Designing Efficient Algorithms for Combinatorial Repairable Threshold Schemes

by

Bailey Kacsmar

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Masters of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2018

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Repairable secret sharing schemes are secret sharing schemes where, without the original dealer who distributed the shares, the participants can combine information from their shares to perform a computation that reconstructs a share for a participant who has lost their share. In this work, we study the repairability of a threshold scheme with respect to the probability that it is possible to perform a repair for a failed share, where each participant in the scheme is available with some probability p. We measure the repairability of a scheme in terms of probability that a repair set is available and in terms of the expected number of available repair sets. Additionally, we design efficient algorithms for determining who to contact when attempting to perform a repair on a failed share for repairable threshold schemes which use 2-designs. We also introduce the use of t-designs, for $t > 2$, as distribution designs to produce repairable secret sharing schemes with higher repairing degrees and we discuss modifications to the algorithm to account for the different attributes of the designs where $t > 2$.

Acknowledgements

I would like to thank my supervisor Doug Stinson for his invaluable feedback and advice. Thanks also goes to my committee members Ian Goldberg and David Jao. Finally, I would like to thank Kyle T. for suggesting that pirates could use secret sharing and to thank the members of the CrySP lab for listening to early pirate examples and drawing combinatorial designs on so many of the whiteboards throughout the lab.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Secret sharing is of particular use when a group of people or an organization either lacks trust or requires redundancy. Assume that a group has, in their possession, something that they all value. The valuable could, for example, be a document, a bank account, or even a treasure chest. For our example, we could consider the group of people to be employees at a corporation, tellers at a bank, or classical pirates from novels and films [9]. In the case of the pirates who share a treasure chest, we can see how they may have trust issues, given their penchant in stories for betraying one another to steal the treasure for themselves.

Assume that we have a group of seven pirates who collectively have their pirate treasure. They decide to secure the treasure in such a way that it cannot be accessed by any less than some minimum number of them. Each pirate will possess a share which, when combined with the shares of enough of the other pirates, will provide them with access to their treasure. One interesting thing about using pirates as an example for secret sharing is that, in addition to their trust issues, they also have a need for redundancy. Between storms crashing over ships on open waters, vicious attacks from other pirates, and a dangerous lack of lemons, the life expectancy of a pirate is too volatile to require all seven pirates in order to recover the treasure. All it would take is one extreme act of nature to prevent one of the pirates from ever combining their share with the others and the remaining six pirates would never be able to recover their treasure. Therefore, the pirates can set up a protocol such that, out of the seven pirates who share the treasure chest, at least three of them must work together in order to recover it and no group of fewer than three should be able to acquire the treasure. This is an example of a $(3, 7)$-threshold scheme with seven participants and threshold three. Constructions for threshold schemes were independently developed by by Blakley [2] and Shamir [15] in 1979.

Related to redundancy, we can consider the problem of recovery in the case where a participant loses their share. Extending our example from earlier, assume that, for one of the pirates in the scheme, their ship was sunk in the ocean along with their share. Since they have not violated the pirates' secret sharing code of conduct, they should still be able to participate in recovering the treasure. The pirates do not want to rely on the original distributor of their shares to be around to reconstruct the share that the pirate lost. If the pirates want to enable shares to be repaired, they therefore need to be able to reconstruct a share using only information from pirates that are part of the threshold scheme.

In this thesis, we will be focusing on such threshold schemes, which enable a participant who has lost their share to recover it through communication with other participants in the scheme. After performing a repair, the participant will have regained their original share. Further, no additional information should be revealed to them or any of the other participants. Schemes which enable such repairability have been developed where any subset of participants of sufficient size can perform the repair [8, 13, 19] as well as where only certain subsets of participants of sufficient size can perform the repair [19].

In a paper from 2017, Stinson and Wei [19] presented a repairable threshold scheme which used combinatorial designs to achieve repairability. In this thesis, we will be looking at combinatorial repairability for threshold schemes, specifically with respect to the probability that a repair can be performed using the underlying combinatorial designs. We evaluate properties associated with the probability that a repair can be performed and we construct generalized formulas for these probabilities. Additionally, we investigate how best to choose other participants to contact when attempting to perform a repair and how to design algorithms with trade-offs between storage and computation which enable a participant to find a repair set for their failed share.

1.1 Contributions

This thesis focuses on "combinatorial" repairable threshold schemes. We are interested in the problem of finding participants to participate in a repair and the probability that sufficient participants are available to perform the necessary repairs, where participants in the scheme are not always available.

In this thesis we demonstrate the following thesis statements:

- Combinatorial repairable threshold schemes can be analyzed using methods found in network reliability to demonstrate the robustness of the scheme with respect to performing a repair.

- The reliability of combinatorial repairable threshold schemes can be improved by using t-designs where $t > 2$.

Our contributions can be summarized here as:

- Evaluate and generalize the probability that a set of participants sufficient for performing a repair is available and exists for repairable threshold schemes using 2-designs (defined as (v, k, λ)-BIBDs in Section 2.2)
- Evaluate and generalize the expected number of available sets of participants who are sufficient to perform a repair for repairable threshold schemes using 2-designs
- Design and analyze algorithms for contacting participants sufficient to perform a repair
- Present t-designs, for $t \geq 2$, which can be used to produce repairable threshold schemes. Evaluate these designs against the previous 2-designs and discuss any required modifications to our original algorithms.

1.2 Organization

The organization of this thesis is as follows. Chapter 2 contains background information on secret sharing schemes in general, as well as the combinatorial designs we will need for our discussion of combinatorial repairability and threshold schemes. In Chapter 3, we present some related work on repairable threshold schemes as well as corresponding definitions for such schemes that we will use throughout this thesis. Chapter 4 presents formulas for the probability that a repair set(s) exists, given the threshold scheme is based on a 2-design. In Chapter 5, we present the algorithms for contacting participants in order to perform a repair. The algorithms presented in this chapter will be analyzed under the assumption that the underlying designs for the repairable threshold schemes are 2-designs. Finally, in Chapter 6 we introduce the use of t-designs values of $t > 2$ and we evaluate the implications for the previous repair set properties and earlier algorithms.

Chapter 2

Background

2.1 Secret Sharing Schemes

In this thesis we will be discussing secret sharing schemes which are *unconditionally secure*. This means that all security results are valid against adversaries with unlimited computational power. The schemes presented here will consist of a dealer $\mathbf{D}$, who distributes the shares using a share distribution algorithm to the set of participants $\mathcal{P}$, where $\mathbf{D} \notin \mathcal{P}$. Each scheme will include three phases: *initialization*, *share distribution*, and *reconstruction*. Finally, each scheme will have a *secrecy* property which defines who can access the secret and in what cases they can do so.

2.1.1 Threshold Schemes

Definition 2.1. Let n be the number of participants in the scheme and let τ be the number of participants required to recover the secret, where τ and n are positive integers such that $2 \leq \tau \leq n$. The parameter τ is called the *threshold*. A (τ, n)-*Threshold Scheme* has a dealer choose a secret s and additionally distribute a share to each of the n participants such that:

- *Reconstruction*: Any subset of the n participants of size τ can determine the secret from the shares they hold.
- *Secrecy*: No subset of the n participants consisting of fewer than τ participants is able to gain any knowledge about the secret.

In 1979, constructions for threshold schemes were independently developed by Blakley [2] and Shamir [15]. Blakley's scheme distributes shares as hyperplanes in a finite geometry while Shamir's scheme distributes points lying on a polynomial. For Shamir's scheme, when we combine τ shares using polynomial interpolation we are able to reconstruct the secret. We will present Shamir's construction here, but first we include the following formula for polynomial interpolation.

Theorem 2.2. *[18, Thm. 11.3] Suppose p is prime, suppose $x_1, x_2, \ldots, x_{m+1}$ are distinct elements in $\mathbb{Z}_p$, and suppose $a_1, a_2, \ldots, a_{m+1}$ are (not necessarily distinct) elements in $\mathbb{Z}_p$. Then there is a unique polynomial $F(x) \in \mathbb{Z}_p[x]$ having degree at most m, such that $F(x_i) = a_i, 1 \leq i \leq m+1$. The polynomial $F(x)$ is as follows:*

$$F(x) = \sum_{j=1}^{m+1} a_j \prod_{1 \leq h < m+1, h \neq j} \frac{x - x_h}{x_j - x_h}.$$

Construction 2.3. Shamir (τ, n)-threshold scheme, $\tau \leq n$

Initialization

Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of n participants.

Let the secret space $\mathcal{S}$ be a finite field $\mathbb{Z}_p$ such that p is prime[1] and $p \geq n + 1$.

Share distribution

Let $s \in \mathcal{S}$ be the secret.

1. The dealer **D** selects $\tau - 1$ values independently and uniformly at random from $\mathbb{Z}_p$ as $r_1, \ldots, r_{\tau-1}$.
2. Choose $f \in \mathbb{Z}_p[x]$ as $f(x) = r_{\tau-1}x^{\tau-1} + r_{\tau-2}x^{\tau-2} + \cdots + r_1 x + s$.
3. Dealer distributes $s_i = (i, f(i))$ to participant P_i for $1 \leq i \leq n$.

Reconstruction

[1]Note that Shamir threshold schemes can also be constructed over fields of order q where q is a prime power

A collection of τ or more participants uses their combined shares and performs polynomial interpolation using Theorem 2.2 to recover the equation f of degree $\tau - 1$, which allows the collection of participants to determine the value of the secret $s = f(0)$.

Remark 2.4. Note that there exists optimizations such that you do not have to reconstruct the entire polynomial. See Section 11.5 in *Cryptography, Theory and Practice* [18].

Secrecy

Let there be a coalition of most $\tau - 1$ participants who attempt to determine the secret $s \in \mathbb{Z}_p$. By combining their $\tau - 1$ shares, the coalition computes a polynomial g that is consistent with their shares and a guessed value for the secret $t \in \mathbb{Z}_p$. Observe that t can be any value from $\mathbb{Z}_p$ and still be consistent with a polynomial computed from the $\tau - 1$ shares. Therefore, the probability that s has a particular value has not changed from the point where a coalition has 0 shares or $\tau - 1$ shares. Without τ shares no additional information about s can be learned.

Example 2.5. Consider an example of a $(3, 5)$-threshold scheme. Let the secret $s = 3 \in \mathbb{Z}_{11}$ and let $f(x) = 9x^2 + 4x + 3$.

Share distribution:

P_1 is given $s_1 = f(1) = 5$.

P_2 is given $s_2 = f(2) = 3$.

P_3 is given $s_3 = f(3) = 8$.

P_4 is given $s_4 = f(4) = 9$.

P_5 is given $s_5 = f(5) = 6$.

Reconstruction:

Let P_1, P_3, and P_5 be the collective wanting to recover the secret.

