How to Build Robust Shared Control Systems

Ross Anderson[1], Cunsheng Ding[2], Tor Helleseth[3], Torleiv Kløve[3]

[1] Computer Laboratory, Pembroke Street, Cambridge, CB2 3QG
[2] Turku Centre for Computer Science, Fin-20520 Turku, Finland
[3] Department of Informatics, University of Bergen, HIB, N-5020 Bergen, Norway
Institutt for informatikk, Universitetet i Bergen, Bergen, Norway; torh@ii.uib.no, torleiv@ii.uib.no

Abstract. Previous researchers have designed shared control schemes with a view to minimising the likelihood that participants will conspire to perform an unauthorised act. But, human nature being what it is, systems inevitably fail; so shared control schemes should also be designed so that the police can identify conspirators after the fact. This requirement leads us to search for schemes with sparse access structures. We show how this can be done using ideas from coding theory. In particular, secret sharing schemes based on geometric codes whose dual $[n, k, d]$ codes have d and n as their only nonzero weights are suitable. We determine their access structures and analyse their properties. We have found almost all of them, and established some relations among codes, designs and secret-sharing schemes.

1 The problems with existing shared control schemes

The origins of shared control schemes are lost in the mists of time. For generations, banks have had dual keys for strongrooms, and rules that instruments over a certain threshold of value needed to be signed by two or more managers. Such schemes do not just protect bank; they also protect managers from having their families taken hostage.

With the advent of electronic banking, similar functionality has often been implemented using obvious mechanisms. For example, the cryptographic keys used to initialise automatic teller machines are typically constructed by exclusive-or'ing two 56 bit components together. The components are kept on paper under the physical control of two different people — typically the branch manager and the branch accountant. More sensitive keys, such as interbank master keys, are similarly broken into three components [13].

Some problems with such schemes are described in [1]. It is inconvenient to have a really strict separation between (say) "orange" and "blue" tellers, so someone who has had access to one half of a key or password one month may get access to the other half the next month. This problem has led to real losses.

A more sophisticated approach, *secret sharing*, was introduced in 1979 by Blakley [3] and Shamir [12] simultaneously. Their idea was to construct schemes in which a dealer can split up a secret into a number of shares and distribute them to a group of participants. Certain authorised subsets of this group — *access sets* — can then combine their shares to recover the secret (the set of access sets is called the *access structure*). Such schemes allow each participant to have his own unique secret information; the custody of passwords and keys never has to pass from one individual to another, and thus hopefully we can avoid the risks mentioned above.

The simplest kind of secret sharing system is the *threshold scheme* in which any m out of n participants can recover the secret. A practical construction for sharing digital signature keys in this way is given by Desmedt and Frankel [5]; this is used, for example, in the Omega key management service [11].

However, threshold schemes are not ideal in the banking world, and a simple example will make this clear. Suppose we have a vault with an access structure of "any three out of six tellers". One day, the vault is empty. The police investigate and find that precisely one of the tellers has a watertight alibi. This leaves $\binom{5}{3} = 10$ possible 'minimal conspiracies', and in the absence of further information, the conspirators may well get away with it.

But suppose the vault's access structure had been

{1,2,4}, {3,4,5}, {2,5,6}, {1,3,6}

Then if, for example, it was teller 1 who had a solid alibi, we know that the conspiracy must include either tellers 3, 4 and 5, or tellers 2, 5 and 6. So we know that teller 5 is guilty for sure; hopefully he can be persuaded to name his accomplices by the carrot of a more lenient jail sentence.

Note also that the above access structure is still fairly resilient in that it always allows access to the vault if one teller is off sick, while if two are off simultaneously, it can still be opened 80% of the time (the blocking combinations are {1, 5}, {2, 3}, and {4, 6} — and this is clearly optimal, if an alibi for one teller is to mean a conviction for another). So it is indeed a practical approach to shared control. But how did we find this particular access structure, and how can we construct shared control schemes with any desired (and practical) combination of resilience and cheater detection?

2 Sparse access structures based on linear codes

In the above section, we motivated the need for secret sharing schemes with sparse access structures. We could always construct them by choosing k access sets of m participants at random from a total of n participants, but schemes would be inefficient (the details are left as an exercise for the reader). In what follows, we will develop techniques for the systematic construction of secret sharing schemes with sparse access structures.

The inspiration comes from the following observation. In the m out of n threshold scheme, there are $\binom{n}{m}$ *minimal access sets* (access sets of which no proper subset is also an access set). It is this large number of minimal access sets that makes it hard for the police to identify conspirators. Now recall that an $[m, m, 1]$ linear code over $GF(q)$ has the maximum possible number of codewords, and because of this the code has no capacity to detect (let alone correct) errors. In some sense m out of n threshold schemes are like $[m, m, 1]$ linear codes.

This suggests that we should turn to the theory of linear codes to look for systematic constructions.

