Further Results on Strongbox Secured Secret Sharing Schemes

Greg Gamble[*]

School of Mathematics and Statistics, Curtin University of Technology

Kent Street, Bentley WA 6102, Australia

gregg@math.rwth-aachen.de

Barbara M Maenhaut[†]

Centre for Discrete Mathematics and Computing

The University of Queensland, Brisbane 4072, Australia

bmm@maths.uq.edu.au

Jennifer Seberry[‡]

Centre for Computer Security Research

University of Wollongong, NSW 2522, Australia

j.seberry@uow.edu.au

Anne Penfold Street[‡]

Centre for Discrete Mathematics and Computing

The University of Queensland, Brisbane 4072, Australia

aps@maths.uq.edu.au

Abstract

We extend our earlier work on ways in which defining sets of com-
binatorial designs can be used to create secret sharing schemes. We
give an algorithm for classifying defining sets of designs according to
their security properties and summarise the results of this algorithm
for many small designs. Finally, we discuss briefly how defining sets
can be applied to variations of the basic secret sharing scheme.

Key words and phrases: Combinatorial designs, defining sets, information
content, influence, power, nest, strongbox, secret sharing, access schemes.

1

1 Introduction

Schemes in which a dealer distributes partial information (shares) to a group of participants in such a way that only pre-designated collections of participants are able to re-create the password (secret or key) are called **access schemes** or **secret sharing schemes**. Access schemes are used in financial institutions, communications networks, computer systems and the military; see [2, 15]. In [20], Seberry and Street introduced a way to use combinatorial designs to create a conditionally perfect access scheme. In this paper, we extend the ideas that were presented in [20].

A t-(v, k, λ) **design** is a set of subsets of size k (blocks) chosen from a set of v elements in such a way that every set of t elements occurs in exactly λ blocks. A 2-(v, k, λ) design is often referred to as a (v, k, λ) design.

A 2-$(7, 3, 1)$ design based on the set $\mathcal{X} = \{1, 2, 3, 4, 5, 6, 7\}$ is given by the set of blocks

$$\mathcal{B} = \{\mathbf{1} : 124, \ \mathbf{2} : 235, \ \mathbf{3} : 346, \ \mathbf{4} : 457, \ \mathbf{5} : 561, \ \mathbf{6} : 672, \ \mathbf{7} : 713\}$$

where the boldface numbers label the blocks, and the elements of each block are listed without commas or brackets. We write $\mathcal{F}$ to denote this $(7, 3, 1)$ design, or more explicitly $\mathcal{F} = (\mathcal{X}, \mathcal{B})$. In this paper we are concerned both with **simple** designs, that is, those in which no block is repeated, and with **non-simple** designs.

We now consider certain subsets of blocks of the design $\mathcal{F}$. The three blocks $\mathbf{1}, \mathbf{5}, \mathbf{7}$ can be completed to a $(7, 3, 1)$ design in two different ways: by adjoining the blocks $\mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{6}$ of the original design, or by adjoining the blocks $257, 236, 345, 467$ to give another $(7, 3, 1)$ design. Similarly any set of three blocks from $\mathcal{F}$ which contain a common point can be completed to two different $(7, 3, 1)$ designs. However, any set of three blocks which do not contain a common point have the property that they complete to a unique $(7, 3, 1)$ design. Thus the set $\mathcal{S}$ consisting of the blocks $\mathbf{1}, \mathbf{2}, \mathbf{3}$ must lead to a $(7, 3, 1)$ design containing the blocks $\mathbf{6}$ (since 2 must appear with 6 and 7), $\mathbf{4}$ (since 4 must appear with 5 and 7), and so on, giving our original design $\mathcal{F}$.

Since the set $\mathcal{S}$ has a unique completion to a $(7, 3, 1)$ design we call it a **defining set** of the design. Since no set of one or two blocks forces a unique completion to $\mathcal{F}$, $\mathcal{S}$ is a **smallest** defining set of the design. A **minimal**

defining set is a defining set which does not properly contain any other defining set. In this design every minimal defining set is in fact a smallest defining set, but in many designs there are sets of blocks which are minimal but not smallest defining sets; these will be referred to as *other* minimal defining sets. The notion of a defining set was introduced by Curtis [3] for the 5-$(24, 8, 1)$ design, but the formal definitions were introduced in 1990 by K.Gray, in a series of three papers [9, 10, 11]. He proved several theoretical results on defining sets as well as finding bounds on the size of minimal defining sets.

Closely related to the concept of a defining set is that of a trade in a design. Suppose that a design $\mathcal{D}$ contains a set $\mathcal{T}_1$ of blocks, and that this set has the following property: there exists another set of blocks $\mathcal{T}_2$, with $\mathcal{T}_1 \cap \mathcal{T}_2 = \emptyset$, such that $(\mathcal{D} \setminus \mathcal{T}_1) \cup \mathcal{T}_2$ is another design with the same parameters as $\mathcal{D}$. In this case, the pair $(\mathcal{T}_1, \mathcal{T}_2)$ is said to be a **trade**. The collection $\mathcal{T}_2$ is said to be a **trade mate** of $\mathcal{T}_1$. Note that when a particular design $\mathcal{D}$ is being considered, a single collection of blocks $\mathcal{T}_1$ is often referred to as a trade, with the understanding that a trade mate $\mathcal{T}_2$ exists. However, this convention may lead to confusion in the study of the enumeration of trades. An example of a trade in our $(7, 3, 1)$ design $\mathcal{F}$ above is the set $\mathcal{T}_1$ consisting of the blocks **2**, **3**, **4**, **6** of $\mathcal{F}$, with trade mate $\mathcal{T}_2 = \{257, 236, 345, 467\}$. Clearly any defining set must contain at least one block from each trade in a design.

Defining sets of designs can be used to create an access scheme in the following way. The dealer chooses a t-(v, k, λ) design. Note that the parameters t, v, k and λ might be public knowledge. The dealer then finds a minimal defining set $\mathcal{S}$ for that design. The dealer partitions the set of blocks of $\mathcal{S}$ into p shares and distributes the shares among the p participants. When these p participants combine their shares, they can reconstruct the minimal defining set $\mathcal{S}$ and hence find the design. The key will be located in the strongbox of the design, which we define in the next section.

2 Access schemes using defining sets

In [20], the ideas of Fitina [7] were adapted, giving rise to the definitions of the *nest*, *power* and *influence* of a block within a defining set, as well as the

strongbox of a defining set. These concepts provide some measure of how secure the defining set access scheme is. Here we review these definitions, illustrating them with the design $\mathcal{F}$ listed in the previous section, and its minimal defining set $\mathcal{S}_{\mathcal{F}} = \{124, 235, 346\}$.

Let $\mathcal{D}$ be a t-(v, k, λ) design with minimal defining set $\mathcal{S}$ and let B be a block of $\mathcal{S}$. The **nest** of B in $\mathcal{S}$, $\mathcal{N}(\mathcal{S}, B)$, is the set of blocks of $\mathcal{S} \setminus \{B\}$, together with all the complete and partial blocks forced by the information in $\mathcal{S} \setminus \{B\}$ and the parameters of the design. More precisely, we write

$$\mathcal{N}(\mathcal{S}, B) = (\mathcal{S} \setminus \{B\}) \cup N'(\mathcal{S}, B) \cup N''(\mathcal{S}, B)$$

where $N'(\mathcal{S}, B)$ is the collection of complete blocks forced by the information in $\mathcal{S} \setminus \{B\}$ and $N''(\mathcal{S}, B)$ is the collection of partial blocks of size at least $t + 1$ forced by the information in $\mathcal{S} \setminus \{B\}$. The nest of the block 124 in $\mathcal{S}_{\mathcal{F}}$ is

$$\mathcal{N}(\mathcal{S}_{\mathcal{F}}, 124) = \{235, 346, 137\}.$$

Here $N'(\mathcal{S}_{\mathcal{F}}, B)$ consists of the single block 137 and the set $N''(\mathcal{S}_{\mathcal{F}}, B)$ is empty since there are no partial blocks of size greater than $t = 2$.