We know $f(x) = a_0 + a_1x + a_2x^2$ for some values of a_0, a_1, and a_2.

Determine $f(x_i)$ for each participant:

$$f(1) = a_0 + a_2 + a_3$$
$$f(3) = a_0 + 3a_2 + 9a_3$$
$$f(5) = a_0 + 5a_2 + 3a_3$$

Through solving the above system of linear equations, or through using polynomial interpolation, we get the result $a_0 = 3$, $a_1 = 4$, and $a_2 = 9$. Therefore we have $a_0 = f(0) = 3 = s$.

2.1.2 Ramp Schemes

Definition 2.6. Let n be the number of participants in the scheme and let τ_1 and τ_2 be the *lower* and *upper thresholds* respectively such that $1 \leq \tau_1 < \tau_2 \leq n$. A (τ_1, τ_2, n)-*Ramp Scheme* has a dealer choose a secret s and distribute a share to each of the n participants such that:

- *Reconstruction*: Any subset of the n participants of size τ_2 can determine the secret from the shares they hold.
- *Secrecy*: No subset of the n participants consisting of at most τ_1 participants is able to gain any knowledge about the secret.

A (τ_1, τ_2, n)-Ramp Scheme is equivalent to a (τ, n)-Threshold Scheme, when $\tau_2 = \tau_1 + 1 = \tau$.

Construction 2.7. Shamir (τ_1, τ_2, n)-ramp scheme, $\tau_1 < \tau_2 \leq n$

Initialization

Let $\mathcal{P}$ be a set of n participants, where $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$.

Let the secret space $\mathcal{S}$ be a finite field $\mathbb{Z}_p$ such that p is prime and $p \geq n + 1$.

Share distribution

Let $s \in \mathcal{S} = (\mathbb{Z}_p)^{t_0}$ be the secret, where $t_0 = \tau_2 - \tau_1$. Note that s is a t_0-tuple, where $s = (r_0, \ldots, r_{t_0-1})$.

1. The dealer **D** selects $\tau_1 = \tau_2 - t_0$ values independently and uniformly at random from $\mathbb{Z}_p$ as $r_{t_0}, \ldots, r_{\tau_2-1}$.

2. Choose $f \in \mathbb{Z}_p[x]$ as $f(x) = r_{\tau_2-1}x^{\tau_2-1} + r_{\tau_2-2}x^{\tau_2-2} + \cdots + r_0$.

3. Dealer distributes $s_i = f(i)$ to participant P_i for $1 \leq i \leq n$.

Reconstruction

A collection of τ_2 or more participants uses their combined shares and performs polynomial interpolation using Theorem 2.2 to recover the equation f of degree $\tau_2 - 1$, which allows the collection of participants to determine the value of the secret $s =$ the t_0 low order coefficients of the function f.

Secrecy

Let there be a coalition of most τ_1 participants who attempt to determine the secret $s \in (\mathbb{Z}_p)^{t_0}$. By combining their τ_1 shares, the coalition computes a polynomial g that is consistent with their shares and a secret, that is a t_0-tuple, $t \in (\mathbb{Z}_p)^{t_0}$. However, t can be any value from $(\mathbb{Z}_p)^{t_0}$ and still be consistent with the polynomial computed from the $\tau - 1$ shares. Therefore, the probability that s has a particular value has not changed from the point where a coalition has 0 shares or τ_1 shares. Without at least $\tau_1 + 1$ shares no additional information about s can be learned. See Example 2.10.

Remark 2.8. Any coalition of at least $\tau_1 + 1$ and at most $\tau_2 - 1$ will not learn the secret; however, the number of possible secrets will be reduced with each additional share beyond τ_1. Therefore, any such coalition of at least $\tau_1 + 1$ will be require fewer guesses to determine the secret. See Example 2.9.

Example 2.9. Consider an example of a $(1, 3, 5)$-ramp scheme where $s \in (\mathbb{Z}_{41})^{t_0}$, where $t_0 = \tau_1 - \tau_2 = 2$. Let $s = (3, 5) \in \mathbb{Z}_{41} \times \mathbb{Z}_{41}$. Let $f(x) = 19x^2 + 5x + 3 \pmod{41}$.

Share distribution:

P_1 is given $s_1 = f(1) = 27$.

P_2 is given $s_2 = f(2) = 7$.

P_3 is given $s_3 = f(3) = 25$.

P_4 is given $s_4 = f(4) = 40$.

P_5 is given $s_5 = f(5) = 11$.

Reconstruction:

Let P_1, P_3, and P_5 be the collective wanting to recover the secret.

We know $f(x) = a_0 + a_1x + a_2x^2$ for some values of a_0, a_1, and a_2.

Determine $f(x_i)$ for each participant:

$$f(1) = a_0 + a_2 + a_3$$
$$f(3) = a_0 + 3a_2 + 9a_3$$
$$f(5) = a_0 + 5a_2 + 3a_3$$

Through solving the above system of linear equations, or through using polynomial interpolation, we get the result $a_0 = 3$, $a_1 = 5$, and $a_2 = 19$. Therefore we have $(a_0, a_1) = (3, 5) = s$.

Example 2.10. Consider an example with respect to secrecy for coalitions of size x, where $\tau_1 < x < \tau_2$. We will continue with the ramp scheme from Example 2.9. Since $\tau_1 = 1$ and $\tau_2 = 3$, the only value for x that satisfies the above conditions would be a coalition of size $x = 2$.

For a coalition of size $\tau_1 = 1$, one can verify that any "guessed" secret is consistent with the share, however, there are 41^2 possible solutions.

For a coalition of size $x = 2$, there will only be 41 possible solutions which are consistent with both of the shares in the coalition. Therefore, for coalitions of size greater than τ_1, but less than τ_2 there is a reduction in the set of possible solutions in comparison to the possible solutions for a coalition of size τ_1 or less.

2.2 Designs

Definition 2.11. A *design* is a pair $(X, \mathcal{A})$ such that X is a finite set of elements called *points*, and $\mathcal{A}$ is a finite collection of non-empty subsets of X called *blocks*, of X.

2.2.1 Balanced Incomplete Block Designs

Definition 2.12. A (v, k, λ)-*Balanced Incomplete Block Design*, or (v, k, λ)-*BIBD*, is a design such that:

1. $|X| = v$,
2. each block contains exactly k points, and
3. every pair of distinct points is contained in exactly λ blocks.

Note that when writing a block B, we can write it as abc rather than $\{a, b, c\}$.

Example 2.13. A $(7, 3, 1)$-BIBD, (X, A), where

$X = \{1, 2, 3, 4, 5, 6, 7\}$ and

$A = \{123, 145, 167, 246, 257, 347, 356\}$.

Theorem 2.14. *[17, Thm. 1.8] Every point in a (v, k, λ)-BIBD occurs in exactly*

$$r = \frac{\lambda(v-1)}{k-1}$$

blocks. The value r is termed the replication number*.*

Theorem 2.15. *[17, Thm. 1.9] A (v, k, λ)-BIBD has exactly*

$$b = \frac{vr}{k} = \frac{\lambda(v^2 - v)}{k^2 - k}$$

blocks of size k.

Example 2.16. Consider the previous example of a $(7, 3, 1)$-BIBD. The replication number is $r = 3$ and the number of blocks in the design is $b = 7$.

2.2.2 Steiner Triple Systems

Definition 2.17. A *Steiner Triple System*, or $STS(v)$, is a $(v, 3, 1)$-BIBD.

Theorem 2.18. *[17, Lem. 6.11] There exists an $STS(v)$ if and only if $v \equiv 1, 3 \pmod 6$, $v \geq 7$.*

2.2.3 Projective Planes

Definition 2.19. An $(n^2+n+1, n+1, 1)$-BIBD with $n \geq 2$ is called a *projective plane* of order n.

Definition 2.20. A BIBD where $b = v$ (or equivalently from Theorem 2.14, $r = k$, or $\lambda(v-1) = k^2 - k)$) is called a *symmetric BIBD*.

Theorem 2.21. *[17, Thm. 2.10] For every prime power $q \geq 2$ there exists a symmetric $(q^2+q+1, q+1, 1)$-BIBD (i.e., a projective plane of order q).*

2.2.4 Difference Sets

Definition 2.22. Let G be an additively written abelian group of order v and let D be a k-subset of G. Let ΔD be the unordered list (or multiset) of differences defined as

$$\Delta D = (d - d' \colon d, d' \in D, d \neq d').$$

Then D is called a (v, k, λ)-*difference set* if $\Delta D = \lambda(G \setminus \{0\})$. What this notation means is that, if we were to consider a list of the differences in such a set, each non-zero element from G would occur exactly λ times.

Example 2.23. Let the group $G = \mathbb{Z}_{21}$ and let $D = \{3, 6, 7, 12, 14\}$.

If we compute all of the differences from D we get the following:

$3 - 6 = 18$	$3 - 7 = 17$	$3 - 12 = 12$	$3 - 14 = 10$
$6 - 3 = 3$	$6 - 7 = 20$	$6 - 12 = 15$	$6 - 14 = 13$
$7 - 3 = 4$	$7 - 6 = 1$	$7 - 12 = 16$	$7 - 14 = 14$
$12 - 3 = 9$	$12 - 6 = 6$	$12 - 7 = 5$	$12 - 14 = 9$
$14 - 3 = 11$	$14 - 6 = 8$	$14 - 7 = 7$	$14 - 12 = 2$

By computing all of the differences using the elements of D, every element in $\mathbb{Z}_{21} \setminus 0$ is produced.

Definition 2.24. Let G be any finite abelian group (written additively) and let $D = \{D_1, \ldots, D_\ell\} \neq \emptyset$ be any subset of G. Then the design $dev\ (D) \coloneqq (G, \mathbf{B})$ with $\mathbf{B} \coloneqq \{D + x : x \in G\}$ is called the *development* of D.

Definition 2.25. Suppose $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ are two designs with $|X| = |Y|$. $(X, \mathcal{A})$ and $(Y, \mathcal{B})$ are *isomorphic* if there exists a bijection $\alpha : X \mapsto Y$ such that

$$[\{\alpha(x) : x \in A\} : A \in \mathcal{A}] = \mathcal{B}.$$

The bijection α is called an *isomorphism.*

Definition 2.26. For a design (X, A), an *automorphism* of (X, A) is an isomorphism of the design with itself.

Definition 2.27. The *automorphism group* of a design consists of all automorphisms of the design.The group operation is composition of permutations.

Definition 2.28. A group G acts transitively on a set D if, for every d_1, d_2 in D, there is a permutation $\pi \in G$ such that π maps d_1 to d_2.

Definition 2.29. A (v, k, λ)-BIBD with an automorphism group Γ is said to be *regular* if Γ contains a subgroup Γ' of order v which acts transitively on the elements.