2.1 Notation and previous work

Now let the set of participants be $\mathbf{P}$ and the access structure be $\Gamma \subseteq 2^{\mathbf{P}}$. Recall that an $[n, k; q]$ code is a k-dimensional subspace of $GF(q)^n$ whose elements are called codewords. The (Hamming) weight of a codeword $\mathbf{c}$ is the number of nonzero positions in $\mathbf{c}$. The *minimum distance* d of the code is the smallest (Hamming) distance between any two distinct code words. Because of linearity, this is also the smallest weight of a nonzero codeword. Sometimes we include d in the notation and describe the code as an $[n, k, d; q]$ code. A generator matrix G of an $[n, k; q]$ code $\mathcal{C}$ is a $k \times n$ matrix over $GF(q)$ whose rows form a basis for $\mathcal{C}$.

One approach to the construction of secret-sharing schemes based on linear codes is as follows. Choose an $[n+1, k; q]$ code $\mathcal{C}$. Let G be a generator matrix of $\mathcal{C}$. Let $s \in GF(q)$ denote the secret, and $\mathbf{g}_0 = (g_{00}, g_{10}, \cdots, g_{k-1,0})^T$ be the first column of the generator matrix G. Then the information vector $\mathbf{s} = (s_0, \cdots, s_{k-1})$ is chosen to be any vector of $GF(q)^k$ such that $s = \mathbf{s}\mathbf{g}_0 = \sum_{i=0}^{k-1} s_i g_{i0}$.

The codeword corresponding to this information vector $\mathbf{s}$ is $\mathbf{t} = (t_0, t_1, \cdots, t_n) = \mathbf{s}G$. We give t_i to the party p_i as their share, and the first component $t_0 = s$ of the codeword $\mathbf{t}$ is the secret.

It is not hard to prove that in the secret sharing scheme based on a generator matrix $G = [\mathbf{g}_0 \mathbf{g}_1 \cdots \mathbf{g}_n]$ of an $[n+1, k; q]$ linear code such that $\mathbf{g}_0$ is a linear combination of the other n columns $\mathbf{g}_1, \cdots, \mathbf{g}_n$, the secret t_0 is determined by the set of shares $\{t_{i_1}, \cdots, t_{i_m}\}$ if and only if $\mathbf{g}_0$ is a linear combination of the vectors $\mathbf{g}_{i_1}, \cdots, \mathbf{g}_{i_m}$, where $1 \le i_1 < \cdots < i_m \le n$ and $m \le n$.

Computing the secret is straightforward: solve the linear equation $\mathbf{g}_0 = \sum_{j=1}^{m} x_j \mathbf{g}_{i_j}$ to find x_j, and the secret is then given by $t_0 = \mathbf{s}\mathbf{g}_0 = \sum_{j=1}^{m} x_j \mathbf{s}\mathbf{g}_{i_j} = \sum_{j=1}^{m} x_j t_{i_j}$.

Secret sharing schemes based on this general approach were considered by Karnin, Green and Hellman [6], and by Massey [8, 9]. The approach of McEliece and Sarwate is different but closely related [10].

2.2 Massey's lemma

For secret sharing schemes based on the Karnin-Green-Hellman approach, Massey introduced the concept of minimal codewords and characterised the resulting access structures [8, 9]. We state his characterization in the following lemma which will be needed in later sections.

Lemma 2.1. *Let G be a generator matrix of an $[n+1, k; q]$ code $\mathcal{C}$, and let $\mathcal{C}^{\perp}$ be the dual code of $\mathcal{C}$. In the secret-sharing scheme based on G, a set of shares $\{t_{i_1}, t_{i_2}, \cdots, t_{i_m}\}$ determines the secret if and only if there is a codeword*

$$(1, 0, \cdots, 0, c_{i_1}, 0, \cdots, 0, c_{i_m}, 0, \cdots, 0)$$

in the dual code $\mathcal{C}^{\perp}$, where all $c_{i_j} \neq 0$ for $j = 1, 2, \cdots, m$, $1 \leq i_1 < \cdots < i_m \leq n$ and $1 \leq m \leq n$.

We also mention the fact that for secret sharing schemes based on the above approach, a set of shares either determines the secret or gives no information about it, i.e., such schemes are *perfect*. This fact and Massey's lemma will be used to determine the access structure of some schemes based on Reed-Muller, Hamming and other codes.

2.3 Democratic secret sharing schemes

For a perfect secret sharing scheme, a group of participants can determine the secret if and only if it contains one minimal access set. Thus, the determination of the access structure is simply that of the minimal access sets. We call a secret-sharing scheme *democratic* if each party serves on the same number of minimal access sets; so it is democratic if these sets form a 1-design.

In this paper we describe some perfect and democratic secret-sharing schemes based on Reed-Muller and Hamming codes and analyse their properties. Our schemes are based on $[n, k, d; q]$ codes such that all the nonzero codewords have weight d or n. We have found almost all secret sharing schemes based on such codes whose minimal access sets form a 1-design. We have also established some connections between linear codes, designs and secret sharing schemes.

3 Designs, codes and secret sharing

Let X be a v-set (i.e., a set with v elements) whose elements are called points. A t-(v, k, λ) design is a collection of distinct k-subsets (called blocks) of X with the property that any t-subset of X is contained in exactly λ blocks. A Steiner system is a t-design with $\lambda = 1$ and $t \geq 2$.

