Recovery Sets of Subspaces From a Simplex Code

Yeow Meng Chee, *Senior Member, IEEE*, Tuvi Etzion, *Life Fellow, IEEE*, Han Mao Kiah, *Senior Member, IEEE*, and Hui Zhang, *Member, IEEE*

***Abstract*— Recovery sets for vectors and subspaces are important in the construction of distributed storage system codes. These concepts are also interesting in their own right. In this paper, we consider the following very basic recovery question: what is the maximum number of possible pairwise disjoint recovery sets for each recovered element? The recovered elements in this work are d-dimensional subspaces of a k-dimensional vector space over $\mathbb{F}_q$. Each server stores one representative for each distinct one-dimensional subspace of the k-dimensional vector space, or equivalently a distinct point of $\mathrm{PG}(k-1,q)$. As column vectors, the associated vectors of the stored one-dimensional subspaces form the generator matrix of the $[(q^k-1)/(q-1),k,q^{k-1}]$ simplex code over $\mathbb{F}_q$. Lower bounds and upper bounds on the maximum number of such recovery sets are provided. It is shown that generally, these bounds are either tight or very close to being tight.**

***Index Terms*— Availability, distributed storage, recovery sets, subspaces.**

I. Introduction and Preliminaries

An important problem in distributed storage systems is to design a code that can recover any d-dimensional subspace U (d-subspace in short) of a k-space over $\mathbb{F}_q$ from n disjoint subsets of servers, where each server stores one distinct 1-subspace of the k-space. Such a subset of servers is called a ***recovery set*** if the subspace which is spanned by their 1-subspaces (or equivalently by the vectors of their 1-subspaces) contains U. The set of vectors in this recovery set of servers will be also called a recovery set. Similarly, the set of 1-subspaces of this recovery set of servers will be also called a recovery set. Given d and k, one wishes to know what is the minimum number of servers required for a given multiple recovery of each d-subspace of the k-space over $\mathbb{F}_q$, using linear combinations of pairwise disjoint sets of servers. This problem is of interest for distributed storage system codes and it is called availability [24]. A related problem is to find the availability of distributed storage codes whose codewords are subspaces of possibly higher dimension [28]. It is also related to a problem associated with a new model in private information retrieval codes defined for minimizing storage, e.g. [1], [3], [6], [14], [15], [21], [22], [30], [32], and [33]. In general, coding for subspaces has become quite fashionable, mainly for wide applications for distributed storage and also for random network coding, e.g. [4], [12], [13], [19], [20], and [23].

In this paper, we will solve another related problem for distributed storage systems. Let d and k be integers such that $1 \leqslant d \leqslant k$, let U be any d-subspace of $\mathbb{F}_q^k$, and assume that each server stores a 1-subspace of $\mathbb{F}_q^k$. What is the maximum number of subsets of pairwise disjoint servers such that each such subset can recover U, i.e., their 1-subspaces span U? This is equivalent to the recovery of d-subspaces from the columns of the generator matrix of the simplex code. Recovering a batch of elements from the columns of a generator matrix for the $[(q^k-1)/(q-1),k,q^{k-1}]$ simplex code was considered in the past [17], [31], [32], [33], but only for binary alphabet and not for subspaces. This work is a generalization in this direction.

Let $\mathbb{F}_q^k$ be the vector space of dimension k over $\mathbb{F}_q$, the finite field with q elements, $q \in \mathbb{P}$, where $\mathbb{P}$ is the set of prime powers. Clearly, $\mathbb{F}_q^k$ consists of q^k vectors of length k and it contains $\frac{q^k-1}{q-1}$ distinct 1-subspaces. In general $\mathbb{F}_q^k$ contains $\begin{bmatrix} k \\ \ell \end{bmatrix}$ distinct ℓ-subspaces, where

$$\begin{bmatrix} k \\ \ell \end{bmatrix}_q \triangleq \frac{(q^k-1)(q^{k-1}-1)\cdots(q^{k-\ell+1}-1)}{(q^\ell-1)(q^{\ell-1}-1)\cdots(q-1)}$$

is the well-known *q-binomial coefficient*, known also as the *Gaussian coefficient*. To distinguish between the set of q^k vectors of $\mathbb{F}_q^k$ and its 1-subspaces we will denote the set of all 1-subspaces of $\mathbb{F}_q^k$ by V_q^k. When $q=2$, V_q^k corresponds to the nonzero elements of $\mathbb{F}_q^k$ and hence we will not distinguish between them. We note that each 1-subspace of V_q^k can be represented by $q-1$ elements of $\mathbb{F}_q^k$. Two such representations can be obtained from each other using multiplication by an element of $\mathbb{F}_q$. Such representation for a 1-subspace of V_q^k can be viewed as a projective point in the projective geometry $\mathrm{PG}(k-1,q)$. Note, that a d-subspace in $\mathbb{F}_q^k$ is a $(d-1)$-subspace in $\mathrm{PG}(k-1,q)$. Using this representation the generator matrix of the $[(q^k-1)/(q-1),k,q^{k-1}]$ simplex code contains the projective points of the projective geometry $\mathrm{PG}(k-1,q)$. Finally, we will use the isomorphism between the

Manuscript received 26 October 2023; revised 18 March 2024; accepted 20 May 2024. Date of publication 30 May 2024; date of current version 17 September 2024. An earlier version of this paper was presented in part at the 2020 IEEE International Symposium on Information Theory [5]. *(Corresponding author: Tuvi Etzion.)*

Yeow Meng Chee is with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576 (e-mail: ymchee@nus.edu.sg).

Tuvi Etzion is with the Faculty of Computer Science, Technion—Israel Institute of Technology, Haifa 3200003, Israel (e-mail: etzion@cs.technion.ac.il).

Han Mao Kiah is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371 (e-mail: hmkiah@ntu.edu.sg).

Hui Zhang was with the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576. She is now with AiTreat Pte. Ltd., Singapore 068914 (e-mail: hzhang.sg@gmail.com).

Communicated by A.-L. Horlemann-Trautmann, Associate Editor for Coding and Decoding.

Digital Object Identifier 10.1109/TIT.2024.3407197

finite field $\mathbb{F}_{q^n}$ in a way that sometimes we will consider the information in the servers as elements in the vector space $\mathbb{F}_q^n$ and sometimes the information in the servers will be presented by elements from the finite field $\mathbb{F}_{q^n}$. But, there will never be a mixed notation in the representation.

In a distributed storage system, each server stores an ℓ-subspace of $\mathbb{F}_q^k$ and a typical user is interested to obtain a d-subspace U of $\mathbb{F}_q^k$. The information of this d-subspace is usually stored in sub-packets across several servers. It is quite common that some servers, which hold important sub-packets of U, are not available. Hence, it is required that it will be possible to recover U from other sets of servers, which implies that for some n, each d-subspace will be recovered from n pairwise disjoint sets of servers.

Assume now, that each server stores a different 1-subspace of $\mathbb{F}_q^k$, i.e., an element of V_q^k. Let $N_q(k,d)$ be the maximum number of such pairwise disjoint sets of servers that will be able to recover any given d-subspace U. W.l.o.g. (without loss of generality), in some places, we will assume that U is $\mathbb{F}_q^d$ (in $\mathbb{F}_q^k$) since $\mathbb{F}_q^k$ can be written as $W+U$ for any d-subspace $U \in \mathbb{F}_q^k$ and some $(k-d)$-subspace W of $\mathbb{F}_q^k$. The $(k-d)$-subspace W can be taken w.l.o.g. and abuse of notation as $\mathbb{F}_q^{k-d}$. In this case, an element of $\mathbb{F}_q^k$ can be written as (x,y), where $x \in W$ and $y \in U$. It can be also written as $x+y$. Before we start to discuss the general results, let us consider a simple scenario that could be very useful in the sequel.

Theorem 1: The set V_q^d contains $n = \left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor$ disjoint recovery sets for the d-subspace V_q^d.

Proof: Let α be a primitive element of $\mathbb{F}_{q^d}$. Any d consecutive powers of α are linearly independent and any $\frac{q^d-1}{q-1}$ consecutive powers of α are contained in a distinct set of 1-subspaces. Hence, if $S_i = \{\alpha^{(i-1)d}, \alpha^{(i-1)d+1}, \ldots, \alpha^{id-1}\}$, we have that $\langle S_i \rangle = \mathbb{F}_q^d$ for any $1 \leqslant i \leqslant n$. ∎

Since any two d-subspaces are isomorphic we have the following consequence.

Corollary 2: If U is the d-subspace to be recovered from U, then the 1-subspaces of U can be partitioned into $\left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor$ recovery sets for U and possibly one more subset with less than d 1-subspaces.

Corollary 3: For each $q \geqslant 2$ and $k \geqslant 2$ we have

$$N_q(k,k) = \left\lfloor \frac{q^k-1}{k(q-1)} \right\rfloor .$$

We start with the most simple upper and lower bounds on $N_q(k,d)$. First, the upper bound will be derived.

Theorem 4: If $q \in \mathbb{P}$ and $k \geqslant d$ is a positive integer, then

$$N_q(k,d) \leqslant \left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor + \left\lfloor \frac{\ell(q-1)+q^k-q^d}{(d+1)(q-1)} \right\rfloor ,$$

where ℓ is the reminder from the division of $\frac{q^d-1}{q-1}$ by d.

Proof: Let U be the d-subspace which has to be recovered. Recovery sets of size d can be obtained only from d linearly independent 1-subspaces of U. By Theorem 1, there are at most $\left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor$ such recovery sets. The number of remaining nonzero elements from U is $\ell(q-1)$ and the number of nonzero elements in $\mathbb{F}_q^k$ which are not contained in U is q^k-q^d. All the remaining recovery sets are from these elements and they must be of size at least $d+1$ which implies the claim of the theorem. ∎

The bound of Theorem 4 will be improved when we are more precise in the number of recovery sets of size $d+1$ and we will have some approximation on the number of recovery sets of larger size.

We continue to derive a related simple lower bound. For this bound, we need the following lemma and its consequence.

Lemma 5: If $f(z) = z^d + a_{d-1}z^{d-1} + \cdots + a_1 z + a_0$ is a primitive polynomial over $\mathbb{F}_q$, then

$$\sum_{i=0}^{d-1} a_i + 1 \neq 0.$$

Proof: If $\sum_{i=0}^{d-1} a_i + 1 = 0$, then $f(1) = 1 + \sum_{i=0}^{d-1} a_i = 0$ and hence 1 is a root of $f(z)$, a contradiction. ∎

For $x_1, x_2, \ldots, x_\ell \in \mathbb{F}_{q^n}$ and $\gamma \in \mathbb{F}_q$ let

$$\gamma(x_1, x_2, \ldots, x_\ell) = (\gamma x_1, \gamma x_2, \ldots, \gamma x_\ell).$$

Corollary 6: If α is a root of the primitive polynomial $f(z) = z^d + a_{d-1}z^{d-1} + \cdots + a_1 z + a_0$ over $\mathbb{F}_q$, then the d vectors in

$$R \triangleq \{(x, \alpha^i), (x, \alpha^{i+1}), \ldots, (x, \alpha^{i+d})\}$$

form a recovery set for $\mathbb{F}_q^d$ in $\mathbb{F}_q^k$, where $x \in \mathbb{F}_{q^{k-d}} \setminus \{0\}$. (note that α is a primitive element in $\mathbb{F}_{q^d}$.). In other words, R is a recovery set for $\mathbb{F}_q^d$.