Theorem 2.30. *[1, Thm. 1.6] Let G be a finite abelian group, and let D be a proper, non-empty subset of G. Then D is a (v, k, λ)-difference set if and only if dev (D) is a symmetric (v, k, λ)-BIBD which is regular with respect to G. Moreover, every regular symmetric (v, k, λ)-BIBD may be represented this way.*

In other words, a (v, k, λ)-difference set D can serve as a base block for a (v, k, λ)-BIBD. All of the blocks in the design can be generated from the "base block". Note, however, that the above theorem does not generalize to difference families, which will be discussed later.

Example 2.31. Consider $D = \{031\}$ and $G = \mathbb{Z}_7$. We can generate all of the blocks in a $(7, 3, 1)$-BIBD by taking the base block $B = 031$ and increasing the value of each of its points by each possible value $x \in G$. For example:

1. Let $x = 4$
2. Increase each point by x modulo 7: $0 + 4$, $3 + 4$, $1 + 4$ to produce the block 405

If we apply this process for all values of x, we get all of the blocks:

$$A = \{124, 235, 346, 450, 156, 260, 130\}.$$

Definition 2.32. Let G be an automorphism group of a design. The *orbit* of a block B is the set of all blocks that can be obtained from B by applying an automorphism in G. The orbit of B is denoted by orbit(B).

Definition 2.33. If a base block B is fixed by some automorphism $g \neq 0$, then its *orbit*(B) is called a short orbit.

Example 2.34. Consider the base blocks $\{0, 1, 4\}$, $\{0, 2, 9\}$, $\{0, 5, 10\}$ for $\mathbb{Z}_{15}$.

1. Let $g = 5$, and apply it to $B = \{0, 5, 10\}$.
2. Since we have $0 + 5 = 5$, $5 + 5 = 10$, and $10 + 5 = 0$ in $\mathbb{Z}_{15}$, we return back to the same $B = \{0, 5, 10\}$.
3. Thus, the base block $\{0, 5, 10\}$ has a short orbit of size five.

2.2.5 Difference Families

Definition 2.35. Let G be an additive abelian group of order v. Then, ℓ k-element subsets of G, $B_i = \{B_{i,1}, B_{i,2} \dots B_{i,k}\}$, $(1 \leq i \leq \ell)$ form a (v, k, λ)-*Difference Family* if every non-zero element of G occurs λ times among the differences $b_{i,x} - b_{i,y}$, $(i = 1, \dots, \ell; x, y = 1, \dots, k, x \neq y)$. The sets B_i are base blocks.

Example 2.36. Let us consider a $(13, 3, 1)$-difference family where G is $(\mathbb{Z}_{13}, +)$ and $D = \{014, 028\}$.

The differences in ΔD produced from 014 are $1, 3, 4, 9, 10, 12$. The differences in ΔD produced from 028 are $2, 5, 6, 7, 8, 11$.

The set $D = \{014, 028\}$ contains two base blocks.

Theorem 2.37. *[17, Thm. 3.46] Suppose $D = \{D_1, \dots, D_\ell\}$ is a (v, k, λ)-difference family in the abelian group $(G, +)$. Then,*

1. *$(G, dev(\{D_1, \dots, D_\ell\}))$ is a (v, k, λ)-BIBD, and*
2. *$Aut(G, dev(\{D_1, \dots, D_\ell\}))$ contains a subgroup $\widehat{G}$ that is isomorphic to G.*

2.2.6 Cyclic Steiner Triple Systems

We can consider a special case of difference families for triple systems.

Definition 2.38. For each integer v, a *difference triple* is defined as a subset of three distinct elements of $\{1, 2, \ldots, v-1\}$ such that:

1. their sum is $0 \pmod{v}$, or
2. one element is the sum of the other two $\pmod{v}$

Definition 2.39. An $STS(v)$ is *cyclic* if it has an automorphism that is a permutation consisting of a single cycle of length v.

Definition 2.40. Heffter's Difference Problems

1. Let $v = 6m + 1$. Is it possible to partition the set $\{1, 2, \ldots, (v-1)/2 = 3m\}$ into difference triples?
2. Let $v = 6m+3$. Is it possible to partition the set $\{1, 2, \ldots, (v-1)/2 = 3m+1\} \backslash \{v/3 = 2m+1\}$ into difference triples?

Solutions to Heffter's first difference problem HDP_1 and Heffter's second difference problem HDP_2 exist due to Peltesohn.

Theorem 2.41. *[6, Thm. 2.17] For all $m \geq 1$, there exists an $HDP_1(m)$ and an $STS(6m+1)$.*

Theorem 2.42. *[6, Thm. 2.18] For all $m \geq 2$, there exists an $HDP_2(m)$ and an $STS(6m+3)$. There is no $HDP_2(1)$.*

There are additional constructions for solving Heffter's difference problems using integer sequences.

Example 2.43. Consider Theorem 2.41 and let $m = 1$.

If $m = 1$, then $v = 6(1) + 1 = 7$. It is possible to partition the set $\{1, 2, 3\}$ into a difference triple.

Each triple (in this case there is only one) from the solution to the HDP_1 can be used to construct the base block for a cyclic $STS(v)$. The difference triple 123 produces the base block $\{0, 1, 1+2)$. That is, the base block for this cyclic $STS(v)$ is $B = 013$.

Example 2.44. Consider Theorem 2.41 and Theorem 2.42 for $m = 2$. For $m = 2$, we have solutions to both HDP_1 and HDP_2.

If $m = 2$, then by Theorem 2.41 $v = 6(2) + 1 = 13$. It is possible to partition the set $\{1, 2, 3, 4, 5, 6\}$ into difference triples, 134 and 256.

Each triple from the solution to the HDP_1 can be used to construct the base block(s) for a cyclic $STS(v)$. The difference triple 134 produces the base block $\{0, 1, 1 + 3)$. The difference triple 256 produces the base block $\{0, 2, 2 + 5\}$. So, the base blocks for the constructed $STS(13)$ are $\mathcal{B} = \{014, 027\}$.

If $m = 2$, then by Theorem 2.42 $v = 6(2) + 3 = 15$. It is possible to partition the set $\{1, 2, 3, 4, 5, 6, 7\} \setminus \{5\}$ into difference triples. For this example we can partition the set such that the set of difference triples is $\{134, 267\}$. These result in the base blocks $\mathcal{B} = \{014, 028\}$. Each of these base blocks can generate 15 blocks including itself. This would result in a total of 30 of the 35 blocks in the design. For this solution we also need to include a base block that has a short orbit. A base block with a short orbit for this design can be specified as $\{0, 5, 10\}$. Therefore, the base blocks for the $STS(15)$ constructed here are $\mathcal{B} = \{014, 028, \{0, 5, 10\}\}$.

Chapter 3

Related Work

In this section we will discuss the concept of share repairability and the corresponding related work. For a threshold scheme, assume that a participant P_ℓ has lost their share. We want P_ℓ to be able to recover their share without intervention form the dealer (who we assume is no longer part of the scheme) through communication with the other participants. Additionally, we assume that for each participant in the scheme that there exists secure connections[2] with each of the other participants in the scheme that they can use to communicate.

3.1 Repairable Secret Sharing Schemes

Definition 3.1. A threshold scheme with n participants and a threshold τ is *repairable* if in addition to reconstruction and secrecy we have the following property:

- Repairability: Assume a participant P_ℓ has lost their share. Then, there exists a subset of the n participants of size d, where $d \geq \tau$, such that each of the d participants can use their share which when combined results in the reconstruction of the failed share. Additionally, after the repair takes place, none of the participants will have gained any information they did not already possess.

Within the repairability property, the results may be combined by either the participants who are providing the "subshares" and who together perform a computation, or

[2]We assume both authentication and confidentiality.

by the participant with the failed share, once they have received all of the required components to recover their share. Note that, when we are discussing performing a repair, we are considering scenarios in which the share cannot be reissued by a dealer, who is unavailable, or "offline". Additionally note that the reconstruction of the share is not the same as recovering the secret.

For a repairable threshold scheme to maintain its threshold it is apparent that it is necessary to require no fewer than τ participants, excluding P_ℓ, to perform a repair, so $d \geq \tau$. When evaluating the security of the scheme, it is additionally important to note that it must retain security against coalitions of participants of size $\tau - 1$ who may pool their information in an attempt to learn the secret. In the case of a coalition of size $\tau - 1$, the participant with the failed share, P_ℓ, may be a part of the coalition but is not required to be. In the repairable threshold schemes presented here, the participants involved in performing a repair do not acquire any additional information that they did not hold before the repair. They will still only possess their own shares. Therefore, we will see throughout that we retain security against a coalition of $\tau - 1$ participants.

Definition 3.2. The *repairing degree* d is the minimum size of the subset of participants required to perform a repair on a failed share.

Definition 3.3. A (τ, d, n)-RTS is a *repairable threshold scheme* with n participants, threshold τ and repairing degree d. It is necessary to have $d \geq \tau$.

When discussing repairable secret sharing schemes there are two types of repairability.

Definition 3.4. *Universal repairability* is the case where any subset of d participants is able to perform a repair for a participant with a failed share.

Definition 3.5. *Restricted repairability* is the case where there exists at least one (possibly more) subset(s) of d participants which is able to perform a repair for a participant with a failed share.

The repairable threshold schemes in this thesis all have only restricted repairability.

3.2 Combinatorial Repairability

We can construct repairable threshold schemes from other secret sharing schemes (such as the Shamir scheme shown in Construction 2.3) through using combinatorial designs

to "expand" the scheme. The following presents the expansion of a threshold scheme to a repairable threshold scheme using designs, but this idea also can be applied to ramp schemes.

Definition 3.6. A *combinatorial repairable threshold scheme* consists of the following:

- a *base scheme*, such as an (ℓ, m)-threshold scheme or an (ℓ_1, ℓ_2, m)-ramp scheme,
- a *distribution design* with m points and n blocks of size k; see Definition 3.8 for when the base scheme is a threshold scheme and Definition 3.11 for when the base scheme is ramp scheme,
- the resulting *expanded* (τ, d, n)-repairable threshold scheme.

Definition 3.7. A (τ, d, n)-repairable threshold scheme has the following components:

- A set of n participants $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$.
- Each participant P_j has a share S_j which corresponds to a block from the distribution design, consisting of k points (subshares).
- Each subshare in a share S_j is represented as $s^j_{i_m}$. The superscript j indicates which share the subshare corresponds to and i_m indicates the point in the design corresponding to the m^{th} subshare, for the share $S_j = \{s^j_{i_1}, s^j_{i_2}, \ldots s^j_{i_k}\}$.

Definition 3.8. A (τ, ℓ)-*distribution design* is a design with n blocks of size k and m points. The distribution design yields an expanded scheme with threshold τ if the following conditions are satisfied:

1. The union of any τ blocks contains at least ℓ points.
2. The union of any $\tau - 1$ blocks contains at most $\ell - 1$ points.