To prove the main theorem of this section, we need a number of known results in design theory which are summarized in the following two lemmas.

Lemma 3.1. [7, p.59] *In a t-(v, k, λ) design, let $p_1, p_2, \cdots, p_t$ be any t distinct points. Let λ_i be the number of blocks containing $p_1, \cdots, p_i$ for $1 \leq i \leq t$, and let λ_0 be the total number of blocks. Then λ_i is independent of the choice of $p_1, \cdots, p_i$ and*

$$\lambda_i = \frac{\lambda\binom{v-i}{t-i}}{\binom{k-i}{t-i}}, \quad \text{for } 0 \leq i \leq t.$$

By this lemma a t-(v,k,λ) design is also an i-(v,k,λ_i) design for $1 \le i \le t$. One way to get a $(t-1)$-$(v-1,k-1,\lambda)$ design from a t-(v,k,λ) design is the following. Let $\mathcal{D} = (\mathcal{P}, \mathcal{B})$ be a t-(v,k,λ) design, where $\mathcal{P}$ and $\mathcal{B}$ are the set of points and the set of blocks respectively. Take all the blocks of $\mathcal{B}$ that contain a point $p \in \mathcal{B}$, and omit the point in those blocks. Denote the new blocks by $\mathcal{D}_2$. Then we have the following result.

Lemma 3.2. [7, p.62] *$\mathcal{D}_2 = (\mathcal{P} - \{p\}, \mathcal{B}_2)$ forms a $(t-1)$-$(v-1,k-1,\lambda)$ design.*

Let $\mathcal{C}$ be an $[n+1,k;q]$ code, and let A_i be the number of codewords of $\mathcal{C}$ with Hamming weight i. The sequence $\{A_i\}$ is called the weight distribution of $\mathcal{C}$. Obviously, $A_0 = 1$. The weights $0, \sigma_1, \cdots, \sigma_s$ are the subscripts of those $A_i \neq 0$. Let $\mathbf{c} = (c_0, c_1, \cdots, c_n)$ be a codeword over $GF(q)$ with Hamming weight w. The set of w subscripts i with $c_i \neq 0$ is called the support of $\mathbf{c}$. Clearly, all $q-1$ multiples of a codeword have the same support. It is also possible that two linearly independent codewords have the same support. In the sequel a support is counted only once.

Our main result on secret sharing in this section is the following.

Theorem 3.3. *Let $\mathcal{C}^\perp$ be an $[n+1, n-k+1, d; q]$ code with d and $n+1$ as its only nonzero weights, where $n+1 > d \ge 3$, and $\mathcal{C}^\perp$ is the dual code of $\mathcal{C}$. Suppose there is an integer t with $2 \le t < d$ such that there are at most $d-t$ σ_i's in the range $1 \le \sigma_i \le n+1-t$, where these σ_i are the weights of $\mathcal{C}$. Let t be the largest such t. Then the secret-sharing scheme based on $\mathcal{C}$ has the following properties:*

1. *There are $dA_d/(n+1)(q-1)$ minimal access sets consisting of $d-1$ participants that can determine the secret, and any group of participants can determine the secret if and only if it contains one of them (thus, no group of less than $d-1$ parties can determine the secret).*
2. *Every set of $t-1$ parties serves on exactly*

$$\frac{\binom{d}{t} A_d}{\binom{n+1}{t}(q-1)}$$

minimal access sets.

Proof: By the Assmus-Mattson Theorem [7, p.177], the supports of the codewords of weight d in $\mathcal{C}^\perp$ form a t-$(n+1,d,\lambda)$ design. By Lemma 3.1

$$\frac{A_d}{q-1} = \frac{\lambda \binom{n+1}{t}}{\binom{d}{t}}.$$

Hence

$$\lambda = \frac{A_d \binom{d}{t}}{\binom{n+1}{t}(q-1)}.$$

Let $0, 1, \cdots, n$ be the subscripts of the coordinates of the codewords of $\mathcal{C}^\perp$ and let

$$\mathcal{E} = \{E_1, E_2, \cdots, E_{A_d/(q-1)}\}$$

be the set of all supports of codewords of weight d in $\mathcal{C}^\perp$. By Lemma 3.1 the number of supports in $\mathcal{E}$ containing the point 0 is

$$\frac{\lambda\binom{n}{t-1}}{\binom{d-1}{t-1}} = \frac{\binom{n}{t-1}}{\binom{d}{t}(q-1)} = \frac{dA_d}{(n+1)(q-1)}.$$

Without loss of generality, assume that $E_1, E_2, \cdots, E_{dA_d/(n+1)(q-1)}$ are the supports in $\mathcal{E}$ containing 0. Then the $dA_d/(n+1)(q-1)$ minimal access sets are

$$E_1 \setminus \{0\}, \cdots, E_{dA_d/(n+1)(q-1)} \setminus \{0\}.$$

By Lemma 2.1 and the assumptions of the theorem, each of these minimal access sets can determine the secret, and every group of parties determining the secret must contain one of them.