Proof: Since α is a root of $f(z)$, it follows that

$$\alpha^d + a_{d-1}\alpha^{d-1} + \cdots + a_1\alpha + a_0 = 0.$$

and hence for each $i \geqslant 0$ we have

$$\alpha^{i+d} + a_{d-1}\alpha^{i+d-1} + \cdots + a_1\alpha^{i+1} + a_0\alpha^i = 0.$$

On the other hand, since by Lemma 5, $\sum_{j=0}^{d-1} a_j + 1 \neq 0$, it follows that

$$(x, \alpha^{i+d}) + a_{d-1}(x, \alpha^{i+d-1}) + \cdots + a_1(x, \alpha^{i+1}) + a_0(x, \alpha^i)$$

$$= (x, \alpha^{i+d}) + \sum_{j=0}^{d-1} a_j (x, \alpha^{i+j})$$

$$= \left(\left(\sum_{i=0}^{d-1} a_i + 1 \right) x, \mathbf{0} \right) = (\gamma x, \mathbf{0}),$$

for some $\gamma \in \mathbb{F}_q \setminus \{0\}$. Thus, $(x, \mathbf{0})$ can be recovered from the vectors in R and therefore

$$(\mathbf{0}, \alpha^i), (\mathbf{0}, \alpha^{i+1}), \ldots, (\mathbf{0}, \alpha^{i+d-1}), (\mathbf{0}, \alpha^{i+d}) \tag{1}$$

can be recovered too. Since each d consecutive elements in Eq. (1) are linearly independent, it follows that

$$\{(x, \alpha^i), (x, \alpha^{i+1}), \ldots, (x, \alpha^{i+d})\}$$

is a recovery set for $\mathbb{F}_q^d$ in $\mathbb{F}_q^k$. ∎

The last part of the proof of Corollary 6 can be used to prove the following lemma.

Lemma 7: If $x \in \mathbb{F}_{q^{k-d}} \setminus \{0\}$ and α is a primitive element in $\mathbb{F}_{q^d}$, then

$$R \triangleq \{(x, \mathbf{0}), (x, \alpha^i), (x, \alpha^{i+1}), \ldots, (x, \alpha^{i+d-1})\}$$

is a recovery set for $\mathbb{F}_q^d$ in $\mathbb{F}_q^k$, where $x \in \mathbb{F}_{q^{k-d}} \setminus \{0\}$.

Theorem 8: If $q \in \mathbb{P}$ and $k \geqslant d$ is a positive integer, then

$$N_q(k,d) \geqslant \left\lfloor \frac{q^d - 1}{d(q-1)} \right\rfloor + \left\lfloor \frac{q^d}{d+1} \right\rfloor \frac{q^{k-d} - 1}{q-1}. \tag{2}$$

Furthermore, if $d+1$ divides q^d, then there is equality in Eq. (2).

Proof: The lower bound is obtained similarly to the upper bound of Theorem 4 by observing that recovery sets of size d can be obtained only from the elements of U. From the remaining space, we construct as many as possible recovery sets of size $d+1$. Let $\mathcal{L}$ be the set of nonzero vectors from $\mathbb{F}_q^{k-d}$ whose first nonzero entry is a *one*. Clearly,

$$|\mathcal{L}| = q^{k-d-1} + q^{k-d-2} + \cdots + q + 1 = \frac{q^{k-d} - 1}{q-1},$$

and the set $\{(x,y) \ : \ x \in \mathcal{L}, \ y \in \mathbb{F}_q^d\} \cup \hat{U}$, where $\hat{U}$ is the set of one subspaces of U, is a set of representatives for V_q^k. By Corollary 6 and Lemma 7, for each $x \in \mathcal{L}$ we can form $\left\lfloor \frac{q^d}{d+1} \right\rfloor$ recovery sets for U.

From U we obtain $\left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor$ recovery sets. There are $\frac{q^{k-d}-1}{q-1}$ elements in $\mathcal{L}$ and each one yields $\left\lfloor \frac{q^d}{d+1} \right\rfloor$ recovery sets. Thus, the claim of the theorem follows. ∎

The value of $N_q(k,1)$ was analyzed in [5], where the following results were obtained.

Theorem 9: Let $q \in \mathbb{P}$ and $n \geqslant 2$ be positive integers.

(1) If q is even, then $N_q(k,1) = 1 + \frac{q^k - q}{2(q-1)}$.

(2) If q is odd, then $N_q(k,1) = 1 + \frac{q^{k-1}-1}{2} + \left\lfloor \frac{q^{k-1}-1}{3(q-1)} \right\rfloor$.

In this paper, an analysis for $d > 1$ will be done and the rest of this paper is organized as follows. In Section II we describe some of the ideas for our main construction for a lower bound on the maximum number of recovery sets. We demonstrate these ideas first only for $q = 2$, where we don't distinguish between the vectors of $\mathbb{F}_2^n$ and its 1-subspaces. The ideas for a lower bound will be demonstrated now for $d = 2^m - 1$ and subspaces over $\mathbb{F}_2$ using a binary perfect code of length $2^m - 1$. Also, some ideas for an upper bound will be demonstrated for $d = 2$ using integer programming. Tight bounds or almost tight bounds for $q = 2$ and specific small values of d, will be presented in Sections III and IV, where lower bounds will be considered in the first and upper bounds will be considered in the second. The bounds on the number of recovery sets for $q > 2$ are presented in Section V. Some of the bounds that we obtain are tight, e.g., when $k - d$ is even and $d + 2$ divides $q + 1$. In general, optimality can be obtained in other cases by constructing certain partitions. Furthermore, the upper bound for $q > 2$ will be improved compared to $q = 2$. Conclusion and problems for further research are presented in Section VI.

II. The Main Ideas for a Lower Bound and an Upper Bound

We start this section by describing first our main ideas for constructing many recovery sets from the columns of the parity check matrix of the simplex code over $\mathbb{F}_q$. This will yield a lower bound on the maximum number of possible recovery sets. We will continue by describing the main ideas to obtain an upper bound on this number. Then in subsection II-A we will demonstrate the main ideas to obtain a lower bound by using binary perfect codes. In subsection II-B we will rephrase the main ideas of the upper bound in terms of integer programming to obtain a specific upper bound. Most of the discussion in this section is for $q = 2$, but the main ideas will be similar for $q > 2$.

We represent the points of $\mathrm{PG}(k-1,2)$ which are the 1-subspaces of V_2^k by a $2^{k-d} \times 2^d$ matrix T. The rows of the matrix T will be indexed and identified by the 2^{k-d} elements of $\mathbb{F}_2^{k-d}$ and the columns of T by the elements of $\mathbb{F}_2^d$, where the first one will be $\mathbf{0}$, and the other columns will be identified, and ordered, as $\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{2^d-2}$ and also by their appropriate vector representation of length d, where α is a primitive element in $\mathbb{F}_{2^d}$. Entry $T(x,y)$, $x \in \mathbb{F}_2^{k-d}$, $y \in \mathbb{F}_2^d$, in the matrix will contain the pair (x,y), which represents an element in $\mathbb{F}_2^k$. It will be equivalent if the entry $T(x,y)$ will contain the element $x + y$, where $x \in W$, $y \in U$, W is a $(k-d)$-subspace and U is a d-subspace such that $\mathbb{F}_2^k = W + U$. The elements of the matrix T represent together all the 1-subspaces (elements) of $\mathbb{F}_2^k$, i.e., the points of $\mathrm{PG}(k-1,2)$. Note, that $T(0,0)$ is the only entry of T which does not contain a 1-subspace.

Lemma 10: The entries of the matrix T, excluding the entry $T(0,0) = 0$, represent exactly all the points of $\mathrm{PG}(k-1,2)$ (which are the 1-subspaces of $\mathbb{F}_2^k$).

Using the representation of $\mathbb{F}_2^k$ with the matrix T we can now design recovery sets for $\mathbb{F}_2^d$ as follows.

1) Any d consecutive powers of α in the first row of T are linearly independent and hence can be used to generate $\lfloor \frac{2^d-1}{d} \rfloor$ pairwise disjoint recovery sets (see Theorem 1).
2) Any $d+1$ consecutive elements in a row x of T, $x \neq \mathbf{0}$, span $\mathbb{F}_2^d$ (see Corollary 6). Similarly, any d consecutive elements in a row x of T together with $T(x,\mathbf{0})$ span $\mathbb{F}_2^d$ (see Lemma 7). Hence, any such row of T can be used to generate $\left\lfloor \frac{2^d}{d+1} \right\rfloor$ pairwise disjoint recovery sets. The same is true if there exists another construction for partition (or almost a partition) of the elements in a row of T into sets for which each one spans $\mathbb{F}_2^d$.
3) The remaining elements in each row are combined and recovery sets, usually of size $d + 2 + \epsilon$, for small ϵ, are generated.

We will try to improve the bound of Theorem 8 in a simple way. After the construction of $\lfloor \frac{q^d}{d+1} \rfloor$ recovery sets from each row of the matrix T, not all the elements in T will be used in these recovery sets. The remaining elements, which will be called ***leftovers***, can be used to obtain more recovery sets

whose size is $d+2$ or more (as we will see in the sequel). If there were many leftovers in the first row of T, then some recovery sets of size $d+1$ can be obtained from these leftovers and combined with the leftovers from the other rows of T. For the other leftovers, it will be proved that in some cases recovery sets of size $d+2$ are sufficient. If $d+1$ divides q^d, there are no leftovers in each row and hence we cannot obtain more recovery sets. Note, that in such case the leftovers from the first row of T cannot be of any help to provide more recovery sets. Hence, the bound of Theorem 8 in Eq. (2) is attained.

The representation of $\mathrm{PG}(k-1,2)$ with the matrix T can be used also to obtain an upper bound on the maximum number of recovery sets. Recovery sets of size d can be constructed only from the elements in the first row of T and we cannot have more than $\left\lfloor \frac{2^d-1}{d} \right\rfloor$ such recovery sets. It is readily verified that involving elements from the first row of T with elements from the other rows of T will require more than d elements in the recovery sets. This will imply that the leftovers from the first row of T can be used with elements from the other rows of T to form a recovery set of size at least $d+1$. Other leftovers in other rows can be used in recovery sets of size at least $d+2$. This can imply bounds that are better than the one in Theorem 4. The upper bounds obtained by these arguments can be found either by careful analysis or by using integer programming.

In the two subsections which follow we will give examples for the implementation of these ideas. In Sections III, IV, and V we will implement these ideas for various parameters and also for $q>2$.