Before presenting Definition 3.11, it is useful to motivate it with some efficiency metrics.

Definition 3.9. The *information rate* of a repairable threshold scheme is defined as the ratio

$$\rho = \frac{\log_2 |\mathcal{S}|}{\log_2 |\mathcal{V}|},$$

where $\mathcal{V}$ is defined as the set of all possible shares and $\mathcal{S}$ is defined as the set of all possible secrets. This metric is used to evaluate the amount of information each player is required to store in comparison to the size of the secret.

Definition 3.10. The *communication complexity* of a repairable threshold scheme is defined as the sum of the sizes of all the messages that have to be transmitted among the players in order to successfully perform a repair for a player within the scheme, divided by the size of the secret.

The second scheme presented by Stinson and Wei [19] includes the use of ramp schemes as base schemes to produce repairable threshold schemes. The use of ramp schemes can result in schemes with higher information rate and lower communication complexity. Recall from Construction 2.7 that the secret for a ramp scheme is a t_0-tuple and so it is larger than the secret in a comparable threshold scheme. From the definition for information rate (Definition 3.9), we know that it compares the size of the information each player stores to the size of the secret. Since a ramp scheme results in a larger secret it is intuitive that using a ramp scheme results in a higher information rate. This applies in a similar fashion to communication complexity as defined in Definition 3.10. Communication complexity divides the amount of messages to be transmitted for a repair by the size of the secret and since the size of the secret is larger for ramp schemes we therefore have a lower communication complexity than for threshold schemes. Essentially, using ramp schemes leads to improvements in efficiency. In general, when using ramp schemes there is a trade-off between the size of the share and security, however, here we have no loss of security and therefore no such trade-off. For this reason, it is useful to consider a more general definition of distribution designs as seen next in Definition 3.11.

Definition 3.11. A $(\tau, \ell_1, \ell_2)-$*distribution design* is a design with n blocks and m points. The distribution design yields an expanded scheme with threshold τ if the following conditions are satisfied:

1. The set of points from the union of any τ blocks contains at least ℓ_2 points.
2. The set of points from the union of any $\tau - 1$ blocks contains at most ℓ_1 points, where $\ell_2 - \ell_1 \geq 1$.

Definition 3.12. A distribution design is *repairable* if every point in the distribution design occurs in at least two blocks.

Consider the condition in Definition 3.12. Each of the τ participants in the scheme correspond to one of the blocks in the distribution design. Assume that a participant, P_ℓ has lost their share corresponding to their block. In order for a repair to be possible, and by extension for the scheme to be repairable, it is necessary that P_ℓ can acquire all of the lost subshares from some other participants in the scheme. If no such set of participants exists,

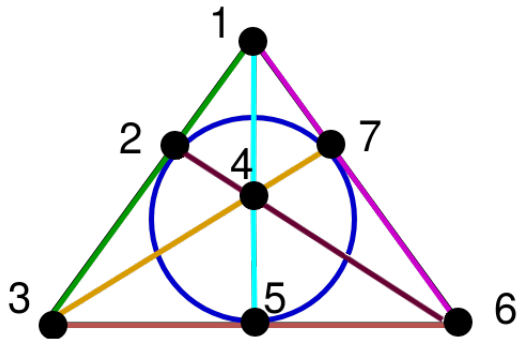


Figure 3.1: Fano Plane, $(7, 3, 1)-$BIBD

then no participant can provide the subshares and it cannot be repaired. However, if both P_ℓ and one additional participant possess the lost secret, then the additional participant could communicate it to P_ℓ and the secret be repaired.

Theorem 3.13. *[19, Thm. 4.1] Suppose there exists a repairable (τ, ℓ_1, ℓ_2)-distribution design on m points with n blocks of size k, and suppose that Q is prime and $Q \geq m + 1$. Then there is a (τ, d, n)-repairable threshold scheme with restricted repairability, having information rate $(\ell_2 - \ell_1)/k$ and communication complexity $k/(\ell_2 - \ell_1)$, where $d \leq k$ and every share is in $(\mathbb{F}_Q)^d$.*

For Theorem 3.13, each participant P_j will be given a distinct block from the n blocks in the distribution design to determine their share S_j. Each of the k points within that block corresponds to a subshare belonging to S_j. In order to perform a repair, if we assume that any other randomly chosen participant in the scheme can provide at most one subshare, then it will require $d = k$ participants to perform a repair. If a participant may be able to provide more than one subshare to another participant, then it may be sufficient for $d < k$ participants to perform a repair. In both of these cases, it is still necessary that $d \geq \tau$ to preserve the secrecy requirements.

Example 3.14. Consider a $(\tau, d, n)-$repairable threshold scheme where the underlying distribution design is a $(7, 3, 1)-$BIBD such as that found in Figure 3.1. Then the $n = 7$ participants correspond to the $b = 7$ blocks in the design (here $n = b$). The union of any two blocks contains at least five points and the union of any one block contains at most three points. Therefore, we can use the $(7, 3, 1)$-BIBD as a $(2, 3, 5)$-distribution design. The repairing degree is $d = 3$ as any pair of blocks has at most one point in common. The resulting expanded scheme is a $(2, 3, 7)$-repairable threshold scheme.

Base Scheme

Construct a $(5, 7)$-threshold scheme or a $(3, 5, 7)$-ramp scheme. The shares from the base scheme are $S_1, S_2 \ldots, S_7$.

Distribution Design

Assign the blocks of the $(7, 3, 1)$-BIBD as follows:

$P_1 = 123$	$P_3 = 167$	$P_5 = 257$	$P_7 = 356$
$P_2 = 145$	$P_4 = 246$	$P_6 = 347$	

Distribute Base Scheme Shares to Participants

Distribute each S_i to all players having point i from the block design.

P_1's expanded scheme share contains S_1, S_2, S_3.

P_2's expanded scheme share contains S_1, S_4, S_5.

P_3's expanded scheme share contains S_1, S_6, S_7.

P_4's expanded scheme share contains S_2, S_4, S_6.

P_5's expanded scheme share contains S_2, S_5, S_7.

P_6's expanded scheme share contains S_3, S_4, S_7.

P_7's expanded scheme share contains S_3, S_5, S_6.

Reconstruction

Let P_1 and P_2 wish to reconstruct the secret.

Between them they have five distinct points S_1, S_2, S_3, S_4, and S_5.

Since the base scheme has $\tau = 5$ or $\tau_2 = 5$ (depending on the base scheme chosen), the participants P_1 and P_2 can recover the secret s.

Repair

Let P_6 require a repair.

Either P_1 or P_7 can provide S_3.

Either P_2 or P_4 can provide S_4.

Either P_3 or P_5 can provide S_7.

Efficiency Metrics

Information rate can be computed using Theorem 3.13.

If we choose to use a $(5, 7)$-threshold scheme the information rate is $1/3$.

If we choose to use a $(3, 5, 7)$-ramp scheme the information rate is $2/3$.

Communication Complexity can be computed using Theorem 3.13.

For a $(5, 7)$-threshold scheme the communication complexity is 3.

For a $(3, 5, 7)$-ramp scheme the communication complexity is $3/2$.

3.3 Other Schemes

In this section we will first introduce related work due to Stinson and Wei [19]. Their work is directly related to this thesis in the sense that the reliability metrics we will define apply directly to their scheme and our defined algorithms apply to their scheme. Furthermore, in Chapter 6.1, we will introduce different combinatorial designs that were not included in their scheme.

Stinson and Wei [19] present two techniques for constructing repairable threshold schemes. The first of these schemes (the "enrollment protocol") is based on work by Nojoumian et al. [13]. The second scheme uses distribution designs to allocate shares from the base scheme to the participants in order to produce an expanded repairable threshold scheme. In Section 5 of their paper, they present and evaluate different combinatorial designs as distribution designs. The designs they consider include Steiner triple systems, balanced incomplete block designs with $\lambda = 1$, and projective planes. Projective planes were used to provide more possible values for τ as well as to achieve smaller repairing sets. Each of these designs produce repairable threshold schemes with restricted repairability. They additionally consider possible combinatorial solutions which produce universal repairability, although we will only be working with repairable threshold schemes which have restricted repairability. The algorithms presented here can be applied when using the designs described in Stinson and Wei's work as distribution designs. Additionally, in

Chapter 6 we will consider the use of different designs than those used by Stinson and Wei as distribution designs (namely, t-designs with $t > 2$) which will also work with our algorithms (although with some modifications).

In the following we include related constructions for repairable threshold schemes. None of the following, or preceeding, or works consider the reliability of their schemes under a model where participants can be unavailable. Therefore, they are included as reference to other methodologies as opposed to direct comparisons to our work.

Other constructions of repairable threshold schemes are obtained from *secure regenerating codes*. These codes can be separated into two types: *Minimum Bandwidth Regenerating* (MBR) codes, which minimize communication requirements when performing a repair; and *Minimum Storage Regenerating* (MSR) codes, which, of course, minimize the storage required in order to enable share repairing. An example of a repairable threshold scheme using MBR codes is due to Guang et al. [8]. Unlike the combinatorial schemes considered in this thesis, the scheme due to Guang et al. provides universal repairability.

Rouayheb and Ramchandran introduced an *exact MBR* code in a paper in 2010 [7] using what they call *fractional repetition* codes. The fractional repetition codes are based on *regular graphs* and Steiner systems. The fractional repetition codes can be understood as serving a role similar to the distribution designs, however, they do not preserve both properties of the distribution design. In our terminology, each participant contributing to a repair provides one subshare only. The data is distributed such that the union of d participants will contain at least k distinct subshares. Unlike the distribution designs, however, they do not have the property that the union of $\tau - 1$ participants contains at most ℓ_2 points which is necessary for maintaining the threshold property security against coalitions of $\tau - 1$ which we require. Therefore, their work does not yield a threshold scheme.

In a survey due to Laing and Stinson [11], they include a comparison of the different approaches to repairable threshold schemes. The comparison is with respect to information rate, communication complexity, and repairing degree. One such comparison looks at comparing an MBR based scheme due to Shah et al. [14] to the combinatorial schemes due to Stinson and Wei [19]. When compared, these two schemes generally achieve equivalent information rate and communication complexity with the exception of a few cases. For the exceptions, the MBR schemes do better then the combinatorial schemes using BIBDs for information rate and communication complexity. Laing and Stinson [11] only compare MBR based repairable threshold schemes and combinatorial based repairable threshold schemes as there are no secure MSR based repairable threshold schemes in the literature[3]

[3]Note that we help minimize the storage for the combinatorial schemes with Algorithm 4.

to compare with the combinatorial based threshold schemes. The conclusion of the comparison accounts for this computation requirement. Combinatorial schemes, are efficient in that they do not require further computation to repair a share. In the case of the MBR based schemes it is necessary to compute linear combinations when performing a repair. In the case where we are prioritizing communication complexity and when the participants are able to compute linear combinations, then the MBR based schemes are most appropriate. In the case where communication complexity is a priority, but such linear computations are not possible it is most appropriate to use combinatorial based repairable threshold schemes.