The **power** of B in $\mathcal{S}$, $\mathcal{P}(\mathcal{S}, B)$, is the number of completions of $\mathcal{S} \setminus \{B\}$ to a design with the same parameters as $\mathcal{D}$. In the design $\mathcal{F}$ with defining set $\mathcal{S}_{\mathcal{F}}$, $\mathcal{P}(\mathcal{S}_{\mathcal{F}}, 124) = 2$, since the set of blocks $\{235, 346\}$ can be completed to a $(7, 3, 1)$ design by adding either the blocks

$$\{137, 124, 156, 267, 457\},$$

or the blocks

$$\{137, 126, 145, 247, 567\}.$$

The **influence** of B in $\mathcal{S}$, $\mathcal{I}(\mathcal{S}, B)$, is the number of complete blocks of $\mathcal{D}$ which are not entirely determined by the information in $\mathcal{S} \setminus \{B\}$. More precisely, $\mathcal{I}(\mathcal{S}, B) = |\mathcal{D} \setminus (\mathcal{S} \setminus \{B\})| - |N'(\mathcal{S}, B)|$. The design $\mathcal{F}$ contains seven blocks and the nest of 124 contains three complete blocks, so the influence of the block 124 in $\mathcal{S}_{\mathcal{F}}$ is four.

The **strongbox** of a set of blocks $\mathcal{S}$ is the set of complete blocks of the design $\mathcal{D}$ which are not contained in the nest of any block of $\mathcal{S}$. More precisely it is the set

$$(\mathcal{D} \setminus \mathcal{S}) \setminus \bigcup_{B_i \in \mathcal{S}} N'(\mathcal{S}, B_i).$$

4

Consider the defining set $\mathcal{S}_{\mathcal{F}}$ of the $(7, 3, 1)$ design $\mathcal{F}$. The nest of 124 contains the three complete blocks:

$$235, 346, 137.$$

The nest of 235 contains the three complete blocks:

$$124, 346, 457.$$

The nest of 346 contains the three complete blocks:

$$124, 235, 267.$$

Taking the union of the nests gives only six complete blocks, so there is one complete block of $\mathcal{F}$ not contained in any of them. The strongbox of the defining set $\mathcal{S}_{\mathcal{F}}$ is this single block 156.

These definitions give us a way of measuring how secure our defining set access scheme is. If, in our defining set access scheme, each participant receives a share consisting of one block, then the nest of each block gives an indication of how much information a maximum size unauthorized subset of participants could calculate about the design, and the power of the block gives the number of possibilities they would have to consider if they were to guess the design. The influence of a block gives an indication of how much of the design would remain secret if that participant was missing. The strongbox of a defining set gives an indication of how much of the design would remain secret, regardless of which unauthorized subsets of participants collaborated in an attempt to cheat. In our access schemes we would like the power of each block to be quite large. We would also like the influence of each block, B, to be close to the size of $\mathcal{D} \setminus (\mathcal{S} \setminus \{B\})$, and the size of the strongbox to be large.

We say that a block B, in the defining set $\mathcal{S}$ of design $\mathcal{D}$, has **perfect influence** if the influence of B is the number of blocks in $\mathcal{D} \setminus (\mathcal{S} \setminus \{B\})$. That is, if no complete blocks outside the defining set can be determined without knowledge of B.

Example: Consider the 2-$(15, 7, 3)$ design, $PG(3, 2)$, based on the underlying set of 15 elements $0, 1, \ldots, 9, \mathtt{a}, \ldots, \mathtt{e}$, and developed from the difference set $B_0 = \mathtt{012458a}$ (mod 15). Then block B_i of the design is $B_0 + i$, (mod 15). Here the design has a smallest defining set $\mathcal{S}$ of size nine [12], consisting of the blocks B_i, for $i = 0, 1, 2, 3, 4, 5, 6, 7, 9$.

For each block B_i in the defining set, there are three completions of $\mathcal{S} \setminus \{B_i\}$, so the power of each block is three. The only complete blocks in the nest of $\mathcal{B}_i$ are the blocks in $\mathcal{S} \setminus \{B_i\}$, so each block has perfect influence. The strongbox of this defining set has size six, being the set of blocks in $\mathcal{D} \setminus \mathcal{S}$.

3   Keys and pointwise defining sets

There are many possibilities for the key of such a defining set based access scheme. Clearly, it would be advantageous in this type of scheme to have the key be dependent on the strongbox of the defining set. One possibility is to write each block of the strongbox in lexicographic order, create an integer from a string of k integers $a_1 a_2 \ldots a_k$ where a_i is the sum of the ith elements of the blocks, and then subtract this integer from a suitably large number. The difference would then be the key of the access scheme. Applying this idea to the example above, based on the design $PG(3, 2)$, and choosing the suitably large number to be 9876543210123 gives a key of 7753700612941. The blocks in the strongbox written in lexicographic order, with the letters $\mathtt{a} = 10$, $\mathtt{b} = 11$, $\mathtt{c} = 12$, $\mathtt{d} = 13$ and $\mathtt{e} = 14$ are

$$
\begin{array}{ccccccc}
1 & 3 & 8 & 9 & 10 & 12 & 13 \\
0 & 3 & 5 & 10 & 11 & 12 & 14 \\
0 & 1 & 4 & 6 & 11 & 12 & 13 \\
1 & 2 & 5 & 7 & 12 & 13 & 14 \\
0 & 2 & 3 & 6 & 8 & 13 & 14 \\
0 & 1 & 3 & 4 & 7 & 9 & 14.
\end{array}
$$

Summing the columns of this array and concatenating the sums to form an integer gives 2122842597182. Subtracting this number from 9876543210123 gives the key of 7753700612941. Of course, there are many other ways of hiding the key in the strongbox of a defining set.

The definitions of nest, power, influence and strongbox can be applied to access schemes involving defining sets of block designs, provided that the defining sets involve only complete blocks. Our work on defining set access schemes leads us to wonder whether **pointwise** defining sets, which contain partial and/or complete blocks, may be more useful in certain access

schemes than blockwise defining sets. The following example illustrates one of the extra complexities that arise when considering pointwise defining sets.

Example: Consider the 2-$(11, 5, 2)$ design generated by the block `13459` cycled modulo 11, which has two non-isomorphic smallest defining sets $\mathcal{S}_1$ and $\mathcal{S}_2$ where

$$\mathcal{S}_1 = \{13459, 2456a, 35670, 79a04, 90126\}.$$
$$\mathcal{S}_2 = \{13459, 2456a, 35670, 46781, 57892\}.$$

Each of these has a strongbox consisting of just one block: `a1237` for $\mathcal{S}_1$ and `79a04` for $\mathcal{S}_2$. If two people are to share access to the secret, one could be given three blocks and the other the remaining two.

However if we attempt to share access to a secret hidden in this design by giving two people portions of each block instead of complete blocks, we may run into problems, since we may inadvertently give one participant a pointwise defining set. The defining set $\mathcal{S}_1$ has eight non-isomorphic pointwise defining sets contained within its blocks, and $\mathcal{S}_2$ has 26; see [5]. For instance, the following five 4-element subsets of the blocks of $\mathcal{S}_2$ form a defining set:

$$\mathcal{S}_2 = \{3459, 456a, 3567, 6781, 7892\}.$$

In these two examples the secret is hidden in the strongbox. Hence the key can be the block in the strongbox or some function thereof. By definition the strongbox has the property that no $n - 1$ of the n sets in the smallest defining set can recreate any of the information in the strongbox, so we have an n out of n secret sharing scheme. In the case where each share consists of one or more blocks and the secret is one block, the information content of the secret is less than or equal to the information content of each share.