A. Recovery Sets via Perfect Codes

To demonstrate our main ideas for lower bounds in the following sections, we start with a special case which on one hand explains the main idea, and on the other hand it has its beauty as it is based on binary perfect codes with radius one, and finally, it is optimal. For a word $x \in \mathbb{F}_2^n$, the ***ball*** of radius one around x, $B(x)$, is the set of words at Hamming distance at most one from x, i.e.

$$B(x) \triangleq \{y \ : \ y \in \mathbb{F}_2^n, \ d_H(x,y) \leqslant 1\},$$

where, $d_H(x,y)$ is the Hamming distance between x and y. A ***binary perfect code*** $\mathcal{C}$ of length n is a set of binary words of length n, such that the balls of radius one around the codewords of $\mathcal{C}$ form a partition of $\mathbb{F}_2^n$. It is well known [9] that a binary perfect code exists for each $n=2^m-1$ and there are many nonequivalent such codes from which only one is a linear code, ***the Hamming code***. Assume now that the dimension d for the subspace to be recovered is of the form $d=2^m-1$. Let U be the d-subspace to be recovered and W be a $(k-d)$-subspace of $\mathbb{F}_2^k$ such that $W+U=\mathbb{F}_2^k$. The vectors stored in the servers are represented by the $2^{k-d} \times 2^d$ array T. By Corollary 2, the first row of T is partitioned into $\left\lfloor \frac{2^d-1}{d} \right\rfloor$ recovery sets. Now, let $\mathcal{C}$ be any binary perfect code of length $d=2^m-1$ (linear or nonlinear). $\mathbb{F}_2^d$ is partitioned into $\frac{2^d}{d+1}$ balls of radius one whose centers are the codewords of $\mathcal{C}$. Consider row x of T, where x is nonzero. This row is partitioned into $\frac{2^d}{d+1}$ subsets, where the i-th subset is $P_i \triangleq \{x+y \ : \ y \in B_i\}$ and B_i is the i-th ball from the partition of $\mathbb{F}_2^d$ into balls, of radius one, by the perfect codes $\mathcal{C}$. Clearly, $B_i=\{c_i\} \cup \{c_i+e_j \ : \ 1 \leqslant j \leqslant d\}$, where c_i is the i-th codeword of $\mathcal{C}$ and e_j is a unit vector with a *one* in the j-th coordinate. It implies that $P_i=\{x+c_i\} \cup \{x+c_i+e_j \ : \ 1 \leqslant j \leqslant d\}$ and hence $e_j \in \langle P_i \rangle$ for each $1 \leqslant j \leqslant d$. Therefore, $U \subset \langle P_i \rangle$ and P_i is a recovery set for U. Since a recovery set of size d can be obtained only from elements of U, it follows that except for the $\left\lfloor \frac{2^d-1}{d} \right\rfloor$ recovery sets obtained from the first row of T all the other recovery sets must be of size at least $d+1$. Thus, we have a special case of Theorem 8.

Theorem 11: If $d=2^m-1$ then

$$N_2(k,d)=\left\lfloor \frac{2^d-1}{d} \right\rfloor + \frac{2^k-2^d}{d+1}$$

It should be noted that the same result can be obtained by using the construction based on Theorem 1 and Corollary 6. We introduced the construction based on perfect codes for its uniqueness.

B. Integer Programming Bound

We continue and demonstrate an upper bound on $N_2(k,2)$ using integer programming. As before, consider the $2^{k-2} \times 4$ array T and seven types of recovery sets from this array, whose entries, except for $T(0,0)$, are exactly the elements stored in the 2^k-1 servers. Now, the variables which denote the number of recovery sets of each type are defined as follows.

1) The first type has recovery sets with two elements from the first row of T. The number of recovery sets from this type will be denoted by X_1.
2) The variable X_2 denotes the number of recovery sets with one element from the first row and two elements from an internal row (not the first one) of T. (note that there is no recovery set when one element is taken from the first row of T and two elements are taken from two different internal rows of T.)
3) The variable X_3 denotes the number of recovery sets with one element from the first row and three elements from three internal rows of T.
4) The variable Y_3 denotes the number of recovery sets with three elements from one of the internal rows of T.
5) The variable Y_{22} will denote the number of recovery sets with two elements from one internal row of T and two elements from another internal row of T.
6) The variable Y_4 denotes the number of recovery sets with four elements: two elements from one internal row of T and two elements from two other different internal rows of T.
7) The variable Y_5 denotes the number of recovery sets with five elements from five different internal rows of T.

Note, that four elements from four different internal rows cannot be used to form a recovery set. It can be verified that each other possible recovery set contains by its definition one

of these seven types and hence can be replaced to save some servers for other recovery sets.

We continue with a set of three inequalities related to these variables. The first row of T contains three elements and hence we have that

$$2X_1 + X_2 + X_3 \leqslant 3 \ .$$

The second inequality is also very simple, we just sum the contribution of each type to the total number of elements in the internal rows of T which have a total of $2^k - 4$ elements. This implies that

$$2X_2 + 3X_3 + 3Y_3 + 4Y_{22} + 4Y_4 + 5Y_5 \leqslant 2^k - 4 \ .$$

For the last equation, we consider any specific internal row of T and the types that are using at least two elements from this row. The number of related recovery sets using this row will be denoted by $X_2', Y_3', Y_{22}', Y_4'$. We have the inequality

$$2X_2' + 3Y_3' + 2Y_{22}' + 2Y_4' \leqslant 4 \ .$$

But, since $Y_3' \in \{0,1\}$ while the other variables can get the values 0,1, or 2, it follows that

$$2X_2' + 4Y_3' + 2Y_{22}' + 2Y_4' \leqslant 4 \ .$$

Next, we sum this equation over all the $2^{k-2} - 1$ internal rows, taking into account that Y_{22}' in the equation is counted for two distinct rows. Hence, we have

$$2X_2 + 4Y_3 + 4Y_{22} + 2Y_4 \leqslant 2^k - 4 \ .$$

We can write the following linear programming problem

$$\gamma = \text{maximize} \quad X_1 + X_2 + X_3 + Y_3 + Y_{22} + Y_4 + Y_5$$

subject to nonnegative integer variables and the following three constraints

$$\begin{aligned} &2X_1 + X_2 + X_3 \leqslant 3 \\ &2X_2 + 3X_3 + 3Y_3 + 4Y_{22} + 4Y_4 + 5Y_5 \leqslant 2^k - 4 \\ &2X_2 + 4Y_3 + 4Y_{22} + 2Y_4 \leqslant 2^k - 4 \ . \end{aligned}$$

We apply now the dual linear programming problem [7]

$$\gamma = \text{minimize} \quad 3Z_1 + (2^k - 4)Z_2 + (2^k - 4)Z_3$$

subject to nonnegative variables and the following seven constraints

$$\begin{aligned} 2Z_1 &\geqslant 1 \\ Z_1 + 2Z_2 + 2Z_3 &\geqslant 1 \\ Z_1 + 3Z_2 &\geqslant 1 \\ 3Z_2 + 4Z_3 &\geqslant 1 \\ 4Z_2 + 4Z_3 &\geqslant 1 \\ 4Z_2 + 2Z_3 &\geqslant 1 \\ 5Z_2 &\geqslant 1 \end{aligned}$$

Using IBM software CPLEX it was found that this optimization problem has exactly one solution, $Z_1 = \frac{1}{2}$, $Z_2 = \frac{1}{5}$, and $Z_3 = \frac{1}{10}$, and hence $\gamma = \frac{3}{2} + \frac{3(2^{k-1}-2)}{5}$. But, since our solution is an integer solution, it follows that the solution is $\gamma = \left\lfloor \frac{3}{2} + \frac{3(2^{k-1}-2)}{5} \right\rfloor$ and as a consequence.

Lemma 12: For $k \geqslant 2$, $N_2(k,2) \leqslant \left\lfloor \frac{3}{2} + \frac{3(2^{k-1}-2)}{5} \right\rfloor = \left\lfloor \frac{3 \cdot 2^k + 3}{10} \right\rfloor$.

This method of integer programming can be used for general parameters, i,e., also for $q > 2$, and also for $d > 2$. We will omit the evolved computations which are increased as q and d are increased.

III. Constructions of Recovery Sets for Binary Alphabet

In this section lower bounds on the number of recovery sets for binary alphabet, when the subspace to recover has a low dimension, will be derived.

A. Recovery Sets for Two-Dimensional Subspaces

The goal in this subsection is first to find the value of $N_2(k,2)$ which considers binary 2-subspaces. The proof consists of three steps. In the first step, we will take a recovery set of size two (it is not possible to have two such disjoint recovery sets) and generate as many possible pairwise disjoint recovery sets of size three which is the best possible. In the second step, we will continue and from the remaining 1-subspaces (the leftovers) we generate pairwise disjoint recovery sets of size five. This construction will provide a lower bound on the number of recovery sets. For the upper bound, in the third step, we will use either integer programming or careful analysis for the possible size of recovery sets to prove that our construction yields the largest possible number of pairwise disjoint recovery sets. The integer programming was demonstrated in Section II-B. We can also obtain similar results to the ones obtained with integer programming, by careful analysis. After that, we will finish to find the value of $N_2(k,2)$.

Consider the $2^{k-2} \times 4$ array T whose rows are indexed by the vectors of $\mathbb{F}_2^{k-2}$, where the first row is indexed by $\mathbf{0}$. The columns are indexed by $\mathbf{0}$, u, v, and $u+v$, where $U = \langle u, v \rangle$ is the 2-subspace to be recovered.

We have seen that any two nonzero elements from the first row span the subspace U and any three elements from any other row also span U. Together we have 2^{k-2} recovery sets and 2^{k-2} leftovers, one from each row, where these elements can be chosen arbitrarily. We will prove that these leftovers can be partitioned into recovery sets of size five and possibly one or two sets of a different small size. This will be based on a structure that we call a ***quintriple*** (as it consists of five elements and three ways to generate one of its elements). We start with the definition of a quintriple.

A subset $\{x_1, x_2, x_3, x_4, x_5\} \subset \mathbb{F}_2^{k-2}$ is called a ***quintriple*** if $x_1 = x_2 + x_3 = x_4 + x_5$. A quintriple yields one recovery set with one element from each row associated with the elements of the quintriple. As the leftovers in each row can be chosen arbitrarily we choose the following elements in the rows indexed by x_1, x_2, x_3, x_4, and x_5.

$$(x_1, \mathbf{0})$$
$$(x_2, u)$$
$$(x_3, u+v)$$
$$(x_4, v)$$
$$(x_5, u+v).$$

Since

$$(\mathbf{0}, v) = (x_1, \mathbf{0}) + (x_2, u) + (x_3, u+v)$$

and

$$(\mathbf{0}, u) = (x_1, \mathbf{0}) + (x_4, v) + (x_5, u+v),$$

it follows that U is recovered from $\{(x_1, \mathbf{0})(x_2, u), (x_3, u+v), (x_4, v), (x_5, u+v)\}$. Thus, our goal is to form as many as possible pairwise disjoint quintriples of $\mathbb{F}_2^{k-2}$, and possibly one or two more subsets of small size, to complete our construction.

Now, we will consider a partition of $\mathbb{F}_2^m$ into disjoint quintriples and possibly one or two more subsets of a small size. Each quintriple will yield one recovery set; one more recovery set will be obtained from the other elements (outside the quintriples). We distinguish between four cases, depending on whether m is congruent to 0, 1, 2, or 3 modulo 4. The construction will be recursive, where the initial conditions are quintriples for $m = 4$, $m = 5$, $m = 6$, and $m = 7$. For the recursive construction, rank-metric codes and lifted rank-metric codes will be introduced, and their definitions are provided now.

For two $\tau \times \eta$ matrices A and B over $\mathbb{F}_q$ the *rank distance*, $d_R(A, B)$, is defined by

$$d_R(A, B) \triangleq \text{rank}(A - B) \ .$$

A code $\mathcal{C}$ is a $[\tau \times \eta, \ell, \delta]$ rank-metric code if its codewords are $\tau \times \eta$ matrices over $\mathbb{F}_q$, which form a linear subspace of dimension ℓ of $\mathbb{F}_q^{\tau \times \eta}$, and each two distinct codewords A and B satisfy $d_R(A, B) \geqslant \delta$. Rank-metric codes were well studied [8], [16], [25]. A Singleton bound for rank-metric codes was proved in [8], [16], and [25]:

Theorem 13: If $\mathcal{C}$ is a $[\tau \times \eta, \ell, \delta]$ rank-metric code, then

$$\ell \leqslant \min\{\tau(\eta - \delta + 1), \eta(\tau - \delta + 1)\}, \tag{3}$$

and this bound is attained for all possible parameters.

Codes which attain the upper bound of Eq. (3) are called ***maximum rank distance*** codes (or ***MRD*** codes in short).