Chapter 4

Properties of Repairability

Definition 4.1. A *repair set* consists of d participants from the scheme which provide a subshare to a participant P_ℓ in order to repair P_ℓ's share.

Definition 4.2. The *reliability* of a repairable threshold scheme is the probability that a repair can be performed, given that each participant is available with some probability p, and each probability is independent of one another.

4.1 Availability

When the need for a repair arises, a participant with a failed share can contact other participants. Each contacted participant will respond with some probability p. The following presents two different understandings of the probability p, with respect to the availability of the participants in the scheme.

4.1.1 Permanent Fault

For a permanent fault, when we contact a participant P_j, it is available with probability p. If they are not available and if we were to contact them again, they would still be unavailable. When attempting to perform a repair, this share is therefore of no further value to us and it would no longer be considered when contacting additional participants for a repair.

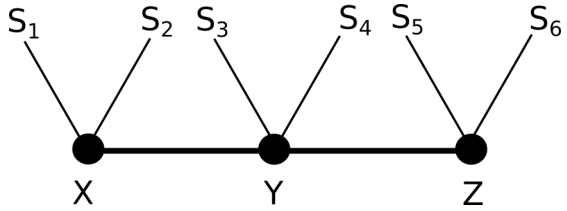


Figure 4.1: Intersecting participants for P_ℓ using a $STS(7)$

4.1.2 Transient Fault

In this case of a transient fault, when we contact a participant to request a repair, they will be available to answer our request with probability p. This means that if we contacted a participant P_j and we did not receive an answer, we could continue to periodically contact them with the expectation that, at some point, we would receive a response. If a participant with a failed share contacts P_j and does not receive a response, it can continually attempt to contact random participants from the scheme, including P_j, until a repair is successful.

Remark 4.3. Throughout this work we will evaluate reliability metrics dependent on the value of p. These metrics include the existence of a repair set and the expected number of available repair sets. The computations and theorems corresponding to these metrics are independent of the fault model. In the permanent model the status of the model does not continually change. In the transient model the status of the model does continually change; however, at any snapshot in time we are able to compute existence and expectation given p.

4.2 Existence of a Repairing Set

Consider an $STS(7)$. Each subshare occurs in $r = 3$ shares. For each subshare x, y, z in a share, there exist two other shares, which also contain that subshare, with no share containing more than one of x, y, or z.

We can begin by considering the probability that there is at least one repairing set available. Let $S_1, \ldots, S_6$, be shares, where the intersection of each share with P_ℓ's share is non-empty. Let x, y, and z (see Figure 4.1) represent participant P_ℓ's share. In order to successfully repair P_ℓ's share, we require at least one of $\{S_1, S_2\}$, one of $\{S_3, S_4\}$ and one of $\{S_5, S_6\}$ to be available. Let p be the probability that a participant is available and let

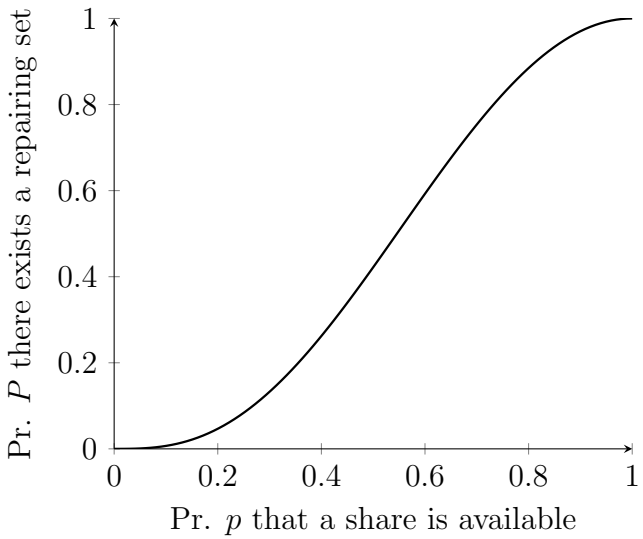


Figure 4.2: Existence of an available repair set for $STS(7)$

$\mathcal{R}(p) = Pr\{\text{a repair set exists}\}$. Then Pr{at least one of $\{S_1, S_2\}$ is available} is

$$1 - (1-p)^2 = p + (1-p)p = 2p - p^2.$$

Since p is the probability for any P_i to be available and each probability is independent of one another, the probability that there exists a repairing set for a failed node P_ℓ is

$$\mathcal{R}(p) = (2p - p^2)^3.$$

This function is graphed in Figure 4.2 for a range of values of p.

More generally, we have the following result for the existence of a repair set in a repairable threshold scheme where the underlying distribution design is a Steiner triple system.

Theorem 4.4. *For an $STS(v)$, the probability that there exists at least one repairing set is:*

$$\mathcal{R}(p) = (1 - (1-p)^{r-1})^3.$$

Proof. Let $\mathcal{R}(p) = Pr\{\text{a repair set exists}\}$ for a failed block B. Let p be the probability that a participant is available.

Each subshare occurs $r-1$ times in the scheme other than its occurrence in B. In order to repair any subshare within the share we need at least one of the other $r-1$ participants

to be available. The probability that at least one of the other $r-1$ participants is available is

$$1-(1-p)^{r-1}.$$

The probability for any participant to be available is independent of one another and each participant can help you recover at most one subshare. Therefore, the probability that we can repair all three subshares is:

$$\mathcal{R}(p)=(1-(1-p)^{r-1})^3.$$

□

The graph found in Figure 4.3 shows the graph of the existence of available repair sets for a number of different values of v. The smallest value of v is the $STS(7)$ with $r=3$ (first thin red line), followed by the $STS(9)$ with $r=4$ (first thick blue line). Alternating colors the remaining values of v graphed are $STS(13)$ with $r=6$, $STS(15)$ with $r=7$, $STS(19)$ with $r=9$, $STS(21)$ with $r=10$, $STS(25)$ with $r=12$, $STS(27)$ with $r=13$, $STS(31)$ with $r=15$, $STS(33)$ with $r=16$, and $STS(37)$ with $r=18$.

We can further generalize from the case $k=3$, to arbitrary values of k.

Theorem 4.5. *For a $(v,k,1)-BIBD$, the probability that there exists at least one repairing set is:*

$$\mathcal{R}(p)=(1-(1-p)^{r-1})^k.$$

Proof. This follows using the same reasoning as in the proof of Theorem 4.4. Instead of having three subshares which are each repairable with probability $(1-(1-p)^{r-1}$, there are k subshares and each participant can help recover at most one subshare. Therefore, the probability that there exists a repair set is

$$\mathcal{R}(p)=((1-(1-p)^{r-1})^k.$$

□

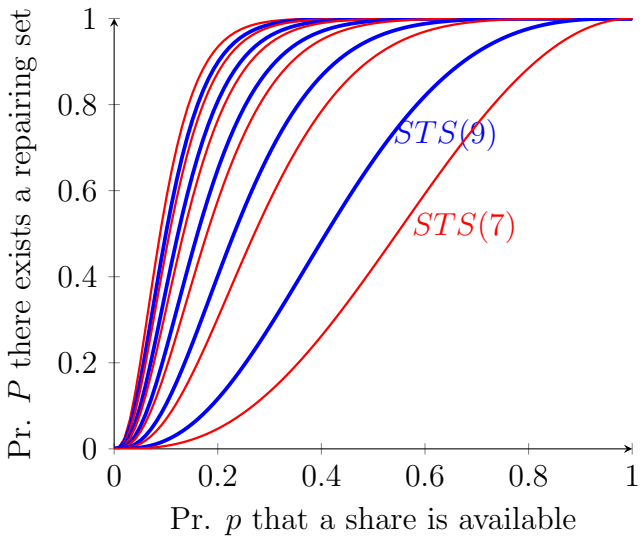


Figure 4.3: Existence of available repairing sets for $STS(v)$

4.3 Expectation for Steiner Triple Systems

Remark 4.6. Besides being a conventional metric, expectation is useful for understanding the difference between repairable threshold schemes with restricted repairability and repairable threshold schemes with universal repairablity. Recall, in a universal schemes, any subset of participants of size d can perform a repair. The expected number of available repair sets for a universal scheme would be $\binom{n-1}{d}p^d$, where n is the number of participants in the scheme, d is the repairing degree, and p is the probability a participant is available.

In this section, we will consider the expected number of repair sets which will be available given the probability p that any share is available. When a single share requires repair, there are six other shares in the set of blocks for the $STS(7)$. If all shares are available there are eight possible repairing sets. If any one share is unavailable, there are then only four possible repairing sets. If two shares are unavailable such that both of them have the same subshare (for example S_1 and S_2), then a repair is not possible. This occurs with probability $3(1-p)^2$. If two shares are unavailable, but do not both supply the same subshare for the repair, then there are two repairing sets available with probability $3(2p-2p^2)^2p^2$. For there to be only one available repairing set, there are three unavailable shares such that none of the unavailable shares contribute the same subshare for the repair. This corresponds to the probability $(2p-2p^2)^3$. This is summarized in Table 4.1.