We plan to adapt the definitions of nest, power, influence and strongbox to take into account pointwise defining sets and to study their applications in access schemes in a subsequent paper.

4   Strongboxes

Our study of strongboxes revealed another more surprising possibility. Consider the 2-$(16, 4, 1)$ design listed as P_1 in Appendix B. There are four in-

equivalent defining sets, with strongboxes of sizes 2, 7, 8, 8 respectively. Each smallest defining set has seven blocks.

Again the secret is hidden in the strongbox, so the key can be the blocks in the strongbox or some function thereof. Since each strongbox has $s > 1$ blocks, if each share is one block and the secret is $s > 1$ blocks, then the information content of the secret is greater than the information content of each share: each share has one block, each secret is a function of two or more blocks. Here we compare information content by considering the number of binary bits required to represent the information.

Furthermore in two cases the information content in the strongbox is larger (8 blocks) than the total information content of the shares (7 blocks).

We obtained some surprising results, as follows:

- for some designs the strongbox is the complement of the smallest (or other minimal) defining set of size s, and for some it is nearly the complement, that is, it misses by one block being the complement;

- for these designs with complementary or near-complementary strongboxes, the total information required to reconstruct the whole design is less than the information contained in the strongbox;

- for these designs, the information size of the secret, which can be a suitable function of the blocks in the strongbox, can be greater than the total information distributed in the shares.

We suspect that any 'missing' information is in fact held in the rules for constructing the design.

In an effort to determine which types of designs give the most secure defining set access schemes, we have developed a program for calculating these measures and have investigated many designs. The program and results are summarised in the next section and in the appendices; for details see also [14].

5   Calculating nests and strongboxes

In this section we give a brief outline of the algorithms used to calculate the nests, powers, influences and strongboxes of defining sets of designs. Given

a design $\mathcal{D}$ and a minimal defining set $\mathcal{S}$ of $\mathcal{D}$, these algorithms calculate the nest, power and influence of each block in $\mathcal{S}$ and the strongbox of $\mathcal{S}$. Automorphism information about the set $\mathcal{S}$ is also generated.

Every defining set must intersect every trade within a design in at least one block, and the automorphism group of a defining set is a subgroup of the automorphism group of the design; see K.Gray [9, 10, 11]. These properties have been essential in the development of algorithms for finding minimal defining sets; see Greenhill [13] and Delaney [6]. In developing fast algorithms that find all completions of a partial design to a design with specified parameters, block intersection patterns have also been important, especially linkage; see Ramsay [19], Utami [21] and Lawrence [16].

The nest calculating program uses the computer algebra system MAGMA [1], and interfaces with a version of the design completion program *cad*, originally part of the *complete* package written and described by Delaney [4] but later modified by Ramsay [18]. Automorphism information is calculated using the *AutomorphismGroup* function provided by MAGMA, which in turn makes use of the program *nauty* [17].

Note that in these programs designs are allowed to be non-simple and so design block-sets are multisets; set differences of multisets where they occur respect these multiplicities.

Given a design $\mathcal{D}$ and a minimal defining set $\mathcal{S}$, the program deletes each block B of $\mathcal{S}$ in turn. It stores all the completions of $\mathcal{S} \setminus \{B\}$ to a design with the parameters of $\mathcal{D}$ and thus the power of B in $\mathcal{S}$ is calculated. By identifying the complete blocks which are common to all the completions of $\mathcal{S} \setminus \{B\}$, the set $N'(\mathcal{S}, B)$ is computed. This allows the calculation of the influence of B in $\mathcal{S}$. The partial blocks in the nest of B are found by considering the partial blocks of size i in decreasing order of i from $i = k-1$ to $i = t + 1$. At each stage the partial blocks of size i which are common to all the completions of $\mathcal{S} \setminus \{B\}$ and which have not occurred in a partial block of larger size are added to the set $N''(\mathcal{S}, B)$. (Note that the program also considers the partial blocks of size t, but these are not really of interest in the calculation of the nest.) These calculations are carried out for each block of $\mathcal{S}$. At the very beginning of the program a set A is initialized to be the set of blocks in $\mathcal{D} \setminus \mathcal{S}$. As each block B is dealt with, the set $N'(\mathcal{S}, B)$ is used to remove blocks from A so that once all the blocks of $\mathcal{S}$ have been considered, the set A is the strongbox of $\mathcal{S}$.

9

Thus this program returns the power, nest and influence of each block in a given defining set, as well as the strongbox for the defining set as a whole. A detailed account of the program is given in Appendix A and the results of this program for several designs are summarised in Appendix B.

6   Choosing the design and defining set

As usual, we write b for the number of blocks in the design $\mathcal{D}$ and s for the size of a defining set $\mathcal{S}$ of $\mathcal{D}$. Clearly, in choosing a design $\mathcal{D}$ and a defining set $\mathcal{S}$ for use in this type of access scheme, we would like the powers of the blocks of $\mathcal{S}$ to be quite large, the influence of each block of $\mathcal{S}$ to be close to $|\mathcal{D} \setminus (\mathcal{S} \setminus \{B\})|$ and the size of the strongbox to be large. For a design $\mathcal{D}$ and a defining set $\mathcal{S}$ for which the size of the strongbox is $|\mathcal{D} \setminus \mathcal{S}| = b - s$, we say that the defining set has a **complementary** strongbox. Similarly if the size of the strongbox is $|\mathcal{D} \setminus \mathcal{S}| - 1 = b - s - 1$. we say the defining set has a **near-complementary** strongbox. From the information about power and influence of each block of $\mathcal{S}$, we calculate two extra measures. The **average strength**, AS, of a defining set $\mathcal{S}$ is the sum of the powers and influences of the blocks of $\mathcal{S}$ divided by s. The **average influence**, AI, of a defining set $\mathcal{S}$ is the sum of the influences of the blocks of $\mathcal{S}$ divided by s.

We applied the algorithms described in the previous section to many small designs, and the results are summarised in Appendix B. We noticed the following trends.

1. If $AS \geq b$, then the strongbox is unlikely to be empty.

2. If $AS \leq \frac{b}{2}$ and $AI \leq \frac{b}{2}$, then the strongbox is likely to be very small or empty.

3. If $AI = |D \setminus (S \setminus B)| = b - s + 1$, then there is a large strongbox.

4. If $AI < \frac{2b}{5}$, then the strongbox is likely to be empty.

7 Applications of defining set access schemes

7.1 Key Management

A participant involved in a number of secret sharing schemes may find it hard to memorize a different share for each scheme, hence reducing the security of each scheme. In this case we would like a **key management system** in which *either* the key and some shares are common to a number of secret sharing schemes *or* we have a collection of secret sharing schemes which have different keys but still have one or more shares in common. In such a system, the shares are chosen so that the primary share(s), necessary for the reconstruction process of each scheme, is held by the common participant(s).