A $\tau \times \eta$ matrix A over $\mathbb{F}_q$ is ***lifted*** to a τ-subspace of $\mathbb{F}_q^{\tau+\eta}$ whose generator matrix is $[I_\tau \ A]$, where I_τ is the identity matrix of order τ. A $[\tau \times \eta, \ell, \delta]$ rank-metric code $\mathcal{C}$ is *lifted* to a code $\mathbb{C}$ of τ-subspaces in $\mathbb{F}_q^{\tau+\eta}$ by lifting all the codewords of $\mathcal{C}$, i.e., $\mathbb{C} \triangleq \{\langle [I_\tau \ A] \rangle_R \ : \ A \in \mathcal{C}\}$, where $\langle B \rangle_R$ is the linear span of the rows of B. This code will be denoted by $\mathbb{C}^{\text{MRD}}$. If $\delta = \tau$ then all the codewords (τ-subspaces) of $\mathbb{C}$ are pairwise disjoint (intersect in the null-space $\{\mathbf{0}\}$). These codes have found applications in random network coding, where their constructions can be found [10], [11], [12], [20], [29].

We start by describing the construction of quintriples from $\mathbb{F}_2^4$ (one can find such quintriples by an appropriate computer search). Let α be a root of the primitive polynomial $x^4 + x + 1$, i.e., $\alpha^4 = \alpha + 1$. The nonzero elements of the finite field $\mathbb{F}_{16}$ are given in the following table.

	α^3	α^2	α^1	α^0
α^0	0	0	0	1
α^1	0	0	1	0
α^2	0	1	0	0
α^3	1	0	0	0
α^4	0	0	1	1
α^5	0	1	1	0
α^6	1	1	0	0
α^7	1	0	1	1
α^8	0	1	0	1
α^9	1	0	1	0
α^{10}	0	1	1	1
α^{11}	1	1	1	0
α^{12}	1	1	1	1
α^{13}	1	1	0	1
α^{14}	1	0	0	1

It is easy to verify that if S is a quintriple, then also $\alpha^i S$ is a quintriple for each $i \geqslant 0$. The set of elements $S = \{\alpha^0, \alpha^1, \alpha^3, \alpha^4, \alpha^7\}$ is a quintriple, where $\alpha^4 = \alpha + 1$ and $\alpha^4 = \alpha^3 + \alpha^7$. The three sets S, $\alpha^5 S = \{\alpha^5, \alpha^6, \alpha^8, \alpha^9, \alpha^{12}\}$, and $\alpha^{10} S = \{\alpha^2, \alpha^{10}, \alpha^{11}, \alpha^{13}, \alpha^{14}\}$, form a partition of the nonzero elements of F_{2^4} and hence also of $\mathbb{F}_2^4$ into three quintriples. This partition will be called the ***basic partition***. Partitions have an important role in our exposition. A basic theorem on partitions will be given later (see Theorem 31), but before the partitions will be based on lifted rank-metric codes and will be described individually.

Now, let $m = 4r$, $r \geqslant 2$, and let $\mathcal{C}$ be a $[4 \times (4r-4), \ell, 4]$ MRD code. By Theorem 13, we have that $\ell \leqslant 4r - 4$ and there exists a $[4 \times (4r-4), 4r-4, 4]$ code $\mathcal{C}$. Let $\mathbb{C}^{\text{MRD}}$ be the lifted code of $\mathcal{C}$. Each codeword of $\mathbb{C}^{\text{MRD}}$ is a 4-subspace of $\mathbb{F}_2^{4r}$ and any two such 4-subspaces intersect in the null-space $\{\mathbf{0}\}$. Hence, each such codeword can be partitioned into three disjoint quintriples, isomorphic to the quintriples of the basic partition. The nonzero elements which are contained in these 2^{4r-4} 4-subspaces are spanned by the rows of matrices of the form $[I_4 \ A]$, where A is a $4 \times (4r-4)$ binary matrix. Hence, the only nonzero elements of $\mathbb{F}_2^{4r}$ which are not contained in these 2^{4r-4} 4-subspaces are exactly the $2^{2r-4} - 1$ vectors of the form $(0, 0, 0, 0, z_1, z_2, \ldots, z_{4r-4})$, where $z_i \in \{0, 1\}$, and at least one of the z_i's is not a *zero*. This set of vectors and the all-zeros vector of length $4r$, form a $(4r-4)$-space isomorphic to $\mathbb{F}_2^{4r-4}$. We continue recursively with the same procedure for $m' = 4(r-1)$. The process ends with the initial condition which is the basic partition of $\mathbb{F}_2^4$. The outcome will be a partition of $\mathbb{F}_2^{4r}$ into $\frac{2^{4r}-1}{5}$ pairwise disjoint quintriples. Each one of the 2^{k-2} rows of the matrix T is used to construct one recovery set and each quintriple obtained in this way yields another recovery set. Hence we have the following result (note that $N_2(2, 2) = 1$).

Lemma 14: If $k \geqslant 2$ and $k \equiv 2 \pmod 4$ then $N_2(k, 2) \geqslant 2^{k-2} + \frac{2^{k-2}-1}{5} = \frac{3 \cdot 2^{k-1}-1}{5}$.

Example 1: The following four 4×4 matrices form a basis for a $[4 \times 4, 4, 4]$ MRD code $\mathcal{C}$.

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \text{ and } \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

The code $\mathcal{C}$ contain sixteen 4×4 matrices. Each one of these sixteen matrices is lifted to a 4-subspace of $\mathbb{F}_2^8$ and a code $\mathbb{C}^{\mathrm{MRD}}$ with sixteen 4-subspaces of $\mathbb{F}_2^8$ is obtained. As an example, the first 4×4 matrix is lifted to a 4×8 matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

which is a generator matrix of a 4-subspace of $\mathbb{F}_2^8$ whose fifteen nonzero vector are ordered to have the isomorphism φ of this 4-subspace to the additive group of $\mathbb{F}_{16}$, where α is a root of the primitive polynomial $x^4 + x + 1$ and $\varphi(\alpha^i) = a_i$ for each $0 \leqslant i \leqslant 14$.

$$\begin{array}{rcccccccc} a_0 & = & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ a_1 & = & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ a_2 & = & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ a_3 & = & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ a_4 = a_0 + a_1 & = & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ a_5 = a_1 + a_2 & = & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ a_6 = a_2 + a_3 & = & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ a_7 = a_0 + a_1 + a_3 & = & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ a_8 = a_0 + a_2 & = & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ a_9 = a_1 + a_3 & = & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ a_{10} = a_0 + a_1 + a_2 & = & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ a_{11} = a_1 + a_2 + a_3 & = & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ a_{12} = a_0 + a_1 + a_2 + a_3 & = & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a_{13} = a_0 + a_2 + a_3 & = & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ a_{14} = a_0 + a_3 & = & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{array}$$

Now, we have a partition of all the nonzero vectors of the 4-subspace which is isomorphic to the basic partition as follows:

$$\{a_0, a_1, a_3, a_4, a_7\}, \{a_5, a_6, a_8, a_9, a_{12}\}, \{a_2, a_{10}, a_{11}, a_{13}, a_{14}\}$$

where $a_4 = a_0 + a_1 = a_3 + a_7$, $a_9 = a_5 + a_6 = a_8 + a_{12}$, and $a_{14} = a_{10} + a_{11} = a_2 + a_{13}$.

The same procedure is done on the sixteen 4-subspace of $\mathbb{C}^{\mathrm{MRD}}$ and to the 4-subspace whose generator matrix is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

These seventeen 4-subspaces of $\mathbb{F}_2^8$ are pairwise disjoint (intersect in the null-space $\{\mathbf{0}\}$ and hence the $51 = 17 \cdot 3$ quintriples which were constructed form a partition of $\mathbb{F}_2^8 \setminus \{0\}$. ■

We continue and describe the partition of $\mathbb{F}_2^5$. Let α be a root of the primitive polynomial $x^5 + x^2 + 1$. Each one of the five sets $S_1 = \{\alpha^5, \alpha^0, \alpha^2, \alpha^7, \alpha^{10}\}$, $S_2 = \alpha S_1$, $S_3 = \alpha^{12} S_1$, $S_4 = \alpha^{13} S_1$, and $S_5 = \{\alpha^{16}, \alpha^4, \alpha^{27}, \alpha^{29}, \alpha^{30}\}$ is a quintriple. These five sets and $\{\alpha^9, \alpha^{21}, \alpha^{24}, \alpha^{25}, \alpha^{26}, \alpha^{28}\}$ form a partition of the 31 nonzero elements of $\mathbb{F}_2^5$. Moreover, the four elements $\alpha^{21}, \alpha^{25}, \alpha^{26}, \alpha^{28}$ are linearly dependent and will be used later to form an additional recovery set. Now, let $m = 4r + 1$ and let $\mathcal{C}$ be a $[4 \times (4r - 3), \ell, 4]$ MRD code. By Theorem 13, we have that $\ell \leqslant 4r - 3$ and there exists a $[4 \times (4r - 3), 4r - 3, 4]$ code $\mathcal{C}$. Let $\mathbb{C}^{\mathrm{MRD}}$ be the lifted code of $\mathcal{C}$. Each codeword of $\mathbb{C}^{\mathrm{MRD}}$ is a 4-subspace of $\mathbb{F}_2^{4r+1}$ and any two such 4-subspaces intersect in the null-space $\{\mathbf{0}\}$. Hence, each such codeword can be partitioned into three disjoint quintriples, isomorphic to the quintriples of the basic partition. The only nonzero elements of $\mathbb{F}_2^{4r+1}$ which are not contained in these 2^{4r-3} 4-subspaces are exactly the $2^{4r-3} - 1$ vectors of the form $(0, 0, 0, 0, z_1, z_2, \ldots, z_{4r-3})$, where $z_i \in \{0, 1\}$, and at least one of the z_i's is not *zero*. This set of vectors and the all-zeros vector of length $4r + 1$, form a $(4r - 3)$-space isomorphic to $\mathbb{F}_2^{4r-3}$. We continue recursively with the same procedure for $m' = 4r - 3$. This recursive procedure ends when $m' = 5$, where we use the partition of $\mathbb{F}_2^5 \setminus \{0\}$ with 5 quintriples and six more elements from which four are linearly dependent. The outcome will be a partition of $\mathbb{F}_2^{4r+1} \setminus \{0\}$ into $\frac{2^{4r+1}-7}{5}$ quintriples, a subset with four linearly dependent elements and a subset of size two. The subset with four linearly dependent elements and the remaining element of the first row yields another recovery set as follows. Let $\{x_1, x_2, x_3, x_4\}$ be the set of linearly dependent elements, i.e., $x_1 + x_2 + x_3 + x_4 = 0$ and assume w.l.o.g. that in the first row $(\mathbf{0}, v)$ was not used in any recovery set. As the remaining elements in each row can be chosen arbitrarily we choose (x_1, u), $(x_2, \mathbf{0})$, $(x_3, \mathbf{0})$, and $(x_4, \mathbf{0})$. Since $(\mathbf{0}, u) = (x_1, u) + (x_2, \mathbf{0}) + (x_3, \mathbf{0}) + (x_4, \mathbf{0})$ and $(\mathbf{0}, v)$ from the first row was not used in a recovery set, it follows that U is recovered from $\{(\mathbf{0}, v), (x_1, u), (x_2, \mathbf{0}), (x_3, \mathbf{0}), (x_4, \mathbf{0})\}$. Thus, the construction leads to the following bound (note that $N_2(3, 2) = 2$).

Lemma 15: If $k \geqslant 3$ and $k \equiv 3 \pmod 4$ then $N_2(k, 2) \geqslant 2^{k-2} + \frac{2^{k-2}-7}{5} + 1 = \frac{3 \cdot 2^{k-1}-2}{5}$.

