ldots, u_{n-1} v_{n-1})). \end{aligned} \tag{12}$$

Assume now that we have $D = \mathrm{GRS}_t(\boldsymbol{\alpha}, \boldsymbol{v})$ for some $\boldsymbol{\alpha} \in (\mathbb{F}_{q^m})^n$ and $\boldsymbol{v} \in (\mathbb{F}_{q^m}^*)^n$. Theorem III.1, (10), and (11) combines to yield the diagram in Fig. 1.

$$\begin{array}{ccc} D = \mathrm{GRS}_t(\boldsymbol{\alpha}, \boldsymbol{v}) & \xleftrightarrow{\text{Dual}} & \mathrm{GRS}_{n-t}(\boldsymbol{\alpha}, \tilde{\boldsymbol{v}}) = D^\perp \\ \Big\downarrow \cap \mathbb{F}_q^n & & \mathrm{Tr} \Big\downarrow \\ D|_{\mathbb{F}_q} = \mathrm{GRS}_t(\boldsymbol{\alpha}, \boldsymbol{v}) & \xleftrightarrow{\text{Dual}} & \mathrm{Tr}(\mathrm{GRS}_{n-t}(\boldsymbol{\alpha}, \tilde{\boldsymbol{v}})) = \mathrm{Tr}(D^\perp) \end{array}$$

Fig. 1. Relation between duals of GRS codes and their subfield subcodes.

Taking the storage code as $C = \mathrm{GRS}_k(\boldsymbol{\alpha}, \boldsymbol{u})$ and the retrieval code as $D|_{\mathbb{F}_q}$ yields a PIR rate of at least

$$\frac{n-k-t+1}{n} \tag{13}$$

and a protection against $d_{\mathrm{Tr}(D^\perp)} - 1$ colluding servers assuming that $t \leq n - k$.

In case a lower bound on the collusion protection is required we can assign $\mathrm{Tr}(D) = \mathrm{Tr}(\mathrm{GRS}_t(\boldsymbol{\alpha}, \boldsymbol{v}))$ as the retrieval code. This yields a collusion protection $t' = d_{D^\perp|_{\mathbb{F}_q}} - 1 \geq t$, however in general we have

$$C \star \mathrm{Tr}(D) \not\subseteq C \star D. \tag{14}$$

Hence, this approach will yield no immediate lower bound on the rate of the new scheme. We collect these observations in the following corollary

Corollary III.3. *Let C and D be a storage and a retrieval code respectively for the scheme $\mathscr{S}$. Then we can define a low complexity variant $\mathrm{Tr}(\mathscr{S})$ by replacing D with $\mathrm{Tr}(D)$. The scheme $\mathrm{Tr}(\mathscr{S})$ has rate $(d_{C\star \mathrm{Tr}(D)} - 1)/n$ and protects against t'-collusion, where $t' = d_{D^\perp|_{\mathbb{F}_q}} - 1$.*

IV. Complexity

This scheme has the evident disadvantage to the GRS scheme in [3] that we have no immediate lower bound on the collusion protection. However, a considerable reduction in time complexity can be achieved by choosing the retrieval code as a subfield subcode. We consider the case $q^m = 2^m$. In the scheme $\mathscr{S}$ the response $\boldsymbol{r}^i$ has n entries, hence, in each iteration n inner products (6) are calculated. Each of these consists of $\mu b - 1$ additions and μb multiplications in $\mathbb{F}_{2^m}$. An addition in $\mathbb{F}_{2^m}$ has time complexity $\mathcal{O}(m)$ and a

multiplication in $\mathbb{F}_{2^m}$ has time complexity $\mathcal{O}(m^2)$ using naïve multiplication. Therefore, determining $\boldsymbol{r}^i$ has complexity

$$n(\mu b - 1)\mathcal{O}(m) + n\mu b\mathcal{O}(m^2) = n\mu b\mathcal{O}(m^2), \tag{15}$$

which is iterated s times.

Let us turn to the case where $D|_{\mathbb{F}_2}$ is the retrieval code. Then $\boldsymbol{q}_j^i \in (\mathbb{F}_2)^{\mu b}$, thus, we have only additions in (6) and determining $\boldsymbol{r}^i$ has complexity

$$n(\mu b - 1)\mathcal{O}(m). \tag{16}$$

We again consider the scheme $\mathscr{S}$ now for $q > 2$. Determining $\boldsymbol{r}^i$ consists of $n(\mu b - 1)$ additions in $\mathbb{F}_{q^m}$ each of complexity $\mathcal{O}(m\log_2(q))$ and $n\mu b$ multiplications each of complexity $\mathcal{O}(m^2(\log_2(q))^2)$. Hence, each iteration of the response computation has complexity

$$n\mu b\mathcal{O}(m^2(\log_2(q))^2). \tag{17}$$

In the subfield subcode case the inner product (6) yields $\mu b - 1$ additions in $\mathbb{F}_{q^m}$ each with complexity $\mathcal{O}(m\log_2(q))$ and μb multiplications of an element of $\mathbb{F}_{q^m}$ and an element of $\mathbb{F}_q$ each with complexity $\mathcal{O}(m\log_2(q)^2)$. Hence, we get time complexity for computing $\boldsymbol{r}^i$ as

$$n\mu b\mathcal{O}(m(\log_2(q)^2). \tag{18}$$

Faster multiplication algorithms than naïve multiplication exist for large integers (see for example [5]) however the naïve case suffices for our analysis since the fields of consideration are relatively small.

V. Examples

As a proof of concept as well as for clarification we will look at a few small examples.