Example: Suppose we require a pair of access schemes with three common participants. We could use a pair of 2-$(15, 3, 1)$ designs with a common 2-$(7, 3, 1)$ subsystem. Let D_1 be the 2-$(15, 3, 1)$ design with blocks

$123, 145, 167, 246, 257, 347, 356;$
$189, 1ab, 1cd, 1ef; 28a, 29b, 2ce, 2df; 38b, 39c, 3af, 3de; 48d, 49a, 4be, 4cf;$
$58f, 59e, 5ad, 5bc; 68e, 69f, 6ac, 6bd; 78c, 79d, 7ae, 7bf;$

and let D_2 be the $2 - (15, 3, 1)$ design with blocks

$123, 145, 167, 246, 257, 347, 356;$
$289, 2ab, 2cd, 2ef; 48a, 49b, 4ce, 4df; 68b, 69c, 6af, 6de; 38d, 39a, 3be, 3cf;$
$18f, 19e, 1ad, 1bc; 78e, 79f, 7ac, 7bd; 58c, 59d, 5ae, 5bf.$

The design D_1 has defining set

$123, 167, 257;$
$189, 1cd, 28a, 4be, 4cf, 59e, 6ac, 6bd, 7bf;$

and the design D_2 has defining set

$123, 167, 257;$
$289, 2cd, 48a, 3be, 3cf, 19e, 7ac, 7bd, 5bf.$

Thus the three common participants can be given one block each from the set $\{123, 167, 257\}$ and the remaining participants in each separate scheme can be given blocks from the appropriate defining set.

11

7.2 Multilevel schemes

In a company with a hierarchical structure, the value of a person's share in an access scheme is often required to reflect their clout in the company. For example, in an access scheme the shares of two junior officers might be able to replace the share of a senior officer, in case the senior officer was unavailable at the appropriate time. These situations call for a **multilevel scheme**. The shares given to s individuals of rank $1, \ldots, r$ are such that if a person of rank i is incapacitated, then the share of a person of rank $j \geq i$ or the shares of a set of individuals of rank $l \leq i$ may replace the missing share.

Example: Suppose we have a data file that may be accessed by two vice-presidents of a company. If one of the vice-presidents is absent, then we need two senior officers to be able to replace that share. The dealer could choose the $(9, 3, 1)$ design D_1:

$$
\begin{array}{cccc}
123, & 147, & 159, & 168 \\
456, & 258, & 267, & 249 \\
789, & 369, & 348, & 357
\end{array}
$$

with $\{123, 456, 147, 258\}$ as a defining set, for use in the secret sharing scheme. The two vice-presidents could be given the shares $\{123, 456\}$ and $\{147, 258\}$ respectively. Other minimal defining sets of this design include $\{123, 456, 369, 348, 249\}$ and $\{147, 258, 789, 348, 249\}$, so we could give four senior officers the shares $\{369\}$, $\{789\}$, $\{249\}$ and $\{348\}$ respectively. If the first of the vice-presidents were absent she could be replaced by senior officers 2, 3 and 4; if the second of the vice-presidents were absent he could be replaced by senior officers 1, 3 and 4.

8 Conclusion and open problems

Clearly the proposed access scheme based on defining sets is not perfect. The security of the scheme depends on the total number of block designs having the chosen parameters as well as the number of block designs containing the pooled information of any unauthorized subset of participants. However, we believe that the number of designs of order n is large enough to make it feasible to use their defining sets in access schemes.

8.1 Open problems

There are many areas which require further investigation in order to determine the feasibility of the suggested defining set access scheme. These areas include, but are not restricted to, the following problems:

- Determining whether pointwise defining sets, containing partial blocks, could be more useful than blockwise defining sets in the creation of hierarchical access schemes;

- Refinement of the definitions of influence and strongbox to measure the amount of information gained from partial blocks as well as complete blocks.

References

[1] W Bosma, J J Cannon and C Playoust, *The Magma algebra system I: The user language*, Journal of Symbolic Comput., **24** (1997) 235–265, http://www.maths.usyd.edu.au:8000/u/magma/.

[2] G Chaudhry, H Ghodosi and J Seberry, *Perfect secret sharing schemes from Room squares,* Journal of Combinatorial Mathematics and Combinatorial Computing, 28 (1998), 55-61.

[3] R T Curtis, *Eight octads suffice*, Journal of Combinatorial Theory (Series A), **36** (1984) 116–123.

[4] Cathy Delaney, *complete* – Rationale and User's Guide, CCRR-01-95, Centre for Combinatorics, Department of Mathematics, The University of Queensland, Brisbane, Australia, 1995.

[5] Cathy Delaney, Brenton D Gray, Ken Gray, Barbara Maenhaut, Martin J Sharry and Anne Penfold Street, *Pointwise Defining Sets and Trade Cores*, Australasian Journal of Combinatorics, **16** (1997) 51–76.

[6] Cathy Delaney, Martin J Sharry and Anne Penfold Street, *bds* – Rationale and User's Guide, CCRR-02-96, Centre for Combinatorics, Department of Mathematics, The University of Queensland, Brisbane, Australia, 1996.

[7] L F Fitina, Jennifer Seberry and Ghulam R Chaudhry, *Back circulant Latin squares and the influence of a set*, Australasian Journal of Combinatorics, **20** (1999) 163–180.

[8] Brenton D Gray and Colin Ramsay, *Some results on defining sets of t-designs*, Bulletin of the Australian Mathematical Society, **59** (1999) 203–215.

[9] Ken Gray, *On the minimum number of blocks defining a design,* Bulletin of the Australian Mathematical Society, **41** (1990) 97–112.

[10] Ken Gray, *Further results on smallest defining sets of well-known designs,* Australasian Journal of Combinatorics, **1** (1990) 91–100.

[11] Ken Gray, *Defining sets of single-transposition-free designs,* Utilitas Mathematica, **38** (1990) 97–103.

[12] Ken Gray and Anne Penfold Street, *The smallest defining set of the 2-(15, 7, 3) design associated with $PG(3, 2)$: a theoretical approach,* Bulletin of the Institute of Combinatorics and its Applications, **11** (1994) 77–83.

[13] Catherine S Greenhill, *An algorithm for finding smallest defining sets of t–designs,* Journal of Combinatorial Mathematics and Combinatorial Computing, **14** (1993) 39–60.

[14] Greg Gamble, *(Mostly) nests of designs,*
http://www.itee.uq.edu.au/~gregg/4Anne/home.html

[15] G Hossein, J Pieprzyk, G R Chaudhry and J Seberry, *How to prevent cheating in Pinch's scheme*, Electronic Letters, **33** (17) (1997), 1453-1454.

[16] Julie L Lawrence, personal communication to Anne Penfold Street, 1999.

[17] Brendan D McKay, *nauty - User's Guide (Version 1.5)*, TR-CS-90-02, Department of Computer Science, Australian National University, Canberra, Australia, 1990.

[18] Colin Ramsay, *An improved version of* `complete` *for the case* $\lambda = 1$, CCRR-03-96, Centre for Combinatorics, Department of Mathematics, The University of Queensland, Brisbane, Australia, 1996.

[19] Colin Ramsay, *An algorithm for completing partials, with an application to the smallest defining sets of the STS(15)*, Utilitas Mathematica, **52** (1997) 205–221.

[20] Jennifer Seberry and Anne Penfold Street, *Strongbox Secured Secret Sharing Schemes*, Utilitas Mathematica, **57** (2000) 147–163.

[21] Juliati Utami, *Algebraic Completions of SBIBDs*, M.Sc(Hons) Thesis, University of Wollongong, 1999.

Appendix A   Outline of nests program

For a t-(v, k, λ) design D, we suppose the following functions are defined:

> FUNCTION Points(D) {*Returns the point-set of cardinality v of D*}
> FUNCTION Blocks(D) {*Returns the blocks of D*}
> FUNCTION Parameters(D) {*Returns the parameters t, v, k, λ of D*}
> FUNCTION tParameter(D) {*Returns the parameter t of D*}

Suppose also that $\#(X)$ returns the cardinality of X for any set X.

Note that designs are allowed to be non-simple and so design block-sets are *multisets* (i.e. sets of blocks with multiplicities possibly greater than one); and set differences of multisets where they occur respect these multiplicities, e.g. $\{x, x, y, z\} \setminus \{x, y\} = \{x, z\}$.