We continue with an example of the partition for $\mathbb{F}_2^6$.

Example 2: Let α be a root of the primitive polynomial $x^6 + x + 1$, i.e., $\alpha^6 = \alpha + 1$. The table of $\mathbb{F}_{63}$ yields the following equalities

$$\alpha^0 = \alpha^1 + \alpha^6 = \alpha^{13} + \alpha^{35},$$
$$\alpha^9 = \alpha^7 + \alpha^{19} = \alpha^{12} + \alpha^{41}.$$

These two equalities yield the following four quintriples

$$\begin{aligned} S_1 &= \{\alpha^0, \alpha^1, \alpha^6, \alpha^{13}, \alpha^{35}\}, \\ S_2 = \alpha^2 S_1 &= \{\alpha^2, \alpha^3, \alpha^8, \alpha^{15}, \alpha^{37}\}, \\ S_3 = \alpha^4 S_1 &= \{\alpha^4, \alpha^5, \alpha^{10}, \alpha^{17}, \alpha^{39}\}, \\ S_4 &= \{\alpha^7, \alpha^9, \alpha^{12}, \alpha^{19}, \alpha^{41}\}. \end{aligned}$$

It is east to verify that for each j, $0 \leqslant j \leqslant 20$, $j \neq 11$, there is exactly one i, such that $i \equiv j \pmod{21}$ and

$\alpha^i \in S_1 \cup S_2 \cup S_3 \cup S_4$; for $j = 11$, there is no $i \equiv j \pmod{21}$ such that $\alpha^i \in S_1 \cup S_2 \cup S_3 \cup S_4$. This implies that the twelve sets

$$S_1, S_2, S_3, S_4, \alpha^{21}S_1, \alpha^{21}S_2, \alpha^{21}S_3, \alpha^{21}S_4,$$
$$\alpha^{42}S_1, \alpha^{42}S_2, \alpha^{42}S_3, \alpha^{42}S_4,$$

and the 2-subspace $\{\mathbf{0}, \alpha^{11}, \alpha^{32}, \alpha^{53}\}$ form a partition of $\mathbb{F}_2^6$ into 12 quintriples and one 2-subspace. ■

The leftover in each row of the subspace $\{\mathbf{0}, \alpha^{11}, \alpha^{32}, \alpha^{53}\}$ of Example 2, and the leftover element from the first row, $(\mathbf{0}, v)$, can be used to form a recovery set which consists of the elements $\{(\mathbf{0}, v), (\alpha^{11}, \mathbf{0}), (\alpha^{32}, \mathbf{0}), (\alpha^{53}, u)\}$. Hence, starting with $m = 4r + 2$ the partition $\mathbb{F}_2^{4r+2} \setminus \{0\}$ yields the following bound (note that $N_2(4, 2) = 5$).

Lemma 16: If $k \geqslant 4$ and $k \equiv 0 \pmod 4$ then $N_2(k, 2) \geqslant 2^{k-2} + \frac{2^{k-2}-4}{5} + 1 = \frac{3 \cdot 2^{k-1}+1}{5}$.

The construction for $m = 4r + 3$ is similar, but the construction for the initial condition of $m = 7$ is more complicated. We manage to obtain a partition of $\mathbb{F}_2^{4r+3} \setminus \{0\}$ into $\frac{2^{4r+3}-8}{5}$ quintriples, a subset with four linearly dependent elements and another subset of size three. The subset with four linearly dependent elements and the remaining element of the first row yields another recovery set and hence we obtain the following bound (note that $N_2(5, 2) = 9$).

Lemma 17: If $k \geqslant 5$ and $k \equiv 1 \pmod 4$ then $N_2(k, 2) \geqslant 2^{k-2} + \frac{2^{k-2}-8}{5} + 1 = \frac{3 \cdot 2^{k-1}-3}{5}$.

Lemmas 14, 15, 16, and 17 imply the following lower bound.

Corollary 18: If $k \geqslant 2$ is a positive integer, then $N_2(k, 2) \geqslant \left\lfloor \frac{3 \cdot 2^{k-1}+1}{5} \right\rfloor$.

Corollary 18 and Lemma 12 imply the following theorem.

Theorem 19: For $k \geqslant 2$, $N_2(k, 2) = \left\lfloor \frac{3 \cdot 2^{k-1}+1}{5} \right\rfloor$.

B. Recovery Sets for 4-Subspaces

Assume that the 4-subspace to be recovered is U which is spanned by the vectors $(\mathbf{0}, u_1)$, $(\mathbf{0}, u_2)$, $(\mathbf{0}, u_3)$, and $(\mathbf{0}, u_4)$ (U will be referred as $\mathbb{F}_2^4$).

When $d = 4$, the first row of T can be partitioned into three recovery sets of size 4 and three leftovers. Each other row of T is partitioned into three recovery sets of size 5 and one leftover.

Consider now the 3-subspace of $\mathbb{F}_2^{k-4}$ spanned by the 3-subset $\{x_1, x_2, x_3\}$, where $x_1, x_2, x_3 \in \mathbb{F}_2^{k-4}$. Consider now the seven vectors of $\mathbb{F}_2^k$,

$$\begin{aligned} R \triangleq \{&(x_1, u_1), (x_2, u_1 + u_2), (x_3, u_1 + u_3), (x_1 + x_2, \mathbf{0}), \\ &(x_1 + x_3, \mathbf{0}), (x_2 + x_3, u_2 + u_3 + u_4), \\ &(x_1 + x_2 + x_3, u_2 + u_3)\}. \end{aligned}$$

These seven vectors form a recovery set for U since we can recover $(\mathbf{0}, u_1)$, $(\mathbf{0}, u_2)$, $(\mathbf{0}, u_3)$, and $(\mathbf{0}, u_4)$ as follows:

$$\begin{aligned} (\mathbf{0}, u_1) &= (x_1, u_1) + (x_2, u_1 + u_2) + (x_3, u_1 + u_3) \\ &\quad + (x_1 + x_2 + x_3, u_2 + u_3), \\ (\mathbf{0}, u_2) &= (x_1, u_1) + (x_2, u_1 + u_2) + (x_1 + x_2, \mathbf{0}), \\ (\mathbf{0}, u_3) &= (x_1, u_1) + (x_3, u_1 + u_3) + (x_1 + x_3, \mathbf{0}), \\ (\mathbf{0}, u_4) &= (x_2, u_1 + u_2) + (x_3, u_1 + u_3) \\ &\quad + (x_2 + x_3, u_2 + u_3 + u_4). \end{aligned}$$

These seven elements of R are formed from the rows indexed by

$$x_1,\ x_2,\ x_3,\ x_1 + x_2,\ x_1 + x_3,\ x_2 + x_3,\ x_1 + x_2 + x_3,$$

in T, associated with exactly one 3-subspace of $\mathbb{F}_2^{k-4}$. This type of recovery set will be called a ***(3,4)-recovery set model***.

Since each internal row of T yields exactly one leftover and this leftover can be chosen to be any element in the row, it follows that to form more recovery sets we have to partition $\mathbb{F}_2^{k-4}$ to many 3-subspaces and some additional elements which can be used with the three leftovers of the first row to form additional recovery sets. We distinguish between three cases, depending on whether k is congruent to 1, 2, or 0 modulo 3.

Case 1: $k \equiv 1 \pmod 3$, i.e., $k - 4 \equiv 0 \pmod 3$. (note that 3 divides $k - 7$.)

By Theorem 13, there exists a $[3 \times (k-7), k-7, 3]$ MRD code $\mathcal{C}$. Let $\mathbb{C}^{\text{MRD}}$ be the lifted code of $\mathcal{C}$. Each codeword of $\mathbb{C}^{\text{MRD}}$ is a 3-subspace of $\mathbb{F}_2^{k-4}$ and any two such 3-subspaces intersect in the null-space $\{\mathbf{0}\}$. By using the (3,4)-recovery set model, each such codeword can be used to form a recovery set from the leftovers of the associated rows of T. The nonzero elements which are contained in these 2^{k-7} 3-subspaces are spanned by the rows of matrices of the form $[I_3\ A]$, where A is a $3 \times (k-7)$ binary matrix. Therefore, the only nonzero elements of $\mathbb{F}_2^{k-4}$ which are not contained in these 2^{k-4} 3-subspaces are exactly the $2^{k-7} - 1$ vectors of the form $(0, 0, 0, z_1, z_2, \ldots, z_{k-7})$, where $z_i \in \{0, 1\}$, and at least one of the z_i's is not a *zero*. This set of vectors and the all-zeros vector of length $k - 4$, form a $(k-4)$-space isomorphic to $\mathbb{F}_2^{k-4}$. We continue recursively with the same procedure. The process ends with the initial condition which is the 3-subspace $\mathbb{F}_2^3$ from which one recovery set is obtained. The outcome will be a partition of $\mathbb{F}_2^{k-4}$ into $\frac{2^{k-4}-1}{7}$ pairwise disjoint 3-subspaces. Each one of the 2^{k-4} rows of the matrix T is used to construct three recovery sets. Hence we have the following result.

Lemma 20: If $k > 6$ and $k \equiv 1 \pmod 3$ then $N_2(k, 4) \geqslant 3 \cdot 2^{k-4} + \frac{2^{k-4}-1}{7} = \frac{11 \cdot 2^{k-3}-1}{7}$.

Case 2: $k \equiv 2 \pmod 3$, i.e., $k - 4 \equiv 1 \pmod 3$.

Using the same technique as in Case 1, the process ends with the initial condition which is the 4-subspace $\mathbb{F}_2^4$. From this 4-subspace, one 3-subspace can be obtained to form one recovery set. From the eight remaining elements, 4 other linearly dependent elements can be used to obtain any element of U, which together with the three leftovers from the first row of T form another recovery set for the 4-subspace U. Thus, we have the following lemma.

Lemma 21: If $k > 6$ and $k \equiv 2 \pmod 3$, then $N_2(k, 4) \geqslant 3 \cdot 2^{k-4} + \frac{2^{k-4}-16}{7} + 2 = \frac{11 \cdot 2^{k-3}-2}{7}$.

Case 3: $k \equiv 0 \pmod 3$, i.e., $k - 4 \equiv 2 \pmod 3$.

Using the same technique as in Case 1 and Case 2, the process ends with the initial condition, i.e., the 5-subspace $\mathbb{F}_2^5$. From this 5-subspace one 3-subspace can be obtained to form one recovery set and the other 24 leftovers with the three leftovers from the first row of T can be used to obtain three more recovery sets for the 4-subspace U. The process can also be ended with the initial condition of the 8-subspace $\mathbb{F}_2^8$ for which there are 34 disjoint 3-subspaces and 17 remaining elements. With the three leftovers from the first row of T, we can construct two recovery sets to have the same result. Thus, we have the following lemma.

Lemma 22: If $k > 6$ and $k \equiv 0 \pmod 3$, then $N_2(k,4) \geqslant 3 \cdot 2^{k-4} + \frac{2^{k-4}-32}{7} + 4 = \frac{11 \cdot 2^{k-3}-4}{7}$.

Lemmas 20, 21, and 22, imply the following lower bound.

Theorem 23: For $k \geqslant 7$,

$$N_2(k,4) \geqslant \left\lfloor \frac{11 \cdot 2^{k-3} - 1}{7} \right\rfloor.$$

The case of $k = 6$ is solven in the following example.