By Lemma 3.2 these minimal access sets form a $(t-1)$-$(n, d-1, \lambda)$ design. Hence, every set of $t-1$ parties serves on exactly $\binom{d}{t}A_d/\binom{n+1}{t}(q-1)$ minimal access sets. **Q.E.D.**

Thus our secret sharing schemes are 'almost democratic', in the sense that all groups of $t-1$ parties serve on the same number of minimal access sets. The larger the parameter t, the more democratic the scheme; but there is a tradeoff between the size of t and the number of minimal access sets. For instance, when $t = n$, the code $\mathcal{C}^\perp$ has dimension one and the access structure is only

$$\{1, 2, \cdots, n\}.$$

Generally, the larger the size of t, the fewer the minimal access sets, and vice versa.

The conditions in Theorem 3.3 are sufficient for the supports of the codewords of minimum weight to form a t-design, but they are not easy to check. Since we are mostly interested in 2-designs for our secret sharing purpose, we prove the following result.

Lemma 3.4. *Let $\mathcal{C}$ be an $[n, k, d; q]$ code with d and n as its only nonzero weights, where $k \geq 2$. Let G be a generator matrix of $\mathcal{C}$ such that no column of G is a zero vector. Then the supports of the codewords with weight d in $\mathcal{C}$ form a 2-design if and only if every two columns of G are linearly independent, or equivalently the minimum distance $d^\perp$ of the dual code $\mathcal{C}^\perp$ is at least 3.*

Proof: Let $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n$ be the columns of G. Since $k \geq 2$, at least two columns of G are linearly independent. Without loss of generality, assume that $\mathbf{g}_1$ and $\mathbf{g}_2$ are linearly independent. Let

$$A(1,2) = [\mathbf{g}_1 \mathbf{g}_2] = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_k \end{bmatrix}.$$

where $\mathbf{a}_i = (g_{i,1} g_{i,2})$ for $1 \leq i \leq k$. Since $A(1,2)$ has rank 2, without loss of generality we can assume that $\mathbf{a}_1$ and $\mathbf{a}_2$ are linearly independent. Clearly, the sum $\sum_{i=1}^{k} y_i \mathbf{a}_i$ takes on each element of $GF(q)^2$ equally often, i.e., q^{k-2} times, when $(y_1, y_2, \cdots, y_k)$ run through $GF(q)^k$. Let A_n be the number of codewords with weight n in $\mathcal{C}$. Then the number of supports of codewords with weights d in $\mathcal{C}$ containing the positions $\{1, 2\}$ is

$$(q-1)^2 q^{k-2} - A_n.$$

Suppose that two columns $\mathbf{g}_1$ and $\mathbf{g}_2$ are linearly dependent, so the rank of the matrix $A(1,2)$ is one. Then it is easily seen that the number of supports containing the positions $\{1, 2\}$ of codewords with weight d in $\mathcal{C}$ is

$$(q-1) q^{k-1} - A_n.$$

Thus, the supports of codewords of weight d in $\mathcal{C}$ form a 2-design if and only if every two columns of G are linearly independent. This proves the lemma. **Q.E.D.**

Now let $\mathcal{C}$ be any $[n, k, d; q]$ code. Define for $s \geq 2$

$$\mathcal{C}_s = \{ (\overbrace{\mathbf{c}|\mathbf{c}|\cdots|\mathbf{c}}^{s}) : \mathbf{c} \in \mathcal{C} \}.$$

This is a $[sn, k, sd; q]$ code consisting of s copies of $\mathcal{C}$.

Corollary 3.5. *If $s \geq 2$ and $k \geq 2$ the supports of the codewords of weight sd of $\mathcal{C}_s$ are not a 2-design.*

In this paper we have only considered secret sharing schemes based on $[n, k, d]$ codes with nonzero weights d and n. For secret sharing schemes based on codes with three or more weights this lemma could show the extent of democracy.

4 On $[n, k, d]$ codes with weights 0, d, and n

We now search for all secret-sharing schemes based on the dual codes of $[n, k, d]$ codes with d and n as their only nonzero weights and whose minimal access sets form a 1-design. We first need to search for all $[n, k, d]$ codes with only weights $0, d, n$.

Two $[n, k; q]$ codes $\mathcal{C}$ and $\mathcal{C}'$ are *equivalent* if there exist a permutation π of $\{1, 2, \ldots, n\}$ and nonzero elements $\alpha_1, \alpha_2, \ldots, \alpha_n$ in $GF(q)$ such that

$$\mathcal{C}' = \{(\alpha_1 c_{\pi(1)}, \alpha_2 c_{\pi(2)}, \ldots, \alpha_n c_{\pi(n)}) \mid (c_1, c_2, \ldots, c_n) \in \mathcal{C}\}$$

We note that equivalent codes give rise to equivalent access structures.

In the classification, we will need the Griesmer bound. Let

$$g_q(k, d) = \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

For any $[n, k, d; q]$ code we have

$$n \geq g_q(k, d).$$

First we describe the *equidistant* codes, that are the $[n, k, d; q]$ codes where all nonzero codewords have weight d.