The following function interfaces with a version of the design completion C program `cad` originally part of the `complete` package written (and described) by Cathy Delaney [4] which was later modified by Colin Ramsay [18] so that it conformed more with the usual input-output conventions of UNIX. To facilitate its use from within the computer algebra system MAGMA [1], Greg Gamble made some further minor modifications.

> FUNCTION CadCompletions(S, D, *simple*)
> > {*Returns the set of designs (sets of blocks) that are all the*
> > *design completions possible for the partial design S on the*

15

parameters $\mathsf{Parameters}(D)$. *If simple = true the design is*
assumed to be simple (i.e. the -s *option is passed to* cad*)*}

END.

The following $\mathsf{AutomorphismGroup}$ function is essentially provided by
MAGMA [1], which in turn uses Brendan McKay's nauty [17].

FUNCTION $\mathsf{AutomorphismGroup}(D,\ S)$
   {*Returns, for the set S of blocks of the design D, the*
   permutation group on $\mathsf{Points}(D)$ that preserves S.
   This is implemented by defining a MAGMA
   $\mathsf{IncidenceStructure}$ *with points $\mathsf{Points}(D)$ and blocks*
   S and then using MAGMA*'s $\mathsf{AutomorphismGroup}$*
   function on the object formed.}

END.

The following iSets function returns the i-sets which are common to all
completions.

FUNCTION $\mathsf{iSets}(C,\ i)$
   {*Returns the i-sets covered by the blocks of C*}

   return $\bigcup \{\{i\text{-subsets of } b\} : b \in C\}$;

END.

The following NDDash function returns the set of partial blocks of sizes
t to $k-1$ which are common to all completions. Observe, that in the code
for NDDash, we could have continued the **for** loop to t (rather than $t+1$)
and returned *UNddi* immediately after the **for** loop, but the reason given
in the comment after the **for** loop allows us to simplify the body of the **for**
loop in the case $i = t$.

FUNCTION $\mathsf{NDDash}(extraC,\ t)$
   {*Returns the N'' component of the $\mathsf{Nest}(S,\ B)$, where S is a*
   defining set of a t-design and B is a block of S, given a set
   of sets of blocks extraC, which is the set of complements
   $(S \setminus B) \cup N'$ of completions of $S \setminus B$, and the parameter

16

t of the t-design.}
LOCAL VARIABLES: *k, UNddi, i, Nddi, c;*

{*Define $N'' = \bigcup_{i \in \{t,...,k\}} N''^{(i)}$*
  where $N''^{(k)} = \emptyset$ and
        $N''^{(i)} = \{i\text{-sets common to elements of } extraC\}$
            $\setminus \{i\text{-sets of } \bigcup_{j \in \{i+1,...,k\}} N''^{(j)}\}$ }

k := #block of an element of *extraC*;
$UNddi$:= $\emptyset$; {*$UNddi = \bigcup_{j \in \{k,...,k\}} N''^{(j)} = N''^{(k)}$*}
for *i* **from** *$k-1$* **downto** *$t+1$* **do**
    {*$Nddi = N''^{(i)}$*}
    $Nddi := \left(\bigcap \{\mathsf{iSets}(c, i) : c \in extraC\} \right) \setminus \mathsf{iSets}(UNddi, i)$
    $UNddi := UNddi \cup Nddi$; {*$UNddi = \bigcup_{j \in \{i,...,k\}} N''^{(j)}$*}
od;
{*All elements of extraC cover the same t-sets.*
 So the common t-sets covered by elements of extraC
 are just the t-sets covered by any element of extraC}
c := any element of *extraC*;
$Nddi := \mathsf{iSets}(c, t) \setminus \mathsf{iSets}(UNddi, t)$;
return *$UNddi \cup Nddi$*;
END.

The following procedure prints the nest data for a minimal defining set *S* of a *t*-design *D*.

PROCEDURE ComputeNests(*S, D, simple*)
    {*Compute and print data associated with the nests of the partial*
     design S, where S consists of a subset of the set of blocks of the
     *design D. If simple = *`true`* the search for design completions is*
     restricted to simple designs.}
    LOCAL VARIABLES: *Scomplement, Strongbox, partialS, C, extraC,*
                 N', N''

    {*Print the defining set itself and its automorphism group*}
    print `"Defining set S:"`, *S*;

17

```
      print "AutS = ", AutomorphismGroup(D, S);


      Scomplement := Blocks(D) \ S;
      Strongbox := Scomplement; {Initialise Strongbox for loop}
      for B in S do
            partialS := S \ {B};
            print "Deleted block B = ", B;
            print "S \ {B} = ", partialS;
            C := CadCompletions(partialS, D, simple);
            {Power(S, B)}
            print "Power(S, B) = #Completions of S \ {B} = ", #C;
            {N' = non-(S \ {B}) blocks common to all completions}
            extraC := {c \ partialS : c ∈ C};
            N' := ⋂ {c : c ∈ extraC};
            print "N' = Extra Common Blocks = ", N';
            {N'' = common partial blocks of completions}
            N'' := NDDash({c \ N' : c ∈ extraC}, tParameter(D));
            print "N'' = ", N'';
            {Influence(S, B)}
            print "Influence(S, B) = ", #Scomplement + 1 − #N';
            Strongbox := Strongbox \ N';
      od;
      {Strongbox(S)}
      print "Strongbox(S)", Strongbox;
      if Strongbox = Scomplement then
            print "Strongbox(S) is the complement of S";
      fi;
END.
```

Appendix B   Summary of Computer Results

The following tables give a summary of the computer results obtained using the nests program described in Section 5 and Appendix A. Each table gives information on designs with particular parameters. For each design

18

considered, we list the automorphism group of the design and how many smallest defining sets and other minimal defining sets were considered. Note that in the tables, the heading minimal defining sets means *other* minimal defining sets. The columns of the tables provide information on the power, influence and strongboxes of the defining sets, under the following headings.

$|DS|$ gives the size of the defining set.

$Aut(DS)$ gives the automorphism group of the defining set.

cases gives the number of defining sets which are described by this row of the table.

σ_P summarises the range of powers found by deleting each block of the defining set. If the distribution of powers is listed exactly then the notation $x^a y^b \ldots$ indicates that the power x occurs a times, power y occurs b times, etc. If the distribution of powers is not given exactly, then the notation $[x, y]$ indicates that the values of the powers are between x and y (inclusive).

σ_I summarises the range of influences found by deleting each block of the defining set, using similar notation to that for the powers.

AS gives the *average strength* for the defining set which is the sum of all the powers and all the influences divided by $|DS|$.

AI gives the *average influence* for the blocks of the defining set.

$|SB|$ gives the number of blocks in the strongbox of the defining set. A complementary strongbox is indicated by a † symbol. The symbol * indicates a near-complementary strongbox, that is, a strongbox of size one less than complementary.

In some cases, defining sets with the same size and strongbox size have been grouped together. In these cases, the columns for average strength and average influence have been left empty. More detailed information on each of these designs (and more designs) is available on the website [14], which provides more information on all of the designs listed in these tables as well as information on designs with the following parameters:

2-$(9, 3, 2)$, 2-$(15, 3, 1)$, 2-$(19, 9, 4)$, 2-$(23, 11, 5)$, 2-$(25, 5, 1)$, 2-$(31, 6, 1)$, 2-$(31, 15, 7)$, 4-$(11, 5, 1)$. It also describes the designs as being simple, non-simple, reducible or irreducible. In the tables in this Appendix, the designs have been labelled using consecutive letters of the alphabet in order to avoid the confusion of having two different designs with the same label. However this is inconsistent with the labelling used on the website, so the caption of each table lists, in parentheses, the labelling used on the website.