Example 3: For $k = 6$ the matrix T has four rows indexed by $0, \beta^0, \beta, \beta^2$, where β in a primitive element is $\mathbb{F}_4$. There are 16 columns in T indexed by $\mathbf{0}, \alpha^0, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{14}$. The first three recovery sets are taken from the first row of T as follows.

$$\{(\mathbf{0}, \alpha^3), (\mathbf{0}, \alpha^4), (\mathbf{0}, \alpha^5), (\mathbf{0}, \alpha^6)\},$$
$$\{(\mathbf{0}, \alpha^7), (\mathbf{0}, \alpha^8), (\mathbf{0}, \alpha^9), (\mathbf{0}, \alpha^{10})\},$$
$$\{(\mathbf{0}, \alpha^{14}), (\mathbf{0}, \alpha^0), (\mathbf{0}, \alpha^1), (\mathbf{0}, \alpha^2)\}.$$

The next nine recovery set are taken from the other rows of T, three from each row as follows.

$$\{(\beta^i, \alpha^4), (\beta^i, \alpha^5), (\beta^i, \alpha^6), (\beta^i, \alpha^7), (\beta^i, \alpha^8)\},$$
$$\{(\beta^i, \alpha^9), (\beta^i, \alpha^{10}), (\beta^i, \alpha^{11}), (\beta^i, \alpha^{12}), (\beta^i, \alpha^{13})\},$$
$$\{(\beta^i, \mathbf{0}), (\beta^i, \alpha^0), (\beta^i, \alpha^1), (\beta^i, \alpha^2), (\beta^i, \alpha^3)\},$$

where $0 \leqslant i \leqslant 2$.

To these twelve recovery set we add the set

$$\{(\mathbf{0}, \alpha^{11}), (\mathbf{0}, \alpha^{12}), (\mathbf{0}, \alpha^{13}), (\beta^0, \alpha^{14}), (\beta^1, \alpha^{14}), (\beta^2, \alpha^{14})\}.$$

Since $(\beta^0, \alpha^{14}) + (\beta^1, \alpha^{14}) + (\beta^2, \alpha^{14}) = (\mathbf{0}, \alpha^{14})$ the last set is also a recovery set.

It can be easily verified that we cannot form more than 13 recovery sets. ■

Corollary 24: $N_2(6,4) = 13$.

The last two cases are trivial and are given in the following lemma.

Lemma 25: $N_2(5,4) = 6$ and $N_2(4,4) = 3$.

C. Recovery Sets for 5-Subspaces

Assume that the 5-subspace to be recovered is U which is spanned by the vectors $(\mathbf{0}, u_1)$, $(\mathbf{0}, u_2)$, $(\mathbf{0}, u_3)$, $(\mathbf{0}, u_4)$ and $(\mathbf{0}, u_5)$ (U will be referred as $\mathbb{F}_2^5$).

When $d = 5$, the first row of T can be partitioned into six recovery sets of size 5 and one leftover. Each other row of T is partitioned into five recovery sets of size 6 and two leftovers.

Consider now four disjoint 2-subspaces of $\mathbb{F}_2^{k-5}$ spanned by $\{x_1, x_2\}$, $\{x_3, x_4\}$, $\{x_5, x_6\}$, and $\{x_7, x_8\}$. The 24 leftovers of these four 2-subspaces are chosen and partitioned into the following three subsets of size eight:

$$\{(x_1, \mathbf{0}), (x_2, \mathbf{0}), (x_1 + x_2, u_3), (x_7, \mathbf{0}),$$
$$(x_1, u_1), (x_2, u_2), (x_1 + x_2, u_4), (x_7, u_5)\}$$
$$\{(x_3, \mathbf{0}), (x_4, \mathbf{0}), (x_3 + x_4, u_3), (x_8, \mathbf{0}),$$
$$(x_3, u_1), (x_4, u_2), (x_3 + x_4, u_4), (x_8, u_5)\},$$
$$\{(x_5, \mathbf{0}), (x_6, \mathbf{0}), (x_5 + x_6, u_3), (x_7 + x_8, \mathbf{0}),$$
$$(x_5, u_1), (x_6, u_2), (x_5 + x_6, u_4), (x_7 + x_8, u_5)\}.$$

It can be readily verified that each one of these three subsets is a recovery set for U.

Each internal row of T yields two leftovers and if one of them is chosen as $(x, \mathbf{0})$ then the second one can be chosen arbitrarily. But, if (x, u_3) was chosen, then $(x, u_4) = (x, \alpha u_3)$ must be the next element in the row, where α is a primitive element of $\mathbb{F}_{2^d}$. Therefore, to obtain these three recovery sets from the leftovers we have to partition $\mathbb{F}_2^{k-5}$ into 2-subspaces and to partition these 2-subspaces into subsets with four 2-subspaces in each one. We distinguish between odd and even k.

Case 1: If k is odd, then $k-5$ is even and hence $\mathbb{F}_2^{k-5}$ can be partitioned into $\frac{2^{k-5}-1}{3}$ 2-subspaces. These disjoint 2-subspaces can be partitioned into $\frac{2^{k-5}-4}{12} = \frac{2^{k-7}-1}{3}$ sets of four 2-subspaces and another 2-subspace. Each such set of four 2-subspaces forms three recovery sets for U for a total of $2^{k-7} - 1$ recovery sets. Assume that the additional 2-subspace is $\{x_1, x_2, x_1 + x_2\}$ and the leftover from the first row of T is $(\mathbf{0}, u_1)$. The following set of seven 1-subspaces

$$\{(\mathbf{0}, u_1), (x_1, \mathbf{0}), (x_1, u_2), (x_2, \mathbf{0}),$$
$$(x_2, u_3), (x_1 + x_2, u_4), (x_1 + x_2, u_5)\}$$

is another recovery set. Thus, we have that

Lemma 26: If $k \geqslant 7$ and k is odd, then $N_2(k,5) \geqslant 6 + 5(2^{k-5} - 1) + 2^{k-7} - 1 + 1 = 21 \cdot 2^{k-7} + 1$.

Case 2: If k is even, then $k-5$ is odd and hence $\mathbb{F}_2^{k-5}$ can be partitioned into $\frac{2^{k-5}-8}{3}$ 2-subspaces and one 3-subspace. These disjoint 2-subspaces can be partitioned into $\frac{2^{k-5}-8}{12} = \frac{2^{k-7}-2}{3}$ sets of four 2-subspaces. Each such set with four 2-subspaces forms three recovery sets for U for a total of $2^{k-7} - 2$ recovery sets. Assume that the remaining 3-subspace is $\langle x_1, x_2, x_3 \rangle$ and one leftover from the first row of T is $(\mathbf{0}, u_1)$. They can form the following two recovery sets:

$$\{(\mathbf{0}, u_1), (x_1 + x_2, \mathbf{0}), (x_1 + x_2, u_2), (x_1 + x_3, \mathbf{0}),$$
$$(x_1 + x_3, u_3), (x_2 + x_3, u_4), (x_2 + x_3, u_5)\},$$
$$\{(x_1, \mathbf{0}), (x_1, u_1), (x_2, \mathbf{0}), (x_2, u_2), (x_3, \mathbf{0}), (x_3, u_3),$$
$$(x_1 + x_2 + x_3, u_4), (x_1 + x_2 + x_3, u_5)\}.$$

Thus, we have that

Lemma 27: If $k \geqslant 7$ and k is even, then $N_2(k,5) \geqslant 6 + 5(2^{k-5} - 1) + 2^{k-7} - 2 + 2 = 21 \cdot 2^{k-7} + 1$.

IV. Upper Bounds on the Size of Recovery Sets for Binary Alphabet

We turn now to consider an upper bound on $N_2(k,4)$. We will have to distinguish again between three cases depending on whether k is congruent to 0, 1, or 2 modulo 3. We start by ignoring the first row of T and consider the other rows of T. From each such row, we have constructed 3 recovery sets of size 5 and we had one leftover. The $2^{k-4}-1$ leftovers we used to form recovery sets of size seven based on disjoint 3-subspaces. The number of disjoint 3-subspaces depends on the value of $k-4$ modulo 3. If $k-4 \equiv 0 \pmod 3$, then there are $\frac{2^{k-4}-1}{7}$ such 3-subspaces. If $k-4 \equiv 1 \pmod 3$, then there are $\frac{2^{k-4}-9}{7}$ such 3-subspaces and eight 1-subspaces are left out of this partition. If $k-4 \equiv 2 \pmod 3$, then there are $\frac{2^{k-4}-25}{7}$ such 3-subspaces and twenty four 1-subspaces are left out of this partition.

We start to check if we can construct other recovery sets by considering combinations of elements from different rows of T. We cannot gain anything by considering seven elements from different rows as this is already guaranteed by the construction of recovery sets of size seven from the leftovers. Hence, we will concentrate on recovery sets of size six to replace some recovery sets of size seven. But, each recovery set of size six from a few rows will require at least two elements from two rows. This will imply that we will have to give up on a recovery set of size five from each one of these two rows. Hence, we further need to use these rows to form more recovery sets of size six. Therefore, we are going to waste at least two elements in these rows which will make it worse than using the only one leftover in each row for a recovery set of size seven. Therefore, if we want to gain something beyond our construction it is required to use the three leftovers from the first row carefully.

Now consider to use the three leftovers from the first row (or even more elements from the first row in a recovery set that contains elements from other rows). We distinguish between three cases, depending on whether an associated recovery set contains one element from the first row, two elements from the first row, or three elements from the first row.

Case 1: Only one element from the first row is contained in the recovery set. Assume first that all the other elements of the recovery set are from different rows of T. But, this cannot be done with less than six leftovers from other rows and hence the recovery set will contain at least seven elements which does not save any element to obtain more recovery sets. If from one row we use more than two elements, then this row cannot be used anymore to form three recovery sets without elements from other rows. This implies that for each element from the first row used in a recovery set used with four elements from another row, we replace one leftover from the first row with one leftover from another row, with whom it is less flexible to form recovery sets. If we use elements from two rows then at least six elements are used in the recovery set and at least two other rows will be used so we will lose the two leftovers from the first row without any gain in the best case.

Case 2: Two elements from the first row are in one recovery set. If we use only the unique leftover from each other row, then the recovery set will be with at least seven elements which does not save any elements as in Case 1. If we use three elements from one row, then we can again use arguments similar to Case 1 to exclude this case. The same goes for generating a recovery set with six elements, two from each other row.

Case 3: All the three elements from the first row are in one recovery set. By the previous analysis, it is clear that at best we can use three leftovers. But, this does not leave us a better option for the other recovery sets than the ones used in the constructions.

This analysis implies that the lower bound of Theorem 23 is tight and hence we proved the following theorem.

Theorem 28:
For $k \geqslant 7$,

$$N_2(k,4) = \left\lfloor \frac{11 \cdot 2^{k-3} - 1}{7} \right\rfloor .$$

The analysis done in this section can be implemented for recovery sets with higher dimensions with extra complexity. By applying the integer programming technique with the careful analysis as was done for $d = 2$ and $d = 4$ we obtain the following upper bound,

$$N_2(k,5) \leqslant 21 \cdot 2^{k-7} + 2.$$

for $k \geqslant 7$. Combining Lemmas 26 and 27, this completes the proof of the following theorem.