Table 4.1: Repair Set Probability Distribution for $STS(7)$

Number of sets X	0	1	2	4	8
Pr of X	$3(1-p)^2$	$(2p-2p^2)^3$	$3(2p-2p^2)^2p^2$	$3(2p-2p^2)p^4$	p^6

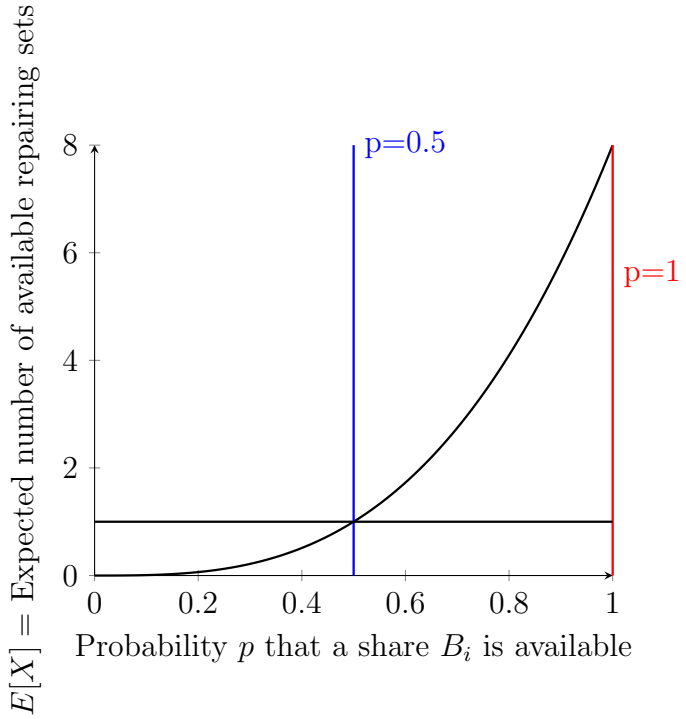


Figure 4.4: Expected number of available repair sets for $STS(7)$

Let X be the number of available repairing sets in Table 4.1. Then,

$$E(X) = 1(2p-2p^2)^3 + 2\cdot 3(2p-2p^2)^2p^2 + 4\cdot 3(2p-2p^2)p^4 + 8p^6$$

$$= 8p^3.$$

Definition 4.7. *Linearity of Expectation* asserts that the expected value of a sum of random variables is equal to the sum of the expected values for each of the random variables. That is,

$$E[X_1 + X_2 + ... + X_n] = E[X_1] + E[X_2] + ... + E[X_n].$$

In the above example, we determined the expectation by considering the probabilities for each possible value of X (as seen in the probability tables). However, using linearity of expectation, the computation is simplified as follows:

Theorem 4.8. *The expected number of available repair sets for an $STS(v)$ is:*

$$(r-1)^3 p^3.$$

Proof. Let X be the number of available repairing sets and let $X = X_1 + X_2 + ... + X_n$, where n is the number of possible repairing sets and let

$$X_i = \begin{cases} 1, & \text{if the } i^{th} \text{ repairing set is available} \\ 0, & \text{otherwise} \end{cases}$$

Then $E[X] = E[X_1] + E[X_2] + ... + E[X_n]$.

When generalizing expectation for Steiner Triple Systems with consideration for linearity of expectation, we need to consider the total number of possible repairing sets for a failed share. For a Steiner Triple System each share consists of three subshares. Each of these subshares occurs in $(r-1)$ other shares. Therefore, for any failed share there exists $(r-1)^3$ possible repairing sets. The expected number of available repair sets is therefore

$$E[X] = E[X_1] + E[X_2] + ... + E[X_{(r-1)^3}].$$

The second thing to consider is the availability of each of these possible repairing sets which is the value for $E[X_i]$. Each repairing set is available with probability p^3. The expected number of available repair sets given the probability is therefore

$$E[X] = p^3 + p^3 + ... + p^3 = \sum_{i=1}^{(r-1)^3} p^3 = (r-1)^3 p^3.$$

□

4.4 Expectation for Balanced Incomplete Block Designs

The expectation for Balanced Incomplete Block Designs follows similarly to that for Steiner Triple Systems. The difference is that there are $(r-1)^k$ possible repairing sets for a $(v, k, \lambda) - BIBD$ and that each of these sets occurs with probability p^k.

Theorem 4.9. *The expected number of available repairing sets for a* $(v, k, \lambda) - BIBD$ *is*

$$(r-1)^k p^k.$$

Proof. This follows in a similar manner to Theorem 4.5. Instead of having three subshares, there are k subshares. □

We can further note that, if $p \geq \frac{1}{r-1}$, then $E[X] \geq 1$.

The equation for the expected number of available repair sets for $STS(7)$ is graphed in Figure 4.4. When the probability that a share is available is 0.5, the expected number of available repairing sets $E[X] = 1$.

Chapter 5

Algorithms

This section considers algorithms with trade-offs between computation complexity and storage complexity. Algorithms within this section assume a security model where the participants are "honest-but-curious" (hbc) in that they will follow the protocol as described but may attempt to collude with one another to gain additional information. For example, a node P_ℓ requesting a subshare from another participant will only request subshares corresponding to its original share. In this chapter it is assumed that each participant has at most one subshare in common with any other participant; however, this assumption will not hold later on in Chapter 6. For this chapter, our assumption means that when P_ℓ requests a subshare from another participant, then that participant can provide at most one of the subshares held by P_ℓ.

For all of the algorithms below, each participant P_j stores their own share, which includes each of their k subshare values $S_j = \{s^j_{i_1}, s^j_{i_2}, \ldots s^j_{i_k}\}$. Any additional storage requirements will be indicated along with the corresponding probability model for that algorithm. The probability model will either be a permanent fault or a transient fault as discussed in Section 4.1.

The algorithms that will be presented in this chapter are:

Algorithm 1: Random Participants. This algorithm is analyzed under the transient fault model and has the smallest storage requirements as each participant only stores their own share. Although it has the smallest storage requirements,it has the highest expected complexity.

Algorithm 2: Stored Intersecting Participants. This algorithm is also analyzed under the transient fault model. It adds an additional storage requirement in order to

achieve a complexity improvement. In addition to storing their own share, each participant also stores a set containing all of the participants which intersect with their share. A participant who has a failed share will not have additional complexity from contacting participants that do not possess any common subshares because of the additional storage.

Algorithm 3: Stored Grouped Participants. For this algorithm, we modify the storage requirements for an additional complexity improvement. Note that for this algorithm we will change the availability model. The algorithm is analyzed under the permanent fault model instead of the transient fault model. For the storage requirements, each participant, in addition to storing a list of participants which intersect with its share, stores which of their failed subshares each participant intersects with.

Algorithm 4: Generating Participants. The final algorithm is also analyzed under the permanent fault model. It maintains the complexity of Algorithm 3, but with a storage improvement due to generating the appropriate intersecting participants instead of storing them.

Remark 5.1. The value T is a constant that will be found in the analysis of all of the algorithms. T is chosen with knowledge of the latency of the network and how long it should take to get a reply from an available participant. Therefore, it appears only as a constant in the analyses we do.

Remark 5.2. In this section we will present algorithms that assume a participant who has lost their share is still able to retain information about the other participant and the labels corresponding the block that maps to their subshares. This storage is possible because unlike storing the share values, we can store the design, the participants, and the information about which points from the design a participant's subshare corresponds to, are not sensitive. All of this information can be public and therefore can be stored in less secure storage, such as ROM or even publicly.

5.1 Algorithm 1: Random Participants

Probability Model: In this algorithm, the probability p can be understood as a transient fault.

Storage: No additional storage beyond each participant P_j's share S_j.

Algorithm 1 RandomParticipants($P_\ell, \mathcal{P}$)

1: /* Performs a repair on S_ℓ */
2: **while** there remains any subshare in S_ℓ requiring repair **do**
3: Contact a random participant from $\mathcal{P}$ and request values for any of $\{s^\ell_{i_1}, s^\ell_{i_2}, \ldots, s^\ell_{i_k}\}$ that P_ℓ does not already have
4: Wait time T for a response
5: **return** $S_\ell = \{s^\ell_{i_1}, s^\ell_{i_2}, \ldots, s^\ell_{i_k}\}$ for P_ℓ

This first algorithm attempts to perform a repair on a failed node $P_\ell \in \mathcal{P}$ without any knowledge of which participants in the scheme have shares which intersect with the player performing a repair. It therefore takes as input the participant P_ℓ, which includes its associated subshare labels (but not the values), as well as the set of other participants, $\mathcal{P}$. If the algorithm is successful, it will produce the k subshare values $S_\ell = \{s^\ell_{i_1}, s^\ell_{i_2}, \ldots, s^\ell_{i_k}\}$ belonging to the participant P_ℓ.

In order to perform a repair, the participant P_ℓ first contacts a random participant $P_j \in \mathcal{P}$ from the scheme who may or not have been previously contacted in the current repair attempt. Participant P_ℓ makes a request for any of the values from S_ℓ that they do not already have. The participant P_j may or may not be online, and so P_ℓ waits time T for a response. If P_j has one of the required subshares, they can then provide it to P_ℓ. Note that it is possible that participant P_j may possess a subshare that P_ℓ has already repaired, but this does not affect how the algorithm proceeds. If, after contacting the current participant P_j, the participant P_ℓ still has remaining subshares requiring repair, then P_ℓ repeats the process from the beginning.

5.1.1 Complexity Analysis

Average Case:

The analysis for Algorithm 1 is a variant on the classic coupon collector problem.

When considering the analysis for this algorithm we can define k states to consider. The *states* are as follows:

State 1: From the beginning, where no subshares have been repaired, until the first successful subshare repair.

State 2: From the first successful subshare repair to the second successful subshare repair.

$\vdots$

State k: From the $(k-1)^{\text{th}}$ successful subshare repair to the final successful repair which will mean that all k subshares have been repaired.

Let $E[n_i]$ be the expected number of participants contacted in the i^{th} state. Then we can represent the expected number of participants contacted in acquiring all k subshares as

$$\sum_{i=1}^{k} E[n_i].$$

For State 1, we have to account for whether the participant P_j is online and whether their share intersects with S_ℓ. For the first part, they are online with probability p. We select P_j from the $b = \frac{vr}{k}$ blocks. Note that this could be $b-1$ blocks, which would be the number of blocks in the design, excluding the one corresponding to the failed share. However, it will be simpler to use b throughout and include player P_ℓ in the possible b participants. Of these b participants, $k(r-1)$ of them intersect S_ℓ in one subshare. The probability that a randomly chosen subshare intersects S_ℓ in one subshare, and therefore can contribute a subshare is

$$\frac{k(r-1)}{b}.$$

For State 2, we will have $(r-1)$ participants, who even if they intersect S_ℓ and are online, cannot contribute a subshare. This is because there are $r-1$ participants where the intersection of their share with S_ℓ is the subshare repaired in the previous state. Therefore, instead of $k(r-1)$ potential intersecting participants who can provide a subshare, there are only $(k-1)(r-1)$ potential participants. Therefore, the probability that a randomly chosen participant intersects S_ℓ and does not contain the subshare previously repaired is

$$\frac{(k-1)(r-1)}{b}.$$

When generalized to the i^{th} stage, we have to consider that of the $k(r-1)$ intersecting participants, $(i-1)(r-1)$ of them have a subshare already provided in the previous stage as their intersecting point.The number of intersecting participants, excluding those which contain subshares already repaired is

$$\begin{aligned} &k(r-1) - (i-1)(r-1) \\ &= (r-1)(k-i+1). \end{aligned}$$

Therefore the expected number of participants contacted in the i$^{\text{th}}$ stage, $E[n_i]$, (including the probability p that they are available) is

$$E[n_i] = \frac{b}{(r-1)(k-i+1)p}.$$

Acquiring all k subshares has a total complexity of

$$\frac{b}{p}\sum_{i=1}^{k}\frac{1}{(k-i+1)(r-1)}$$

$$= \frac{b}{p(r-1)}\sum_{i=1}^{k}\frac{1}{(k-i+1)}$$

$$\approx \frac{b}{p(r-1)}\ln k.$$

Since each time we contact a participant we wait time T for a possible response, we include it as a factor as well, and therefore we have

$$\text{the expected time } = T\frac{b}{p(r-1)}\ln k.$$

5.2 Algorithm 2: Stored Intersecting Participants

Probability Model: In this algorithm, the probability p can be understood as a transient fault.