Equidistant codes

The *simplex codes* $\mathcal{S}_q(k)$ are the $\left[\frac{q^k-1}{q-1}, k, q^{k-1}; q\right]$ codes generated by the $k \times \frac{q^k-1}{q-1}$ matrices G whose columns are nonzero and non-proportional. Without loss of generality we can assume that the first nonzero component in each column vector is 1, in which case G contains all such vectors. A *zero-position* for a code is a position where all codewords are zero. If we repeat the simplex code $s \geq 1$ times and add some zero-positions, we still get an equidistant code. It is known that all equidistant codes are obtained in this way. For completeness, we include the short proof.

Lemma 4.1. *Let $\mathcal{C}$ be an $[n, k, d; q]$ code without zero-positions. Then $\mathcal{C}$ is equidistant if and only if $\mathcal{C}$ is an $\left[s\frac{q^k-1}{q-1}, k, sq^{k-1}\right]$ code for some $s \geq 1$.*

Proof: Let A be the $q^k \times n$ matrix containing all codewords as rows. Counting the number of nonzero elements in A row-wise and column-wise we get

$$(q^k - 1)d \leq \sum_{\mathbf{c} \in \mathcal{C}} w(\mathbf{c}) = (q-1)q^{k-1}n. \tag{1}$$

Suppose $\mathcal{C}$ is equidistant. Then we have equality in (1). Since

$$\gcd\left(\frac{q^k-1}{q-1}, q^{k-1}\right) = 1,$$

we have

$$n = s\frac{q^k-1}{q-1} \text{ and } d = sq^{k-1}$$

for some integer s.

Conversely, suppose $n = s\frac{q^k-1}{q-1}$ and $d = sq^{k-1}$ for some s. Then $(q^k-1)d = (q-1)q^{k-1}n$, and so we must have equality in (1), that is, $w(\mathbf{c}) = d$ for all nonzero $\mathbf{c}$. In other words, the code is equidistant. **Q.E.D.**

Lemma 4.2. *Any* $\left[s\frac{q^k-1}{q-1}, k, sq^{k-1}; q\right]$ *code* $\mathcal{C}$ *without zero-positions is equivalent to the code obtained by* s *repetitions of* $\mathcal{S}_q(k)$.

Proof: It is sufficient to show that no column occurs more than s times. Suppose one column occurs $u \geq s+1$ times. Then without loss of generality we may assume that the code have a generator matrix of the form

$$G = \begin{bmatrix} \overbrace{\begin{matrix} 1 & 1 & \cdots & 1 \end{matrix}}^{u} & \mathbf{x} \\ \begin{matrix} \mathbf{0}^T & \mathbf{0}^T & \cdots & \mathbf{0}^T \end{matrix} & G_0 \end{bmatrix},$$

where $\mathbf{x}$ is some vector of length $s\frac{q^k-1}{q-1} - u$, $\mathbf{0}$ is the all-zero vector, and G_0 generates an $\left[s\frac{q^k-1}{q-1} - u, k-1, sq^{k-1}; q\right]$ code. This contradicts the Griesmer bound, since

$$g(k-1, sq^{k-1}) = s\left(\frac{q^k-1}{q-1} - 1\right) > s\frac{q^k-1}{q-1} - u.$$

Q.E.D.

By Lemmas 3.4 and 4.2, the only equidistant codes of interest for our secret sharing problem are the simplex codes.

Two-weight $[n, k, d; q]$ codes containing codewords of weight n

We now go on to study two-weight codes which have codewords of weight n. Since secret-sharing schemes based on equivalent codes have the same access structure, we may assume without loss of generality that $\mathbf{1}$, the all one vector, is a codeword. We call such codes *self-complementary* codes. We note that any code obtained by $s \geq 2$ repetitions of a self-complementary two-weight code is again a self-complementary two-weight code. A self-complementary two-weight code which can not be obtained by repetitions of a shorter such code will be called *primitive*. We remark that if $\mathcal{C}$ is self-complementary, then clearly $\mathcal{C}$ can not have any zero positions.

An important class of primitive, self-complementary, two-weight codes are the *first order Reed-Muller codes*, $\mathcal{R}_q(1,m)$. The code $\mathcal{R}_q(1,m)$ is the $[q^m, m+1, (q-1)q^{m-1}; q]$ code with generator matrix

$$\begin{pmatrix} \mathbf{1} \\ U_m \end{pmatrix},$$

where the columns of U_m are exactly all the vectors in $GF(q)^m$. We will show that most primitive, two-weight, self-complementary codes are of this form. In particular this is the case if $k \geq 4$. However, for $k=2$ and $k=3$ there are also other such codes.

Any self-complementary two-weight $[n,k,d;q]$ code $\mathcal{C}$ has a generator matrix of the form

$$\begin{pmatrix} \mathbf{1} \\ V \end{pmatrix},$$

where the first column of V is all-zero. The code generated by V will be denoted by $\mathcal{C}_V$. The code $\mathcal{C}_V$ is a subcode of $\mathcal{C}$ and all codewords have weight less than n. Therefore $\mathcal{C}_V$ is an $[n, k-1, d; q]$ equidistant code.