Theorem 29:
For $k \geqslant 7$,

$$21 \cdot 2^{k-7} + 1 \leqslant N_2(k,5) \leqslant 21 \cdot 2^{k-7} + 2.$$

V. Bounds for a Larger Alphabet

When $q > 2$, the defined matrix T for $q = 2$ is no longer appropriate to represent the elements of $\mathrm{PG}(k-1,q)$. We modify the representation by defining a $\frac{q^{k-d}-1}{q-1} \times q^d$ matrix T whose rows are indexed by the points of $\mathrm{PG}(k-d-1,q)$ (or the 1-subspaces of V_q^{k-d}). It is the same as identifying these rows by words of length $k-d$ whose first nonzero entry is a *one*. The columns of the matrix T are indexed by the elements of $\mathbb{F}_{q^d}$, where the first column is indexed by $\mathbf{0}$ and the others by $\alpha^0, \alpha^1, \ldots, \alpha^{q^d-2}$, in this order, where α is a primitive element in $\mathbb{F}_{q^d}$. This matrix is the same as the one defined for $q = 2$ with the omission of the first row of T defined for $q = 2$. Entry $T(x,y)$ in the matrix T can be represented by (x,y) or $x+y$. To this matrix T we add a vector T_d whose length is $\frac{q^d-1}{q-1}$ and whose elements represent the $\frac{q^d-1}{q-1}$ one-dimensional subspaces of V_q^d. These elements can be taken as the ones in the set $\{\alpha^i : 0 \leqslant i < \frac{q^d-1}{q-1}\}$. It is readily verified that the entries of T and T_d together represent all the one-dimensional subspaces of V_q^k.

A. Constructions for Recovery Sets

We turn our attention now to construct recovery sets for d-subspaces, $d \geqslant 2$, when $q > 2$. In this subsection, we first provide a general construction with recovery sets of size d and $d+1$ as before and the leftovers from T will be combined to form recovery sets of size $d+2$.

We start by considering the case of $d = 2$. Note first that $\frac{q^2-1}{q-1} = q+1$, i.e., a 2-subspace contains $q+1$ 1-subspaces. Each 2 consecutive elements in T_2 span $\mathbb{F}_q^2$ and hence we obtain $\left\lfloor \frac{q+1}{2} \right\rfloor$ recovery sets from T_2. From each other row, we will have $\left\lfloor \frac{q^2}{3} \right\rfloor$ recovery sets and one or two leftovers, unless q is divisible by 3. This implies that we have a tight bound when $q \equiv 0 \pmod 3$ as implied by Theorem 8 and when $q \not\equiv 0 \pmod 3$ we have to apply a construction for the leftovers similar to the one in Section III-A.

When d divides $\frac{q^d-1}{q-1}$ there are no leftovers in T_d and if d does not divides $\frac{q^d-1}{q-1}$, then the number of leftovers in T_d is the reminder from the division of $\frac{q^d-1}{q-1}$ by d, i.e., an integer between 1 and $d-1$. The matrix T is of size $\frac{q^{k-d}-1}{q-1} \times q^d$ and by Lemma 7 each $d+1$ consecutive elements in a row span $\mathbb{F}_q^d$. If there are no leftovers in a row of T, then we have the following theorem which is a special case of Theorem 8.

Theorem 30: If $d+1$ divides q^d, then $N_q(k,d) = \left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor + \frac{q^k-q^d}{(d+1)(q-1)}$.

For the next construction and its analysis, it will be required to use the well-known partitions of vector spaces and projective geometries [27]. Such partitions can be found also in [13] and [26] and other places as well. For completeness, one such partition is presented in the following theorem.

Theorem 31: If d divides n, then $\mathbb{F}_q^n$ can be partitioned into $\frac{q^n-1}{q^d-1}$ pairwise disjoint (i.e., intersect in the null-space $\{\mathbf{0}\}$) d-subspaces.

Proof: Let α be a primitive element in $\mathbb{F}_{q^n}$ and $r = \frac{q^n-1}{q^d-1}$. For each i, $0 \leqslant i \leqslant r-1$, define

$$S_i \triangleq \{\alpha^i, \alpha^{r+i}, \alpha^{2r+i}, \ldots, \alpha^{(q^d-2)r+i}\} .$$

It is easy to verify that $S_0 \cup \{0\} = \mathbb{F}_{q^d}$ and hence S_0 is closed under addition. Therefore, each $S_i \cup \{0\}$ is closed under addition and hence it forms a subspace. Finally, the claim of the theorem is obtained by the isomorphism between $\mathbb{F}_{q^n}$ and $\mathbb{F}_q^n$. ∎

Another proof of Theorem 31 is by using lifted MRD codes exactly as in the constructions for the recovery sets, where $d = 2$ or $d = 4$. More information on these partitions for other parameters can be found in [12].

If there are leftovers in a row of T, then the number of leftovers in such a row is $q^d - \left\lfloor \frac{q^d}{d+1} \right\rfloor (d+1)$ and in all the $\frac{q^{k-d}-1}{q-1}$ rows of T the total number of leftovers is

$$\frac{q^{k-d}-1}{q-1} \left(q^d - \left\lfloor \frac{q^d}{d+1} \right\rfloor (d+1) \right) .$$

We choose all the leftovers in any row of T to be consecutive elements in the row as will be described now. Assume for simplicity that $d+2$ divides $q+1$ and also that $k-d$ is even. Since $k-d$ is even, it follows that the elements of $\mathrm{PG}(k-d-1,q)$ can be partitioned into 2-subspaces of $\mathbb{F}_q^{k-d}$ (1-subspaces in the projective geometry, i.e., lines of $\mathrm{PG}(k-d-1,q)$) (see Theorem 31). Each such 1-subspace has $q+1$ points. Since $d+2$ divides $q+1$, it follows that each such 1-subspace can be partitioned into $\frac{q+1}{d+2}$ subsets of size $d+2$. Let $\{x_1, x_2, \ldots, x_{d+2}\}$ be such a subset. Consider the following $d+2$ leftovers in these $d+2$ rows, one leftover for a row as follows, (x_1, u_1), (x_2, u_2),…, (x_d, u_d), $(x_{d+1}, \mathbf{0})$, $(x_{d+2}, \mathbf{0})$, where $u_1, u_2, \ldots, u_d$ are d linearly independent elements of $\mathbb{F}_q^d$. For any i, $1 \leqslant i \leqslant d$, we have that x_i, x_{d+1}, x_{d+2} are linearly dependent since they are all contained in the same 1-subspace. Hence, from (x_i, u_i), $(x_{d+1}, \mathbf{0})$, $(x_{d+2}, \mathbf{0})$ we can recover $(\mathbf{0}, u_i)$. Therefore, the elements $(\mathbf{0}, u_1)$, $(\mathbf{0}, u_2)$,…,$(\mathbf{0}, u_d)$ can be recovered and hence U can be recovered. If each row of T has exactly one leftover, then this is enough and we have proved the following lemma.

Lemma 32: If $q^d \equiv 1 \pmod{d+1}$, $d+2$ divides $q+1$, and $k-d$ is even, then

$$N_q(k,d) \geqslant \left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor + \frac{q^{k-d}-1}{q-1} \left\lfloor \frac{q^d}{d+1} \right\rfloor + \frac{q^{k-d}-1}{(d+2)(q-1)}$$

Based on the analysis done before, we know that we cannot have more recovery sets of size d or $d+1$ and hence it is easily verified that the lower bound of Lemma 32 is tight.

Theorem 33: If $q^d \equiv 1 \pmod{d+1}$, $d+2$ divides $q+1$, and $k-d$ is even, then

$$N_q(k,d) = \left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor + \frac{q^{k-d}-1}{q-1} \left\lfloor \frac{q^d}{d+1} \right\rfloor + \frac{q^{k-d}-1}{(d+2)(q-1)}$$

Theorem 33 is a special case of the next theorem which will be analyzed now.

Assume now that each row of T has at least two leftovers. Again, since $d+2$ divides $q+1$, it follows that each 1-subspace, in the partition into 2-subspaces of $\mathbb{F}_q^{k-d}$, can be partitioned into $\frac{q+1}{d+2}$ subsets, each of size $d+2$. Let $\{x_1, x_2, \ldots, x_{d+2}\}$ be such a subset. Consider the following $d+2$ leftovers in these $d+2$ rows, one leftover for a row as follows, (x_1, u_1), (x_2, u_2),…, (x_d, u_d), (x_{d+1}, u_1), (x_{d+2}, u_1), where $u_1, u_2, \ldots, u_d$ are d linearly independent elements. We can recover the two elements $(x_{d+1} - x_1, \mathbf{0})$ and $(x_{d+2} - x_1, \mathbf{0})$. For any i, $1 \leqslant i \leqslant d$, we have that x_i, $x_{d+1} - x_1$, and $x_{d+2} - x_1$ are linearly dependent since they are all contained in the same 1-subspace. Hence, from the three elements (x_i, u_i), $(x_{d+1} - x_1, \mathbf{0})$, $(x_{d+2} - x_1, \mathbf{0})$ we can recover $(\mathbf{0}, u_i)$. Therefore, $(\mathbf{0}, u_1)$, $(\mathbf{0}, u_2)$,…,$(\mathbf{0}, u_d)$ can be recovered and hence U can be recovered too. Now, we choose a second leftover for each of these rows, $(x_1, \alpha u_1)$, $(x_2, \alpha u_2), \cdots, (x_d, \alpha u_d)$, $(x_{d+1}, \alpha u_1)$, $(x_{d+2}, \alpha u_1)$. Since the columns are indexed by consecutive powers of α, it follows that each two elements chosen in these rows are consecutive and hence they can be chosen as leftovers. The subset U is recovered from these $d+2$ leftovers in the same way as it was recovered from the first $d+2$ leftovers that were chosen. If there are ℓ leftovers in a row, then in these rows we choose the leftovers as $(x_1, \alpha^i u_1)$, $(x_2, \alpha^i u_2)$,…,$(x_d, \alpha^i u_d)$, $(x_{d+1}, \alpha^i u_1)$, $(x_{d+2}, \alpha^i u_1)$, for each i, $0 \leqslant i \leqslant \ell - 1$. For each such i, the subspace U is recovered in the same way.

This construction implies the proof for the following theorem.

Theorem 34: If $q \in \mathbb{P}$, $q > 2$, $d > 1$, $k-d$ even, and $q+1 = \ell(d+2)$, then

$$N_q(k,d) = \left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor + \frac{q^{k-d}-1}{q-1} \left\lfloor \frac{q^d}{d+1} \right\rfloor + \frac{q^{k-d}-1}{q-1} \left(\frac{q^d}{d+2} - \left\lfloor \frac{q^d}{d+1} \right\rfloor \frac{d+1}{d+2} \right).$$

A similar analysis can be done when $k-d$ is odd. The main difference from even $k-d$ is that when $k-d$ is odd the elements of $\mathrm{PG}(k-d-1,q)$ can be partitioned into 2-subspaces of $\mathbb{F}_q^{k-d}$ and q^2 1-subspaces [2], [18].

B. Upper Bound on the Number of Recovery Sets

For more upper bounds on the number of recovery sets, we will not distinguish between the binary and the non-binary case. We have already demonstrated upper bounds based on integer programming and direct analysis. We will develop now another upper bound based on a direct analysis of possible sizes of recovery sets. We must use at least d elements from T_d to recover a given d-subspace. From any given row of T at least $d+1$ elements are required to recover a subspace. Any subset of d elements from two rows of T, with at least one element from each row span a subspace $\mathcal{V}$, where $\dim(\mathcal{V} \cap \mathbb{F}_q^d) \leqslant d-2$ and hence at least $d+2$ elements are required if the recovery set contains at least one element from two distinct rows of T. This will immediately improve the bound of Theorem 4 to

Theorem 35: If $q \in \mathbb{P}$ and $k \geqslant d$ is a positive integer, then

$$N_q(k,d) \leqslant \left\lfloor \frac{q^d-1}{d(q-1)} \right\rfloor + \frac{q^{k-d}-1}{q-1} \left\lfloor \frac{q^k}{d+1} \right\rfloor + \left\lfloor \frac{2\ell + \frac{q^{k-d}-1}{q-1} t}{d+2} \right\rfloor.$$

where ℓ is the reminder from the division of $\frac{q^d-1}{q-1}$ by d and t is the reminder from the division of q^d by $d+1$.