In the listing of automorphism groups the following notation is used: I is the identity group, Z_n is the cyclic group on n elements, $A \wr B$ is the wreath product of A by B, $\mathcal{S}_n$ is the symmetric group on n elements, $\mathcal{A}_n$ is the alternating group on n elements, $\mathcal{D}_n$ is the dihedral group of order $2n$, $PSL_n(p)$ is the projective special linear group of the n-dimensional space over $GF(p)$, $AGL_n(p)$ is the affine general linear group of the n-dimensional space over $GF(p)$, $P\Gamma L_n(p)$ is the projective general linear group of the n-dimensional space over $GF(p)$, $\mathcal{F}_n$ is the Frobenius group of order n and $\mathcal{V}_q$ is the elementary abelian group of order q for q a prime power.

Obviously, for some sets of parameters, there are simply too many different designs, or designs with too many different defining sets, to provide complete summaries here. The number of designs considered for each set of parameters is given in the caption for the table. For the small cases, all non-isomorphic minimal defining sets of all non-isomorphic designs were considered. However, for the parameters 2-$(10, 4, 4)$ (Table 9) only one design was considered (the 3-$(10, 4, 1)$ design viewed as a 2-design), and only one smallest defining set and one other minimal defining set were considered. For the parameters 2-$(13, 3, 1)$ (Table 11) only the smallest defining sets were considered, and for the parameters 2-$(15, 7, 3)$ (Table 13) only a selection of smallest and other minimal defining sets were considered. For design E_9 with parameters 2-$(7, 3, 3)$ (Table 5), information is missing on two of the 219 minimal defining sets.

In Table 2, design B_2 is two copies of design A_2 of Table 1. In Table 4 and 5, designs D_2 and E_2 are two and three copies, respectively, of design C_1 from Table 3. It is interesting to note that when multiple copies of a design are taken, the size of the strongbox of a defining set of the non-simple design is less than that of the simple design.

It is worth noting which of the designs and their defining sets have complementary and near-complementary strongboxes.

- The 2-$(10, 4, 2)$ design H_3 (see Table 8 and the list of blocks in [13]) has two defining sets with complementary strongboxes, the unique smallest defining set $\{1, 2, 7, 10, 15\}$ and one of the other 59 minimal defining sets $\{1, 2, 3, 6, 7, 10\}$.

- The 2-$(13, 3, 1)$ design L_1 (see Table 11) has one smallest defining set with a near-complementary strongbox; this is a cyclic design (modulo 13) with starter blocks $1, 14$, where $1 = 125, 14 = 139$. The smallest defining set with blocks $1, 2, 3, 8, 10, 16, 18, 22, 26$ has strongbox consisting of all the other blocks of the design except block 20.

- The 2-$(15, 7, 3)$ design N_5 (see Table 13) has a smallest defining set with a complementary strongbox; the design is cyclic (modulo 15) with starter block $0 = $ 012458a and the smallest defining set consists of the nine blocks $1, 2, 3, 4, 5, 6, 7, 8, 10$ with complementary strongbox. Interestingly, the other minimal defining set has 10 blocks, namely, $0, 1, 2, 5, 6, 7, 9, 10, 11, 13$, and an empty strongbox.

- The 2-$(21, 5, 1)$ design R_1 (see Table 16) is also cyclic (modulo 21), with starter block $1 = 1, 4, 5, 10, 12$. Its smallest defining set with eight blocks $1, 2, 3, 4, 5, 7, 8, 10$ has a complementary strongbox. Interestingly, its three minimal defining sets with nine blocks each have either an empty strongbox (in two cases) or a strongbox with only one block.

- Although not summarised in the tables here, it is worth noting that the 2-$(19, 9, 4)$ design constructed from a starter block consisting of the quadratic residues $1, 4, 5, 6, 7, 9, 11, 16, 17$ has a smallest defining set of blocks $1, 2, 3, 4, 7, 10, 12, 18$ with complementary strongbox.

- Similarly the 2-$(31, 15, 7)$ design constructed from the starter block of quadratic residues $1, 2, 4, 5, 7, 8, 9, 10, 14, 16, 18, 19, 20, 25, 28$ also has a smallest defining set of blocks $1, 2, 3, 4, 5, 8, 9, 10, 26$ with complementary strongbox.

In Tables 9 and 18, the same design is considered, once as a 2-$(10, 4, 4)$ design, J_1, and once as a 3-$(10, 4, 1)$ design, T_1. B.Gray and Ramsay [8] have shown that if s_t and s_{t+1} are the sizes of smallest defining sets of a design considered as a t-design and a $(t + 1)$-design, respectively, then

21

$s_t \geq s_{t+1} + 1$. Here we have $s_2 = 16$ and $s_3 = 4$ so these values lie well within the bounds.

Table 1: 2-$(6, 3, 2)$ design A_1, #blocks $= 10$ (D)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	|SB|
Design A_1, $Aut(A_1) \cong \mathcal{A}_5$, 1 smallest DS							
3	Z_2	1	$2^2 4^1$	$4^2 8^1$	8	5.33	1

Table 2: 2-$(6, 3, 4)$ designs B_1, B_2, B_3, B_4, #blocks $= 20$ (D_1, D_2, D_3, D_4)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	|SB|
Design B_1, $Aut(B_1) \cong \mathcal{S}_6$, 1 smallest DS							
6	S_4	1	6^6	10^6	16	10	0
Design B_2, $Aut(B_2) \cong \mathcal{A}_5$, 1 smallest DS							
6	Z_2	1	$2^4 4^2$	$4^4 8^2$	8	5.33	0
Design B_3, $Aut(B_3) \cong \mathcal{A}_4 \times Z_2$, 6 smallest DS							
6	Z_2	5	$[2, 7]$	$[4, 11]$			0
6	$Z_2 \times Z_2$	1	$2^4 6^2$	$4^4 10^2$	9.33	6	0
Design B_4, $Aut(B_4) \cong \mathcal{S}_4$, 4 smallest DS, 2 minimal DS							
6	$Z_2 \times Z_2$	1	$5^4 4^2$	$9^4 8^2$	13.33	8.67	0
6	Z_2	2	$2^2 3^2 9^2$	$4^2 6^2 12^2$	12	7.33	0
6	I	1	$2^1 3^2 5^1 6^2$	$4^1 6^2 9^1 10^2$	11.67	7.5	0
7	Z_2	1	$2^2 3^4 5^1$	$4^2 6^4 9^1$	8.86	5.86	0
7	I	1	$2^4 5^1 6^2$	$4^4 9^1 10^2$	10	6.43	0

Table 3: 2-$(7, 3, 1)$ design C_1, #blocks $= 7$ (F)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	|SB|
Design C_1, $Aut(C_1) \cong PSL_2(7)$, 1 smallest DS							
3	S_3	1	2^3	4^3	6	4	1

22

Table 4: 2-$(7,3,2)$ designs D_1, D_2, D_3, D_4, #blocks = 14 (D_1, D_2, D_3, D_4)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design D_1, $Aut(D_1) \cong AGL_1(7)$, 17 smallest DS							
6	Z_2	6	$[2,4]$	$[4,8]$			0
6	I	11	$[2,6]$	$[4,9]$			0
Design D_2, $Aut(D_1) \cong PSL_2(7)$, 1 smallest DS							
6	S_3	1	2^6	4^6	6	4	0
Design D_3, $Aut(D_3) \cong \mathcal{S}_4$, 20 smallest DS							
6	I	7	$[2,4]$	$[4,8]$			0
6	Z_2	10	$[2,7]$	$[4,8]$			0
6	S_3	1	2^6	4^6	6	4	0
6	$Z_2 \times Z_2$	1	$2^4 7^2$	$4^4 8^2$	9	5.33	0
6	Z_4	1	$2^4 7^2$	$4^4 8^2$	9	5.33	0
Design D_4, $Aut(D_4) \cong \mathcal{S}_4 \times Z_2$, 8 smallest DS							
6	Z_2	2	$[2,3]$	$[4,7]$			0
6	$Z_2 \times Z_2$	5	$[2,3]$	$[4,7]$			0
6	$\mathcal{D}_6$	1	2^6	4^6	6	4	0