Lemma 4.3. *Let $\mathcal{C}$ be a self-complementary two-weight $[n,k,d;q]$ code. Then $n-d$ divides n. Let $a_1 = a_1(\mathcal{C}) = n-d$ and $r_1 = r_1(\mathcal{C}) = n/(n-d)$. Each nonzero codeword in $\mathcal{C}_V$ contains exactly r_1 distinct elements from $GF(q)$, each of these elements occurs exactly a_1 times in the codeword.*

Proof: Let $\mathbf{c}$ be an nonzero codeword in $\mathcal{C}_V$, and let $\alpha \in GF(q)$ be an element that appears in $\mathbf{c}$. Suppose that it appears exactly $y > 0$ times. Then $\mathbf{c} - \alpha\mathbf{1} \in \mathcal{C}$ and $w(\mathbf{c} - \alpha\mathbf{1}) = n - y < n$, where $\mathbf{1}$ is the all-one codeword. Hence $n - y = d$, and so $y = n - d = a_1$. Since this is independent of α and $\mathbf{c}$, $\mathbf{c}$ must contain exactly $n/(n-d) = r_1$ elements. In particular, $n-d$ divides n. **Q.E.D.**

We next prove that if $r_1(\mathcal{C}) = q$, then $\mathcal{C}$ is equivalent to some copies of the Reed-Muller code.

Lemma 4.4. *Let $\mathcal{C}$ be a self-complementary two-weight $[n,k,d;q]$ code. If $r_1(\mathcal{C}) = q$, then $\mathcal{C}$ is equivalent to some copies of the Reed-Muller code.*

Proof: The code $\mathcal{C}_V$ is an equidistant code. Let t be the number of zero-positions in $\mathcal{C}_V$. By Lemmas 4.1 and 4.2 $\mathcal{C}_V$ is an $\left[s\frac{q^{k-1}-1}{q-1} + t, k-1, sq^{k-2}\right]$ code for some s. Hence Lemma 4.3 gives $n/(n-d) = q$ which implies that $n(q-1) = dq$. Therefore, $s = t(q-1)$ and it follows that t copies of each vector in $GF(q)^{k-1}$ appears as columns in V. Hence $\mathcal{C}$ is equivalent to t copies of $\mathcal{R}_q(1, k-1)$. **Q.E.D.**

Example 1: Consider the case $k = 2$. We note that if $\alpha_1, \alpha_2, \cdots, \alpha_r$ are distinct elements of $GF(q)$, then the code generated by $\mathbf{1}$ and $(\alpha_1, \alpha_2, \ldots, \alpha_r)$ is a primitive, self-complementary, two-weight $[r, 2, r-1; q]$ code. By Lemma 4.3, up to equivalence, these are the only primitive self-complementary, two-weight codes of dimension 2.

Lemma 4.5. *Let $\mathcal{C}$ be a self-complementary two-weight $[n, k, d; q]$ code where $k \geq 3$. Then q divides d. Let $a_2 = a_2(C) = a_1 - d/q$ and $r_2 = r_2(C) = a_1/a_2$. For a pair of nonzero linearly independent codewords $\mathbf{c}$, $\mathbf{c}'$ in $\mathcal{C}_V$, consider the a_1 positions $\{i \mid c_i = 0\}$. There are exactly r_2 distinct elements from $GF(q)$ in those positions and each of these elements occurs exactly a_2 times.*

Proof: Let z be the number of positions i such that $c_i = c'_i = 0$. Obviously $z > 0$ since the first coordinate of any codeword in $\mathcal{C}_V$ is zero.

Let $\alpha \in GF(q)$ and consider $\mathbf{c} + \alpha\mathbf{c}'$. Since $b_\alpha \stackrel{\text{def}}{=} |\{i : c_i + \alpha c'_i = 0\}| \geq z > 0$, then by Lemma 4.3 we have $b_\alpha = a_1$. Let $S = \sum_{\alpha \in GF(q)} b_\alpha = qa_1$. We can determine S in another way as follows.

- If $c_i \neq 0$ and $c'_i = 0$, i does not contribute to any b_α; there are $a_1 - z$ such i.
- If $c_i = 0$ and $c'_i = 0$, i contributes to all b_α, and there are z such i.
- If $c'_i \neq 0$ then i contributes to exactly one b_α; there are d such i.

Hence $zq + d = qa_1$; in particular, q divides d and $z = a_1 - d/q = a_2$. Let $\alpha \in GF(q)$ such that $c_i = 0$ and $c'_i = \alpha$ for some i. Applying the above argument to the pair $\mathbf{c}, \mathbf{c}' - \alpha\mathbf{1}$ we get $|\{i : c_i = 0,\ c'_i = \alpha\}| = a_2$. Hence, there are exactly $a_1/a_2 = r_2$ distinct elements $\alpha \in GF(q)$ such that $c_i = 0$ and $c'_i = \alpha$ for some i. **Q.E.D.**

Corollary 4.6. *Let $\mathcal{C}$ be a self-complementary two-weight $[n, k, d; q]$ code where $k \geq 3$. Then*

$$q = r_2(q - r_1 + 1). \tag{2}$$

Proof: We have

$$\begin{aligned} r_2(q - r_1 + 1) &= \frac{a_1}{a_1 - \frac{d}{q}} \left(q - \frac{n}{n-d} + 1 \right) \\ &= \frac{q}{qa_1 - d} \left(qa_1 - a_1 \frac{d}{n-d} \right) = q. \end{aligned}$$

Q.E.D.