The additional ℓ in the last summation in Theorem 35 is required in case that each of the ℓ leftovers from T_d can be combined with leftovers from T to form a recovery set of size $d+1$. The upper bound of Theorem 35 can be further improved, but this will be left for future research.

VI. Conclusion and Problems for Future Research

We have looked for the maximum number of recovery sets that can be obtained for any given d-subspace of $\mathbb{F}_q^k$ when the stored elements are all the 1-subspaces of $\mathbb{F}_q^k$, i.e., these are the columns of the generator matrix of the $[(q^k-1)/(q-1), k, q^{k-1}]$ simplex code, which are also the points of the projective geometry $\mathrm{PG}(k-1,q)$. Lower and upper bounds on the number of recovery sets are provided, some of which are tight. Similar bounds can be given using similar techniques. For example, the following bound was obtained and it will be given without details due to its length.

Theorem 36: For $k \geqslant 7$,

$$\left\lfloor \frac{91 \cdot 2^{k-6} + 12}{10} \right\rfloor \leqslant N_2(k,6) \leqslant \left\lfloor \frac{91 \cdot 2^{k-6} + 35}{10} \right\rfloor.$$

The main open problems are associated with partitions of the leftovers into recovery sets. There are several problems whose solutions will yield an optimal or almost optimal number of recovery sets:

- For $d \leqslant q-1$, find a partition of the 1-subspaces of $\mathbb{F}_q^n$ into $(d+2)$-subsets and possible one subset of a smaller size, such that each $(d+2)$-subset spans a 2-subspace.
- When $d > q-1$ and k large, what is the minimum number of leftovers from different rows of T which are required to form one recovery set? Find a partition of the 1-subspaces of $\mathbb{F}_q^n$ into subsets of this minimum size, such that each one can form a recovery set.
- How many leftovers (not from T_d or the first row of T in the binary case) are required to form one d-subspace, where there is a single leftover in each row?
- What is the size of the recovery sets that are obtained from the leftovers when $q=2$ and $2^d \equiv 1 \pmod{d+1}$, i.e., one leftover from each internal row of T?
- Write a program to produce an automatic integer programming problem for the number of recovery sets. The program should be able to develop inequalities as in Section II-B and as analyzed in Section IV. Use the program to improve the upper bounds.

References

[1] H. Asi and E. Yaakobi, "Nearly optimal constructions of PIR and batch codes," *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 947–964, Feb. 2019.

[2] A. Beutelspacher, "Partial spreads in finite projective spaces and partial designs," *Mathematische Zeitschrift*, vol. 145, no. 3, pp. 211–229, Oct. 1975.

[3] S. R. Blackburn and T. Etzion, "PIR array codes with optimal virtual server rate," *IEEE Trans. Inf. Theory*, vol. 65, no. 10, pp. 6136–6145, Oct. 2019.

[4] M. Braun, T. Etzion, P. R. J. Östergård, A. Vardy, and A. Wassermann, "Existence of q-analogs of Steiner systems," *Forum Math., Pi*, vol. 4, pp. 1–14, Aug. 2016.

[5] Y. M. Chee, T. Etzion, H. M. Kiah, and H. Zhang, "Recovery sets for subspaces from a vector space," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 542–547.

[6] Y. M. Chee, H. M. Kiah, E. Yaakobi, and H. Zhang, "A generalization of the blackburn-etzion construction for private information retrieval array codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 1062–1066.

[7] V. Chvatal, *Linear Programming*. New York, NY, USA: Macmillan, 1983.

[8] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. Combinat. Theory A*, vol. 25, no. 3, pp. 226–241, Nov. 1978.

[9] T. Etzion, *Perfect Codes and Related Structures*. Singapore: World Scientific, 2022.

[10] T. Etzion and N. Silberstein, "Error-correcting codes in projective spaces via rank-metric codes and Ferrers diagrams," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 2909–2919, Jul. 2009.

[11] T. Etzion and N. Silberstein, "Codes and designs related to lifted MRD codes," *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 1004–1017, Feb. 2013.

[12] T. Etzion and L. Storme, "Galois geometries and coding theory," *Des., Codes, Cryptogr.*, vol. 78, pp. 311–350, Jan. 2016.

[13] T. Etzion and A. Vardy, "Error-correcting codes in projective space," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1165–1173, Feb. 2011.

[14] A. Fazeli, A. Vardy, and E. Yaakobi, "Private information retrieval without storage overhead: Coding instead of replication," May 2015, *arXiv:1505.00624*.
[15] A. Fazeli, A. Vardy, and E. Yaakobi, "Codes for distributed PIR with low storage overhead," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Hong Kong, Jun. 2015, pp. 2852–2856.
[16] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Problems Inform. Transmiss.*, vol. 21, no. 1, pp. 1–12, 1985.
[17] H. D. L. Hollmann, K. Khathuria, A.-E. Riet, and V. Skachek, "On some batch code properties of the simplex code," *Des., Codes Cryptogr.*, vol. 91, no. 5, pp. 1595–1605, May 2023.
[18] S. J. Hong and A. M. Patel, "A general class of maximal codes ror computer applications," *IEEE Trans. Comput.*, vol. C-21, no. 12, pp. 1322–1331, Dec. 1972.
[19] A. Kohnert and S. Kurz, "Construction of large constant dimension codes with a prescribed minimum distance," in *Mathematical Methods in Computer Science* (Lecture Notes in Computer Science), vol. 5393. Berlin, Germany: Springer, 2008, pp. 31–42.
[20] R. Kötter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.
[21] S. Kurz and E. Yaakobi, "PIR codes with short block length," *Des., Codes Cryptogr.*, vol. 89, no. 3, pp. 559–587, Mar. 2021.
[22] M. Nassar and E. Yaakobi, "Array codes for functional PIR and batch codes," *IEEE Trans. Inf. Theory*, vol. 68, no. 2, pp. 839–862, Feb. 2022.
[23] N. Raviv and T. Etzion, "Distributed storage systems based on intersecting subspace codes," in *Proc. Int. Symp. Inf. Theory*, Hong Kong, 2015, pp. 1462–1466.
[24] A. S. Rawat, D. S. Papailiopoulos, A. G. Dimakis, and S. Vishwanath, "Locality and availability in distributed storage," *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4481–4493, Aug. 2016.
[25] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. Inf. Theory*, vol. 37, no. 2, pp. 328–336, Mar. 1991.
[26] M. Schwartz and T. Etzion, "Codes and anticodes in the Grassman graph," *J. Combin. Theory A*, vol. 97, no. 1, pp. 27–42, 2002.
[27] B. Segre, "Teoria di galois, fibrazioni proiettive e geometrie non desarguesiane," *Annali di Matematica Pura ed Applicata*, vol. 64, no. 1, pp. 1–76, Dec. 1964.
[28] N. Silberstein, T. Etzion, and M. Schwartz, "Locality and availability of array codes constructed from subspaces," *IEEE Trans. Inf. Theory*, vol. 65, no. 5, pp. 2648–2660, May 2019.
[29] D. Silva, F. R. Kschischang, and R. Kötter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, Sep. 2008.
[30] A. Vardy and E. Yaakobi, "Private information retrieval without storage overhead: Coding instead of replication," *IEEE J. Sel. Areas Inf. Theory*, vol. 4, pp. 286–301, 2023.
[31] Z. Wang, H. M. Kiah, Y. Cassuto, and J. Bruck, "Switch codes: Codes for fully parallel reconstruction," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2061–2075, Apr. 2017.
[32] L. Yohananov and E. Yaakobi, "Almost optimal construction of functional batch codes using extended simplex codes," *IEEE Trans. Inf. Theory*, vol. 68, no. 10, pp. 6434–6451, Oct. 2022.
[33] Y. Zhang, T. Etzion, and E. Yaakobi, "Bounds on the length of functional PIR and batch codes," *IEEE Trans. Inf. Theory*, vol. 66, no. 8, pp. 4917–4934, Aug. 2020.

Yeow Meng Chee (Senior Member, IEEE) received the B.Math. degree in computer science, and combinatorics and optimization and the M.Math. and Ph.D. degrees in computer science from the University of Waterloo, Waterloo, ON, Canada, in 1988, 1989, and 1996, respectively. He is currently a Professor of design and engineering with the National University of Singapore. Prior to this, he was a Professor of mathematical sciences with Nanyang Technological University, the Program Director of interactive digital media research and development with the Media Development Authority of Singapore, a Post-Doctoral Fellow with the University of Waterloo and the IBM's Zurich Research Laboratory, the General Manager of Singapore Computer Emergency Response Team, and the Deputy Director of Strategic Programs at the Infocomm Development Authority, Singapore. His research interests include the interplay between combinatorics and computer science/engineering, particularly in combinatorial design theory, coding theory, extremal set systems, and their applications. He is a fellow of the Institute of Combinatorics and its Applications. He is an Editor of the *Journal of Combinatorial Theory, Series A*.

Tuvi Etzion (Life Fellow, IEEE) was born in Tel Aviv, Israel, in 1956. He received the B.A., M.Sc., and D.Sc. degrees from the Technion—Israel Institute of Technology, Haifa, Israel, in 1980, 1982, and 1984, respectively.

In 1984, he held a position at the Department of Computer Science, Technion—Israel Institute of Technology, where he is currently the Bernard Elkin Chair in computer science. From 1985 to 1987, he was a Visiting Research Professor at the Department of Electrical Engineering (Systems), University of Southern California, Los Angeles. During the Summer of 1990 and 1991, he was a Visiting Research at Bellcore, Morristown, NJ, USA. From 1994 to 1996, he was a Visiting Research Fellow at the Computer Science Department, Royal Holloway, University of London, Egham, U.K. He also had several visits to the Coordinated Science Laboratory, University of Illinois Urbana–Champaign, from 1995 to 1998; two visits to HP Bristol during the Summer of 1996 and 2000; a few visits to the Department of Electrical Engineering, University of California at San Diego, from 2000 to 2022; several visits to the Mathematics Department, Royal Holloway, University of London, from 2007 to 2022; a few visits to the School of Physical and Mathematical Science (SPMS), Nanyang Technological University, and the Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore, from 2016 to 2019, and 2022; and a few visits to Jiaotong University, Beijing, from 2017 to 2019. His research interests include applications of discrete mathematics to problems in computer science and information theory, coding theory, coding for storage, and combinatorial designs.

Dr. Etzion was an Associate Editor for Coding Theory for the IEEE TRANSACTIONS ON INFORMATION THEORY from 2006 to 2009. From 2004 to 2009, he was an Editor for the *Journal of Combinatorial Designs*. He has been an Editor for *Designs, Codes, and Cryptography* since 2011 and *Advances in Mathematics of Communications* since 2013. He has been the Editor-in-Chief for *Journal of Combinatorial Theory, Series A*, since 2021.

Han Mao Kiah (Senior Member, IEEE) received the Ph.D. degree in mathematics from Nanyang Technological University (NTU), Singapore, in 2014. From 2014 to 2015, he was a Post-Doctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign. From 2015 to 2018, he was a Lecturer with the School of Physical and Mathematical Sciences (SPMS), NTU, where he is currently an Assistant Professor. His research interests include DNA-based data storage, coding theory, enumerative combinatorics, and combinatorial design theory.

Hui Zhang (Member, IEEE) received the Ph.D. degree in applied mathematics from Zhejiang University, China, in 2013. From 2013 to 2023, she was a Research Fellow or a Post-Doctoral Fellow at various institutions, including Nanyang Technological University, the University of Tartu, the Technion—Israel Institute of Technology, and the National University of Singapore, focusing on research areas, such as combinatorial theory, coding theory, cryptography, and their intersections.