Table 5: 2-$(7,3,3)$ designs $E_1, E_2, E_3 - E_{10}$, #blocks = 21 $(D_1 - D_{10})$

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design E_1, $Aut(E_1) \cong AGL_1(7)$, 2 smallest DS, 18 minimal DS							
7	Z_2	1	9^7	12^7	21	12	7
7	$\mathcal{D}_7$	1	9^7	12^7	21	12	7
8	I or Z_2	5	$5^6 15^2$	$9^6 13^2$	17.5	10	1
9	I or Z_2	10	$[3,9]$	$[6,12]$			0
10	I or Z_2	3	2^{10}	4^{10}	6	4	0
Design E_2, $Aut(E_2) \cong PSL_2(7)$, 1 smallest DS							
9	S_3	1	2^9	4^9	6	4	0
Designs E_3 to E_9: each of these designs has many defining sets but all (except possibly two defining sets of E_9) have empty strongboxes.							
Design E_{10} (only irreducible), $Aut(E_{10}) \cong \mathcal{S}_3$, 378 smallest DS							
9	I or Z_2	7					1
9	I or Z_2	371					0

Table 6: 2-$(8,4,3)$ designs F_1, F_2, F_3, F_4, #blocks $= 14$ (D_1, D_2, D_3, D_4)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design F_1, $Aut(F_1) \cong AGL_3(2)$, 1 smallest DS, 1 minimal DS							
6	$\mathcal{S}_4 \times Z_2$	1	5^6	7^6	12	7	2
8	$\mathcal{D}_4$	1	$3^4 2^4$	$7^4 4^4$	8	5.50	0
Design F_2, $Aut(F_2) \cong \mathcal{S}_4 \times Z_2$, 7 smallest DS, 5 minimal DS							
6	$Z_2 \times Z_2$	1	$3^4 5^2$	$6^4 7^2$	10	6.33	2
6	$Z_2 \times Z_2$	3	$[3,3]$	$[6,7]$			0
6	Z_2	2	$[3,6]$	$[6,9]$			0
6	$\mathcal{D}_6$	1	3^6	6^6	9	6	0
7	I or Z_2	3	$[2,4]$	$[4,8]$			0
8	I or Z_2	2	$[2,3]$	$[4,7]$			0
Design F_3, $Aut(F_3) \cong \mathcal{A}_4$, 32 smallest DS, 2 minimal DS							
6	I or Z_2	32	$[2,9]$	$[4,9]$			0
7	I	2	$4^1 4^1 3^2 2^1 2^2$	$8^1 7^1 7^2 6^1 4^2$	9	6.13	0
Design F_4, $Aut(F_4) \cong \mathcal{F}_{21}$, 27 smallest DS							
6	I	1	$6^2 4^4$	$9^2 8^4$	11.14	8.33	4
6	I	1	$6^4 4^2$	$9^4 8^2$	14	8.67	6
6	I	25	$[2,8]$	$[4,9]$			0

Table 7: 2-$(9,3,1)$ design G_1, #blocks $= 12$ (D_1)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design G_1, $Aut(G_1) \cong AGL_2(3)$, 1 smallest DS, 1 minimal DS							
4	group size 8	1	4^4	8^4	12	8	6
5	Z_2	1	2^5	6^5	8	6	1

Table 8: 2-$(10, 4, 2)$ designs H_1, H_2, H_3, # blocks = 15 (H_1, H_2, H_3)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design H_1, $Aut(H_1) \cong \mathcal{S}_6$, 4 smallest DS, 1 minimal DS							
8	$\mathcal{D}_4$	1	$3^4 2^4$	$7^4 4^4$	8	5.5	0
8	Z_2	1	$3^4 2^4$	$7^4 4^4$	8	5.5	0
8	Z_2	1	$3^3 2^5$	$7^3 4^5$	7.50	5.13	0
8	$\mathcal{D}_4$	1	2^8	4^8	6	4	1
9	$\mathcal{S}_3$	1	2^9	4^9	6	4	0
Design H_2, $Aut(H_2) \cong \mathcal{S}_4 \times Z_2$, 3 smallest DS, 13 minimal DS							
6	I	2	$6^1 3^5$	$10^1 7^5$	11	7.5	0
6	Z_3	1	3^6	7^6	10	7	0
7	I or Z_2	8	$2^6 4^1$	$4^6 9^1$	7	4.71	0
7	I or Z_2	3	2^7	4^7	6	4	0
7	I	2	$4^1 3^2 2^4$	$9^1 7^2 4^4$	8.14	5.57	0
Design H_3, $Aut(H_3) \cong \mathcal{A}_4 \times Z_2$, 1 smallest DS, 59 minimal DS							
5	I	1	8^5	11^5	19	11	10†
6	Z_3	1	$2^3 6^3$	$10^3 10^3$	14	10	9†
6	I	5	$[2, 4]$	$[7, 10]$			4
6	I	9	$[2, 4]$	$[7, 10]$			3
6	I	22	$[2, 4]$	$[7, 10]$			2
6	I or Z_3	18	$[2, 4]$	$[7, 10]$			1
6	I	4	$[2, 4]$	$[7, 10]$			0

Table 9: 2-$(10, 4, 4)$ design J_1, #blocks = 30 (L)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design J_1, $Aut(J_1) \cong P\Gamma L_2(9)$, 1 smallest DS, 1 minimal DS							
16	I	1	$[2, 15]$	$[4, 15]$			0
17	I	1	$[3, 26]$	$[10.14]$			3

Table 10: 2-(11, 5, 2) design K_1, #blocks = 11 (SQ)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design K_1, $Aut(K_1) \cong PSL_2(11)$, 2 smallest DS							
5	Z_2	1	2^5	6^5	8	6	1
5	$\mathcal{D}_5$	1	2^5	6^5	8	6	1

Table 11: 2-(13, 3, 1) designs L_1, L_2, #blocks = 26 (D_1, D_2)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design L_1, $Aut(L_1) \cong \mathcal{F}_{39}$, 17 smallest DS							
9	I	1	$[5, 44]$	$[11, 17]$			4
9	I	1	$[4, 20]$	$[12, 17]$			6
9	I	1	$[8, 32]$	$[13, 18]$			7
9	I	2	$[7, 59]$	$[8, 18]$			8
9	I	3	$[6, 66]$	$[13, 18]$			9
9	I	2	$[4, 40]$	$[12, 18]$			11
9	I	1	$[8, 38]$	$[13, 18]$			12
9	I	2	$[6, 59]$	$[15, 18]$			13
9	I	1	$[8, 41]$	$[15, 18]$			14
9	I	2	$[8, 55]$	$[17, 18]$			15
9	I	1	$[10, 32]$	$[17, 18]$	37.56	17.89	16*
Design L_2, $Aut(L_2) \cong \mathcal{S}_3$, 2 smallest DS							
8	I	1	$[16, 248]$	$[16, 19]$			14
8	I	1	$[16, 432]$	$[16, 19]$			11

Table 12: 2-(13, 4, 1) design M_1, #blocks = 13 (D_1)