Storage: In addition to their share S_j, each participant stores a set $\mathcal{R} \subset \mathcal{P}$ consisting of all participants that intersect with participant P_ℓ and therefore can potentially provide a repair for one of their subshares. For every participant $P_j \in \mathcal{R}$, we have $|P_j \cap P_\ell| = 1$. This set does not indicate which subshare a participant in $\mathcal{R}$ can repair.

This algorithm attempts to perform a repair on a failed node $P_\ell \in \mathcal{P}$. It functions essentially the same as Algorithm 1, but instead of contacting a random participant from the set of all participants $\mathcal{P}$, it contacts a random participant from the set $\mathcal{R}$. It takes as input the participant P_ℓ, which includes its associated subshare labels (but not the values),

Algorithm 2 StoredIntersectingParticipants($P_\ell, \mathcal{R}$)

1: /* Performs a repair on S_ℓ */
2: **while** there remains any subshare in S_ℓ requiring repair **do**
3: Contact a random participant $P_j \in \mathcal{R}$ and request they send values for any of $\{s^\ell_{i_1}, s^\ell_{i_2}, \ldots, s^\ell_{i_k}\}$ that P_ℓ does not already have
4: Wait time T for a response
5: **return** $S_\ell = \{s^\ell_{i_1}, s^\ell_{i_2}, \ldots, s^\ell_{i_k}\}$ for P_ℓ

as well as the set of other potential repairing participants, $\mathcal{R}$. If the algorithm is successful, it will produce the k subshare values $S_\ell = \{s^\ell_{i_1}, s^\ell_{i_2}, \ldots, s^\ell_{i_k}\}$ belonging to the participant P_ℓ.

In order to perform a repair, the participant P_ℓ first contacts a random participant $P_j \in \mathcal{R}$ from the scheme who may or may not have been previously contacted in the current repair attempt. Participant P_ℓ makes a request for any of the values from S_ℓ that it has not already repaired. The participant P_j may or may not be online, and so P_ℓ waits time T for a response. If P_j has one of the required subshares, they can then provide it to P_ℓ. If, after acquiring a subshare and in the case where no subshare is provided, the participant P_ℓ still has remaining subshares requiring repair, then P_ℓ repeats the process from the beginning.

5.2.1 Complexity Analysis

Average Case:

The analysis for Algorithm 2 is also a variant on the classic coupon collector problem.

When considering the analysis for this algorithm we again can define k states. The states are as follows:

State 0: From the beginning, where no subshares have been repaired, until the first successful subshare repair.

State 1: From the first successful subshare repair to the second successful subshare repair.

$\vdots$

State k: From the $(k-1)^{\text{th}}$ successful subshare repair to the final k^{th} successful subshare repair.

Let $E[n_i]$ be the expected number of participants contacted in the i^{th} state. Then we can represent the expected time for acquiring all k subshares as

$$\sum_{i=1}^{k} E[n_i].$$

For State 1, the probability that a subshare is repaired is only dependent upon whether the participant is online or offline. This is because, unlike for Algorithm 1, every participant being contacted intersects S_ℓ. Therefore, whichever participant we contact from the $k(r-1)$ participants who is online will be our first "success".

For State 2, the probability is dependent on whether the participant is online and whether the participant repairs a distinct subshare from that which was repaired in the previous state. Of the $k(r-1)$ participants in the set $\mathcal{R}$ only $(k-1)(r-1)$ can contribute subshares distinct from the subshare repaired in the previous state. Therefore, the probability that a randomly chosen participant from $\mathcal{R}$ contributes a distinct share from the previous state is

$$\frac{(k-1)(r-1)}{k(r-1)} = \frac{k-1}{k}.$$

So, for each of the i states we have to account for, the availability p and the reduction in the number of participants which can provide a subshare that has not been repaired in a previous state.

To get $E[n_i]$, we compute

$$\frac{1}{\frac{(k-i+1)(r-1)}{k(r-1)}p},$$

which reduces to

$$\frac{k}{(k-i+1)p}.$$

Therefore, acquiring all k subshares has a total expected complexity of

$$\frac{1}{p}\sum_{i=1}^{k}\frac{k}{k-i+1},$$

which reduces to

$$\frac{1}{p}k\ln k.$$

Since each time we contact a participant we wait time T we include it as a factor as well and therefore we have

$$\text{the expected time} = \frac{1}{p} Tk \ln k.$$

Note that the change between this analysis and that in the previous subsection is that instead of selecting participants from a set of size $\frac{vr}{k}$, we select participants from a set of size $k(r-1)$.

5.3 Algorithm 3: Stored Grouped Participants

Probability Model: In this algorithm, the probability p can be understood as a *permanent fault*. Note that this is a different availability model then used for the previous two algorithms.

Storage: In addition to their share S^j, each participant P_j, stores a set $\mathcal{R}$ consisting of k sets of size $r-1$, where r is the replication number for the design. So we have $\mathcal{R} = \{R_{i_1}, R_{i_2}, ..., R_{i_k}\}$. Each R_{i_j} comprises a list of participants from $\mathcal{P}$ who can perform a repair on the subshare s^i_j.

Algorithm 3 StoredGroupedParticipants$(P_\ell, \mathcal{R})$

```
/* Performs a repair on S_ℓ */
for each subshare s^ℓ_{i_j} in S_ℓ do
    while s^ℓ_{i_j} requires repair and there exist uncontacted participants from R_{i_j} do
        Contact a new participant P_j from R_{i_j} ∈ R and request they send values for s^ℓ_{i_j}
        Wait time T for a response
        if P_j provides s^ℓ_{i_j} then Break to repair next s^ℓ_{i_j}
return S_ℓ = {s^ℓ_{i_1}, s^ℓ_{i_2}, ..., s^ℓ_{i_k}} for P_ℓ
```

This algorithm attempts to perform a repair on a failed node $P_\ell \in \mathcal{P}$ with the knowledge of which participants in the scheme have a subshare which can be used for performing a repair for P_ℓ. Additionally, these participants are grouped into sets where all members of a particular set have the same intersecting point $s^\ell_{i_j}$. The algorithm takes as input the participant P_ℓ, which includes its associated subshare labels (but not the values), as well as the groups of other potential repairing participants, $\mathcal{R} = \{R_{i_1}, R_{i_2}, \ldots, R_{i_k}\}$. If the algorithm is successful, it will produce the k subshare values $S_\ell = \{s^\ell_{i_1}, s^\ell_{i_2}, \ldots, s^\ell_{i_k}\}$ belonging to the participant P_ℓ.

In order to perform a repair, the participant P_ℓ must attempt to repair each $s_{i_j}^\ell$. Beginning with $s_{i_1}^\ell$, the participant P_ℓ first contacts a random participant P_j from a R_{i_1} who has not been previously contacted in the current repair attempt. Since the probability p is a permanent fault, there is no reason to attempt to contact a participant who has previously not responded. Additionally, it is possible to attempt to contact all $k(r-1)$ participants in $\mathcal{R}$ and not succeed in repairing S_ℓ. Participant P_ℓ makes a request for $s_{i_1}^\ell$. Since the participant P_j may or may not be online, P_ℓ waits time T for a response. After acquiring a subshare value, the participant P_ℓ proceeds to the next remaining $s_{i_j}^\ell$ requiring repair, and repeats the process for all k subshares $S_{i_1}^\ell$.

Remark 5.3. If a repair set exists, this algorithm will find it. We defined the probability a repair set exists with Theorem 4.5. The theorem states that a repair set exists with probability $(1-(1-p)^{r-1})^k$.

5.3.1 Complexity Analysis

Recall that, for this algorithm, the probability model is the permanent fault model. Within the permanent fault model, we can consider a worst case execution, as once a participant is unavailable with probability p, they remain unavailable. It therefore has a termination point that does not exist with the transient fault model.

Worst Case:

In the worst case where a successful repair occurs, the first $(r-2)$ participants contacted for each repair are unavailable and the $(r-1)^{\text{st}}$ participant contacted is available, for each of the k subshares to be repaired. In this case the algorithm will run in time

$$k(r-1)T.$$

Note that repairing each subshare can be done independently.

Average Case:

This is the average case complexity for a successful repair. For the average case, we need to account for which player P_i provides a subshare s required for repair. We can do this by evaluating the sum of the probabilities a successful subshare s is provided by the i^{th} participant for $1 \leq i \leq r-1$. For every subshare s required by P_ℓ there are $(r-1)$ other participants who can send s. More specifically, for any subshare s it could be the case that,

1. P_1 sends s or,
2. P_1 does not send s, but P_2 sends s or,
3. P_1 does not send s and P_2 does not send s, but P_3 sends s or

$\vdots$

$r-1$. P_1 does not send s and $\ldots$ and P_{r-2} does not send s, but P_{r-1} sends s.

For each of the above cases, the i^{th} case corresponds to participant P_i providing the subshare s. The probability of any of the above cases is dependent on the availability of the i^{th} participant and the participants which preceded it being unavailable.

Assume following the algorithm that participant P_1 was contacted to repair s. After contacting P_1, P_ℓ will wait time T for a response. P_1 will respond with probability p and therefore the complexity associated with this first case is pT. Alternatively, P_1 may not respond with probability $1-p$. If P_1 does not respond after time T, then P_ℓ moves on to P_2 and waits time T for a response from P_2. P_2 also responds with probability p. The complexity in the case where P_1 does not respond and P_2 does respond will therefore be $2T(1-p)p$. To determine the overall complexity we need to compute the probability that P_1 sends s, or P_1 does not respond but P_2 sends s, or P_1 and P_2 do not respond but P_3 sends s, or P_{r-1} sends s and all preceding participants did not respond. For any subshare, this complexity can be written as:

$$pT + 2T(1-p)p + 3T(1-p)^2p + ... + (r-1)T(1-p)^{r-2}p.$$

The above sum can be rewritten as:

$$\sum_{i=1}^{r-1}(1-p)^{i-1}piT = pT\sum_{i=1}^{r-1}(1-p)^{i-1}i,$$

where T is the time we wait, and p is the probability that a participant is available to perform a repair. This summation is an arithmetico-geometric sequence that can be easily simplified using a tool such as Wolfram Alpha[4] to produce the following result,

$$T\left(\frac{(p(r-1)+1)(1-p)^r + p - 1}{(p-1)p}\right).$$

[4]https://www.wolframalpha.com, use "Simplify Sum [k (1 - p)^{(k - 1)}, {k, 1, r-1}]".