Corollary 4.7. *Let $\mathcal{C}$ be a self-complementary two-weight $[n, k, d; q]$ code where $k \geq 3$ and q is a prime. Then $\mathcal{C}$ is equivalent to some copies of the Reed-Muller code $\mathcal{R}_q(1, k-1)$.*

Proof: We have $r_1 > 1$. Hence the only solution of (2) when q is a prime is $r_1 = r_2 = q$. The result now follows from Theorem 4.4. **Q.E.D.**

The idea of Lemma 4.5 can be repeated to show the following result.

Theorem 4.8. *Let $\mathcal{C}$ be a self-complementary two-weight $[n,k,d;q]$ code where $k \geq 4$. Then $\mathcal{C}$ is equivalent to some copies of the Reed-Muller code $\mathcal{R}_q(1,k-1)$.*

Proof: Let $\mathbf{c}$, $\mathbf{c}'$ and $\mathbf{c}''$ be three linearly independent vectors in $\mathcal{C}_V$. Let $u = |\{i : c_i = c_i' = c_i'' = 0\}|$. For each $\alpha \in GF(q)$, let

$$b_\alpha = |\{i : c_i'' = 0,\ c_i + \alpha c_i' = 0\}|\,.$$

By Lemma 4.5, $b_\alpha = a_2$, and so

$$\sum_{\alpha \in GF(q)} b_\alpha = q a_2.$$

Counting the terms in an alternative ways, as we did in the proof of Lemma 4.5, we get $qa_2 = qu + a_1 - a_2$, and so q divides $a_1 - a_2$, and $u = a_3 \stackrel{\text{def}}{=} a_2 - \frac{1}{q}(a_1 - a_2)$.

Similarly, $|\{i : c_i' = c_i'' = 0,\ c_i = \alpha\}|$ is a_3 or 0 for all $\alpha \in GF(q)$. Hence $a_3 r_3 = a_2$ for some integer r_3, and this can be rewritten as $q = r_3(q - r_2 + 1)$. Let $q = p^a$ where p is some prime. Since $r_1 > 1$, (2) implies that $r_2 = p^b$ for some integer $b > 1$. Hence $\gcd(p^a, p^a - p^b + 1) = 1$. Therefore $r_3 = q$. In turn this implies that $r_2 = q$, and by (2), $r_1 = q$. The theorem now follows from Lemma 4.4. **Q.E.D.**

We have now completely characterized the self-complementary two-weight $[n,k,d;q]$ codes, except when $k = 3$ and q is a prime power. We consider this situation next. Calderbank and Kantor [4] in their analysis of two-weight codes gave self-complementary two-weight $[(2^a+1)(2^b-1)+1, 3, 2^a(2^b-1); 2^a]$ codes for all a, b such that $1 \leq b < a$. For $b = 1$ these codes are MDS codes.

As an example, we consider the case $k = 3$ and $q = 4$. Let $GF(4) = \{0, 1, \alpha, \beta\}$, where $\beta = \alpha^2 = \alpha + 1$. For $b = 1$ we get an $[6,3,4;4]$ code. We have $r_1 = 3$, $a_1 = 2$, $r_2 = 2$, $a_2 = 1$. The code generated by the matrix

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & \alpha & \alpha \\ 0 & 1 & 1 & \beta & \beta & 0 \end{bmatrix}$$

is such an $[6,3,4]$ MDS code.

Any self-complementary two-weight $[n,k,d;q]$ code $\mathcal{C}$ with $k \geq 4$ is an $[sq^m, 1+m, s(q-1)q^{m-1}; q]$ code consisting of s copies of the first-order Reed-Muller code. Thus, if $s \geq 2$, the minimum distance of $\mathcal{C}^\perp$ is 2. By Lemma 3.4 the supports of all minimum weight codewords of $\mathcal{C}$ can not form a 2-design. Thus, the only self-complementary two-weight codes which are interesting for our secret sharing are the $[r, 2, r-1; q]$ code described in Example 1 and some $[p^m, 3, d; p^l]$ codes.

In the following two sections we describe the access structure of the secret-sharing schemes based on the dual code of the first-order Reed-Muller codes $\mathcal{R}_q(1,m)$ and the Hamming codes $\mathcal{H}_q(m)$.

5 With Reed-Muller codes

The access structure of the secret-sharing scheme based on $\mathcal{R}_q(1,m)^\perp$ is described by the following theorem.