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design M_1, $Aut(M_1) \cong PSL_3(3)$, 2 smallest DS							
6	$\mathcal{S}_4$	1	2^6	6^6	8	6	3
6	S_3	1	2^6	6^6	8	6	1

Table 13:   2-$(15,7,3)$ designs N_1, N_2, N_3, N_4, N_5,   #blocks $=$ 15 $(D_1, D_2, D_3, D_4, D_5)$

$|S|$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$|SB|$
Design N_1, $|Aut(N_1)| = 168$, 14 smallest DS							
7	group size 21 or Z_3	2	2^7	8^7	10	8	1
7	Z_3	2	$3^6 2^1$	$7^6 8^1$	10	7.14	1
7	I	2	$3^6 2^1$	$7^6 8^1$	10	7.14	1
7	I	8	$3^5 2^2$	$7^5 8^2$	10	7.29	0
Design N_2, $|Aut(N_2)| = 96$, 6 smallest DS, 2 minimal DS							
7	Z_3	1	2^7	4^7	6	4	0
7	Z_2	2	$4^1 3^4 2^2$	$9^1 7^4 4^2$	9.29	6.43	0
7	I	2	$4^2 3^3 2^2$	$9^2 7^3 4^2$	9.71	6.71	0
7	I	1	$4^2 3^4 2^1$	$9^2 7^4 4^1$	10.29	7.14	1
8	I	2	$3^2 2^6$	$7^2 4^6$	7	4.75	0
Design N_3, $|Aut(N_3)| = 168$, 10 smallest DS							
7	Z_3 or I	2	$3^6 2^1$	$7^6 4^1$	9.43	6.57	0
7	I	4	3^7	7^7	10	7	0
7	I	4	$3^5 2^2$	$7^5 4^2$	8.86	6.14	0
Design N_4, $|Aut(N_4)| = 576$, 2 smallest DS, 5 minimal DS							
8	$Z_2 \times Z_2$	1	$3^6 2^2$	$7^6 4^2$	9	6.25	0
8	Z_2	1	$3^5 2^3$	$7^5 4^3$	8.25	5.83	0
9	Z_2 or $Z_2 \times Z_2$	3	$3^1 2^8$	$7^1 4^8$	7.33	4.33	0
9	$\mathcal{D}_6$ or $\mathcal{S}_3 \times \mathcal{S}_3$	2	2^9	4^9	6	4	0
Design N_5, $|Aut(N_5)| = 20160$, 1 smallest DS, 1 minimal DS							
9	$\mathcal{S}_3 \wr Z_2$	1	3^9	7^9	10	7	6†
10	$\mathcal{S}_5$	1	2^{10}	4^{10}	6	4	0

Table 14: 2-$(16, 4, 1)$ design P_1, #blocks $= 20$ (A)

$\lvert S \rvert$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$\lvert SB \rvert$
Design P_1, $\lvert Aut(P_1) \rvert = 5760$, 4 smallest DS, 8 minimal DS							
7	$\mathcal{D}_6$	1	4^7	12^7	16	12	8
7	$\mathcal{S}_3$	1	$4^6 2^1$	$12^6 8^1$	15.14	11.43	7
7	Z_2	1	$4^6 2^1$	$12^6 8^1$	15.14	11.43	2
7	$Z_2 \times Z_2$	1	$4^6 2^1$	$12^6 8^1$	15.14	11.43	8
8	I	3	2^8	8^8	10	8	0
8	Z_2	2	2^8	8^8	10	8	0
8	Z_2	1	2^8	8^8	10	8	1
8	Z_4	1	2^8	8^8	10	8	0
8	$\mathcal{S}_3$	1	2^8	8^8	10	8	0

Table 15: 2-$(16, 6, 2)$ designs Q_1, Q_2, Q_3, #blocks $= 16$ (G_1, G_2, G_3)

$\lvert S \rvert$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$\lvert SB \rvert$
Design Q_1, $\lvert Aut(Q_1) \rvert = 11520$, 2 smallest DS, 1 minimal DS							
9	group size 72	1	2^9	4^9	6	4	1
9	group size 8	1	$3^4 2^5$	$7^4 4^5$	7.78	5.33	0
10	group size 60	1	2^{10}	4^{10}	10	4	0
Design Q_2, $\lvert Aut(Q_2) \rvert = 768$, 1 smallest DS, 3 minimal DS							
7	Z_3	1	$6^1 3^6$	$10^1 7^6$	10.86	9.43	0
8	group size 8	1	2^8	4^8	6	4	0
8	Z_2	2	$4^1 2^7$	$9^1 4^7$	6.83	4.50	0
Design Q_3, $\lvert Aut(Q_3) \rvert = 384$, 10 smallest DS							
7	Z_3	1	$4^3 2^4$	$9^3 8^4$	8.67	6.56	1
7	I	1	$4^3 2^4$	$9^3 8^4$	8.67	6.56	2
7	I	2	$4^4 2^2 3^1$	$9^4 8^2 7^1$	9.11	6.56	2
7	I	1	$4^2 2^2 3^3$	$9^2 8^2 7^3$	8.45	6.11	1
7	I	1	$4^3 2^2 3^2$	$9^3 8^2 7^2$	8.67	6.33	0
7	I	1	$4^2 2^2 3^3$	$9^2 8^2 7^3$	8.45	6.11	1
7	I	2	$4^3 2^2 3^2$	$9^3 8^2 7^2$	8.67	6.33	1
7	I	1	$4^3 2^2 3^2$	$9^3 8^2 3^2$	8.67	6.33	0

Table 16: 2-$(21, 5, 1)$ design R_1, #blocks $= 21$ (P)

$\lvert S \rvert$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$\lvert$SB$\rvert$
Design R_1, $\lvert Aut(R_1) \rvert = 120960$, 1 smallest DS, 3 minimal DS							
8	group size 48	1	4^8	14^8	18	14	13†
9	group size 12	1	2^9	8^9	10	8	1
9	Z_2 or Z_4	2	2^9	8^9	10	8	0

Table 17: 3-$(8, 4, 1)$ design S_1, #blocks $= 14$ (D_1)

$\lvert S \rvert$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$\lvert$SB$\rvert$
Design S_1, $\lvert Aut(S_1) \rvert = 1344$, 1 smallest DS							
3	S_3	1	2^3	8^3	10	8	3

Table 18: 3-$(10, 4, 1)$ design T_1, #blocks $= 30$ (L)

$\lvert S \rvert$	$Aut(S)$	cases	σ_P	σ_I	AS	AI	$\lvert$SB$\rvert$
Design T_1, $\lvert Aut(T_1) \rvert = 1440$, 3 smallest DS, 29 minimal DS							
4	group size 24	1	4^4	20^4	24	20	4
4	Z_2	1	$4^2 12^2$	$20^2 26^2$	31	23	13
4	Z_2	1	$4^3 12^1$	$20^3 26^1$	27.5	21.5	8
5	Z_2	1	$4^2 4^2 4^1$	$26^2 25^2 20^1$	28.4	24.4	19
5	I	1	$6^1 4^1 4^2 4^1$	$24^1 26^1 25^2 20^1$	28.4	24	18
5	group size 8	1	$4^4 3^1$	$25^4 18^1$	27.4	23.6	17
5	I or Z_2	2	$[2, 4]$	$[16, 26]$			12
5	I or Z_2	6	$[2, 6]$	$[16, 26]$			11
5	I	1	$6^2 4^1 2^2$	$24^2 25^1 16^2$	25	21	10
5	I or Z_2	6	$[2, 4]$	$[16, 25]$			8
5	I or Z_2	4	$[2, 6]$	$[16, 26]$			7
5	I	1	$6^1 4^1 4^1 2^2$	$24^1 25^1 20^1 16^2$	23.8	20.2	6