Example V.1. For the storage code C_1 we choose the $[4, 1, 4]$ repetition code $C_1 = \mathrm{GRS}_1(\mathbf{1}, \boldsymbol{\alpha}_1)$ for $\boldsymbol{\alpha}_1$ as all of $\mathbb{F}_4$ in any order. The retrieval code is chosen as the Reed-Solomon code $D_1 = \mathrm{GRS}_3(\mathbf{1}, \boldsymbol{\alpha}_1)$ which has the $[4, 1, 4]$ repetition code as its dual. Hence, $D_1|_{\mathbb{F}_2}$ has parity check matrix $H = [1\ 1\ 1\ 1]$, and $D_1|_{\mathbb{F}_2}$ has minimum distance 2 and dimension 3 just as D_1. This yields a scheme $\mathscr{S}_1|_{\mathbb{F}_2}$ with rate $R_1 = 1/4$ and collusion protection 3. Thus, in this case, we can pass to the subfield subcode scheme for free.

As a slightly more interesting example where proper storage coding can be done is as follows.

Example V.2. Choose for $k \leq 3$ the storage code as the length 8 code $C_2 = \mathrm{GRS}_k(\boldsymbol{v}_2, \boldsymbol{\alpha}_2)$ where $\boldsymbol{v}_2$ is any element of $(\mathbb{F}_8^*)^8$ and $\boldsymbol{\alpha}_2$ is all of $\mathbb{F}_8$. The retrieval code is $D_2 = \mathrm{GRS}_5(\mathbf{1}, \boldsymbol{\alpha}_2)$, and $D_2|_{\mathbb{F}_2}$ has generator matrix

$$G_2|_{\mathbb{F}_2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}. \tag{19}$$

We see that $D_2|_{\mathbb{F}_2}$ is the self-dual $[8, 4, 4]$ extended binary Hamming code. Hence, the scheme $\mathscr{S}_2|_{\mathbb{F}_2}$ protects against $t' = 3$ collusion, since $\left(D_2|_{\mathbb{F}_2}\right)^\perp = D_2|_{\mathbb{F}_2}$, compared to $t = 5$ for the scheme $\mathscr{S}$. Computations show that the rate for both schemes is given by $R_2 = (4 - k)/8$.

At last, let us consider a trinary example.

Example V.3. Thus, we take the storage code $C_3 = \mathrm{GRS}_k(\boldsymbol{v}_3, \boldsymbol{\alpha}_3)$ for some $\boldsymbol{v}_3 \in (\mathbb{F}_9^*)^9$, $\boldsymbol{\alpha}_3 \in (\mathbb{F}_9)^9$, and $k \leq 5$. The retrieval code is chosen as $D_3 = \mathrm{GRS}_4(\mathbf{1}, \boldsymbol{\alpha}_3)$. We then get $D_3|_{\mathbb{F}_3}$ as a $[9, 3, 6]$-code with generator matrix

$$G_3|_{\mathbb{F}_3} = \begin{bmatrix} 1 & 0 & 0 & 2 & 1 & 2 & 2 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 & 1 & 2 & 2 & 1 \\ 0 & 0 & 1 & 2 & 1 & 1 & 0 & 2 & 2 \end{bmatrix}.$$

Its dual is a $[9, 6, 3]$ code and hence we have a collusion protection of $t' = 2$, down from $t = 4$ for the scheme $\mathscr{S}$. Calculating the star product codes shows that both schemes have identical rates $R_3 = (6 - k)/9$.

VI. Conclusion & Future Work

We have presented an alteration of the PIR scheme of [3] using subfield subcodes as the retrieval code of the PIR scheme. This scheme achieves a considerable improvement in computational complexity of the server side response calculations while maintaining the same rate. As the exampled show this gain is usually accompanied by a reduction in the collusion resistance.

Future work will explore further examples. For the examples presented it holds that $C \star D = C \star D|_{\mathbb{F}_q}$, hence the rates are equal. Trivial examples for which this equality does not hold are given by codes D of dimension > 1 for which $D|_{\mathbb{F}_q}$ is the repetition code. These lead to schemes $\mathscr{S}|_{\mathbb{F}_q}$ with an increased rate, but at the cost of a complete loss of collusion protection. We will explore the possibility of an example for which the rate increases without the collusion protection completely disappearing. Furthermore, a non-trivial example of a scheme using corollary III.3 would be of interest.

Apart from more examples, the possibility of a class of GRS codes for which the subfield subcodes and their parameters are predictable should be considered. In [6] the authors present some classes of GRS codes with high dimensional subfield subcodes and algorithms to search them.

References

[1] Lavauzelle, Julien. *Private information retrieval from transversal designs* IEEE Transactions on Information Theory 65.2 (2018): 1189-1205

[2] Y. Zhang, E. Yaakobi, T. Etzion, M. Schwartz, *On the Access Complexity of PIR Schemes*, 2019 IEEE International Symposium on Information Theory (ISIT), pp.2134-2138

[3] R. Freij-Hollanti, O. W. Gnilke, C. Hollanti, and D. A. Karpuk. *Private Information Retrieval From Coded Databases With Colluding Servers*, In: SIAM Journal of Applied Algebra and Geometry, vol. 1, pp. 647-664, 2017.

[4] P. Delsarte. *On Subfield Subcodes of Modified Reed-Solomon Codes*, In: IEEE Transactions on Information Theory, vol. IT-21, pp. 575-576, 1975.

[5] J. von sur Gathen and J. Gerhard. *Modern Computer Algebra*, 3rd edition, Cambridge University Press, 2014.

[6] F. Hernando, K. Marshall, and M. E. O'Sullivan. *The Dimension of Subcode-Subfields of Shortened Generalized Reed-Solomon Codes*, In: Des. Codes Cryptogr. (2013) 69:131–142.