Theorem 5.1. *The secret-sharing scheme based on $\mathcal{R}_q(1,m)^\perp$ is for the sharing of secrets among q^m-1 parties with the following properties:*

1. *there are q^m-1 minimum access sets consisting of $(q-1)q^{m-1}-1$ participants;*
2. *each participant is a member of exactly $(q-1)q^{m-1}-1$ minimum access sets.*

Proof: By definition any two columns of a generator matrix of $\mathcal{R}_q(1,m)$ are linearly independent. It follows from Lemma 3.4 that the supports of the minimum weight codewords of $\mathcal{R}_q(1,m)$ form a 2-$(q^m,(q-1)q^{m-1},\lambda)$ design. By Lemma 3.1 we obtain

$$\lambda = (q-1)q^{m-1} - 1.$$

By Lemma 3.2 the minimal access sets form a 1-$(q^m-1,(q-1)q^{m-1}-1,\lambda)$ design. Again by Lemma 3.1 we see that the number of minimum access sets is q^m-1. Since the code $\mathcal{R}_q(1,m)$ has minimum distance $(q-1)q^{m-1}$, each minimum access set consists of $(q-1)q^{m-1}-1$ parties. **Q.E.D.**

Example 2: Consider the binary case. It is not hard to see that the minimal access sets of the secret-sharing scheme based on the code $\mathcal{R}_2(1,4)^\perp$ are

$$\begin{array}{lll}
\{1,2,3,4,5,6,7\} & \{1,2,3,8,9,10,11\} & \{1,4,5,8,9,12,13\} \\
\{2,4,6,8,10,12,14\} & \{1,2,3,12,13,14,15\} & \{1,4,5,10,11,14,15\} \\
\{2,4,6,9,11,13,15\} & \{1,6,7,8,9,14,15\} & \{2,5,7,8,10,13,15\} \\
\{3,4,7,8,11,12,15\} & \{3,5,6,8,11,13,14\} & \{3,4,7,9,10,13,14\} \\
\{2,5,7,9,11,12,14\} & \{1,6,7,10,11,12,13\} & \{3,5,6,9,10,12,15\}.
\end{array}$$

In this example the minimal access sets form a 2-design, but this seems not to be true in general.

6 With Hamming codes

Since the only equidistant codes such that the supports of all minimum weight codewords form a 2-design are the simplex codes, we describe the access structure and properties of the secret-sharing scheme based on Hamming codes.

Theorem 6.1. *The scheme based on the Hamming code $\mathcal{H}_q(m)$ is for sharing secrets among $q(q^{m-1}-1)/(q-1)$ parties has the following properties:*

1. *there are q^{m-1} minimum access sets consisting of $q^{m-1}-1$ participants;*
2. *each participant is a member of exactly $(q-1)q^{m-2}$ minimum access sets.*

Proof: Note that any two columns of a generator matrix of $\mathcal{S}_q(m)$ are linearly independent. The proof is similar to that of Theorem 5.1. **Q.E.D.**

Example 3: The minimal access sets of the secret-sharing scheme based on the binary $[7, 3, 4]$ Hamming code are

$$\{1,2,4\}, \{3,4,5\}, \{2,5,6\}, \{1,3,6\}$$

which form a 1-$(6, 3, 2)$ design. This is the access structure discussed in the first section.

By now the reader should be able to design schemes to suit his application in a systematic way. We leave as an exercise the development of shared signature schemes similar to Desmedt-Frankel but which express access structures of the type described here.

Acknowledgement: This research was carried out when the authors were visiting the Isaac Newton Institute for Mathematical Sciences, Cambridge, UK.

References

1. RJ Anderson, "Why Cryptosystems Fail" in *Communications of the ACM* vol 37 no 11 (November 1994) pp 32–40
2. EF Assmus, JD Key, *Designs and Their Codes*, Cambridge: Cambridge University Press, 1992
3. GR Blakley, "Safeguarding cryptographic keys", in *Proc. NCC AFIPS 1979*, pp 313–317
4. R Calderbank, WM Kantor, *The geometry of two-weight codes*, in *Bulletin of the London Mathematical Society* vol 18 (1986) pp 97–122
5. Y Desmedt, Y Frankel, "Threshold cryptosystems", in *Advances in Cryptology — Crypto 89*, Springer LNCS vol 435, pp 307–315
6. ED Karnin, JW Green, M Hellman, "On secret sharing systems", in *IEEE Transactions on Information Theory* vol IT-29 (1983) pp 644–654
7. FJ MacWilliams, NJA Sloane, *The Theory of Error-Correcting Codes,* North-Holland, 1978
8. JL Massey, "Minimal codewords and secret sharing", in *Proceedings of the 6th Joint Swedish-Russian Workshop on Information Theory*, Mölle, Sweden, August 22-27, 1993, pp 276–279
9. JL Massey, "Some applications of coding theory in cryptography", in *Codes and Ciphers: Cryptography and Coding IV* (Ed. PG Farrell), IMA, England (1995) pp 33–47
10. RJ McEliece, DV Sarwate, "On sharing secrets and Reed-Solomon codes", in *Communications of the ACM* vol 24 (1981) pp 583–584
11. MK Reiter, MK Franklin, JB Lacy, RA Wright, "The Omega Key Management Service", in *3rd ACM Conference on Computer and Communications Security*, March 14-16, 96, New Delhi, India; proceedings published by ACM (ISBN 0-89791-829-0) pp 38–47
12. A Shamir, "How to share a secret", *Communications of the ACM* vol 22 (1979) pp 612–613
13. *'VISA Security Module Operations Manual'*, VISA, 1986