The above equation can be simplified further by removing a factor of $(1-p)$ to produce,

$$T\left(\frac{1-(p(r-1)+1)(1-p)^{r-1}}{p}\right).$$

This equation will be the same for any subshare s being repaired. Therefore, the average case where k subshares are repaired has the expected complexity

$$kT\left[\frac{1-(p(r-1)+1)(1-p)^{r-1}}{p}\right].$$

5.4 Algorithm 4: Generating Participants

Probability Model: In this algorithm, the probability p can be understood as a permanent fault.

Storage: In addition to their share S_ℓ, each participant stores a set $\mathcal{B}$ consisting of the base blocks for the underlying distribution design. The set $\mathcal{B}$ is smaller than the set $\mathcal{R}$ from Algorithm 3. Therefore, for this algorithm, we are reducing the required storage for each participant.

For this algorithm, instead of storing the grouped intersecting participants $\mathcal{R}$ as in Algorithm 3, each participant stores the base blocks for the underlying distribution design. The desired intersecting blocks and corresponding participants can then be generated from the stored base blocks for each subshare requiring repair.

Remark 5.4. You may recall from Section 3.3 that one of the advantages of the combinatorial based repairable threshold schemes is that they do not require additional computation. The computation we are adding here does not remove this advantage. We are only computing over the indices corresponding the blocks of the design which are quite small. In contrast to this, other methods require performing computations on the shares, The shares can be quite large and so we still retain the advantage by computing over only small values.

5.4.1 Generating blocks for specific subshares

From a Single Base Block

Assume for now that the underlying distribution design has only one base block, $B = x_1x_2\ldots x_k$. The distribution design has points belonging to an abelian group G of size v.

Algorithm 4 GeneratingIntersectingParticipants($P_\ell, \mathcal{B}$)

```
/* Performs a repair on S_ℓ */
for each subshare in S_ℓ do
    while s^ℓ_{i_j} requires repair and there exist uncontacted participants do
        Generate a new intersecting block B using the stored base blocks
        Contact participant P_j corresponding to the generated block B and request they send the value for s^ℓ_{i_j}
        Wait time T for a response
        if P_j provides s^ℓ_{i_j} then Break to repair next s^ℓ_{i_j}
return S_ℓ = {s^ℓ_{i_1}, s^ℓ_{i_2}, ... s^ℓ_{i_k}} for P_ℓ
```

Let the blocks for the design $\mathcal{B} = \{B_0, B_1, \ldots, B_{b-1}\}$, where b is the number of blocks. A single base block can generate at most $|G|$ blocks belonging to the design, including itself. Label the blocks such that each block B_j would be generated using the base block B and computing $x_p + j \pmod v$ for each point $x_p \in B$.

Example 5.5. Let the distribution design be a $(7, 3, 1)$-BIBD with the base block $\mathcal{B} = \{013\}$ and blocks labelled $\{B_0, B_1, B_2, B_3, B_4, B_5, B_6\}$ as per the description above. The resulting blocks in the design are:

$$\{013, 124, 235, 346, 450, 561, 602\}.$$

These blocks correspond to players

$$\{P_0, P_1, P_2, P_3, P_4, P_5, P_6\},$$

respectively. Going back to the earlier description of generating labelled blocks, we can consider the block for player P_3 as B_3 in $\mathcal{B}$. The block B_3 would be generated by computing $\{0 + 3 \pmod 7, 1 + 3 \pmod 7, 3 + 3 \pmod 7\}$, which is block 346.

In order to determine which participants to contact for a repair, the player P_ℓ needs to sequentially generate intersecting participants for each subshare as needed. Using the points from the base block B we can generate a block B_j such that it contains the value corresponding to a subshare by computing $j = (s_i - x_p) \pmod v$. The value s_i is the point in the design for the subshare that participant P_ℓ is attempting to repair and x_p is a point from the base block B. The resulting value from the computation will correspond to the participant P_j. Iterating over each $x_p \in B$ will result in all the blocks which contain the relevant s_i.

Example 5.6. Let the distribution design be a $(7, 3, 1)$-BIBD with the base block $\mathcal{B} = \{013\}$ from Example 5.5. Let the participant requiring a repair be P_4 with the corresponding block 450.

For each subshare that P_4 has, there are two other participants in the scheme which have the same subshare. We generate participants who can repair subshare $s_i = 4$ as follows:

1. Begin with the first point $x_0 = 0$ from $B = 013$.
2. Compute $4 - 0 \pmod 7 = 4$. This corresponds to the participant P_4 who requires a repair. If we take each point from the block design and increment it by 4 (mod 7), it would produce 450 (P_4's block) and it is therefore not a member of $\mathcal{R}$.
3. Take the next point $x_1 = 1$ from $B = 013$.
4. Compute $4 - 1 \pmod 7 = 3$. This would give the block 346 held by participant P_3.
5. Take the final point $x_2 = 3$ from $B = 013$.
6. Compute $4 - 3 \pmod 7 = 1$. This would give the block 124 held by participant P_1.

The participants that P_4 can contact in order to attempt to repair the value of $s_i = 4$ are therefore P_1 and P_3. These would correspond to one of the $R_i \in \mathcal{R}$.

Now we generate participants who can repair $s_i = 5$ for participant P_4 as follows:

1. Begin with the first point $x_0 = 0$ from $B = 013$.
2. Compute $5 - 0 \pmod 7 = 5$. This would give the block 561 held by participant P_5.
3. Take the next point $x_1 = 1$ from $B = 013$.
4. Compute $5 - 1 \pmod 7 = 4$. This corresponds to the participant P_4 who requires a repair. This would give the block 450 (P_4's block) and it is therefore not a member of $\mathcal{R}$.
5. Take the final point $x_2 = 3$ from $B = 013$.
6. Compute $5 - 3 \pmod 7 = 2$. This would give the block 235 held by participant P_2.

The participants that participant P_4 can contact in order to attempt to repair the value of $s_i = 5$ are therefore P_5 and P_2.

Generate participants who can repair $s_i = 0$ for participant P_4 as follows:

1. Begin with the first point $x_0 = 0$ from $B = 013$.
2. Compute $0 - 0 \pmod 7 = 0$. This would give the block 013 held by participant P_0.
3. Take the next point $x_1 = 1$ from $B = 013$.
4. Compute $0 - 1 \pmod 7 = 6$. This would give the block 602 held by participant P_6.
5. Take the final point $x_2 = 3$ from $B = 013$.
6. Compute $0 - 3 \pmod 7 = 4$. This would give the block 450 (P_4's block) and it is therefore not a member of $\mathcal{R}$.

The participants that participant P_4 can contact in order to attempt to repair the value of $s_i = 0$ are therefore P_0 and P_6. Considering all the generated sets for block 450 results in the set of participants

$$\mathcal{R} = \{\{P_1, P_3\}, \{P_5, P_2\}, \{P_0, P_6\}\}.$$

Algorithm 5 RepairBlocksFromSingleBaseBlock$(P_\ell, \mathcal{B})$

```
/* Generates all intersecting blocks for P_ℓ using B */
/* Assume base (v, k, 1)-BIBD */
for each point x_p in B do
    for each point p in P_ℓ do
        Compute x = p − x_p (mod v)
        if x ≠ ℓ then Construct the next intersecting block as {p_1 + x, p_2 + x, p_3 + x}
return R = {R_1, R_2, ..., R_k} for P_ℓ
```

In general, we can generate all the relevant intersecting blocks for a participant P_ℓ where there is one base block in the underlying design using Algorithm 5.

From Multiple Base Blocks

In the case of a distribution design which has more than one base block, each base block will generate $|G|$ blocks for the design, with the noted exception of base blocks with short orbits (see Definition 2.33). For multiple base blocks, we modify the labeling technique to indicate which base block generated the block. So, where before we labelled a block B_j we will now label it as $B_{i,j}$. The value i indicates the block is generated from the i^{th} base block from the set $\mathcal{B}$. The value j is the same value as for our original labeling scheme except that it is reset back to zero for each base block.

Example 5.7. Let $\mathcal{B} = \{B_0, B_1\}$ be the set of base blocks for a distribution design over a group G, $|G| = n$. Following the original ordering with the additional labeling, and assuming no block results in a short orbit, then the resulting blocks would be:

$$\{B_{0,0}, B_{0,1}, \ldots B_{0,n}, B_{1,0}, B_{1,1} \ldots B_{1,n}\}.$$

The labelled blocks are ordered such that each block $B_{i,j}$ is in the order it would be generated using the base block B_i and computing $x_p + j \pmod{v}$ for each $x_p \in B_i$.

Example 5.8. Let the distribution design be a $(13, 3, 1)$-BIBD with base blocks $\mathcal{B} = \{014, 027\}$. Let participant $P_{1,12}$ have the share corresponding to block $\{12, 0, 3\}$ in the design. For each subshare that $P_{1,12}$ has, there are five other participants in the scheme which have the same subshare.

We generate participants who can repair $s_i = 12$ for participant P_{12} as follows:

From first base block 014

1. Begin with the first point $x_0 = 0$ from 014.
2. Compute $12 - 0 \pmod{13} = 12$. This corresponds to the participant $P_{0,12}$ who requires repair and would give the block $\{12, 0, 3\}$. It is therefore not part of this $\mathcal{R}$.
3. Take the next point $x_1 = 1$ from 014.
4. Compute $12 - 1 \pmod{13} = 11$. This would give the block $\{11, 12, 2\}$ held by participant $P_{0,11}$.
5. Take the final point from the first base block $x_2 = 4$.
6. Compute $12 - 4 \pmod{13} = 8$. This would give the block $\{8, 9, 12\}$ held by participant $P_{0,8}$.

From second base block 027

1. Begin with the first point $x_0 = 0$ from 027.
2. Compute $12 - 0 \pmod{13} = 12$. This would give the block $\{12, 1, 6\}$ held by participant $P_{1,12}$.
3. Take the next point $x_1 = 2$ from 027.
4. Compute $12 - 2 \ (13) = 10$. This would give the block $\{10, 12, 4\}$ held by $P_{1,10}$.
5. Take the final point from the second base block $x_2 = 7$.

6. Compute $12 - 7 \pmod{13} = 5$. This would give the block $\{5, 7, 12\}$ held by participant $P_{1,5}$.

The participants that participant $P_{0,12}$ can contact in order to attempt to repair the value of $s_i = 12$ are therefore:

$$R_0 = \{P_{0,11}, P_{0,8}, P_{1,12}, P_{1,10}, P_{1,5}\}.$$

The same process as shown for $s_i = 12$ can be applied to each of the other s_i which require a repair.

Algorithm 6 RepairBlocksFromMultipleBaseBlocks($P_\ell, \mathcal{B}$)

```
/* Generates all relevant blocks for P_ℓ using B */
/* Assume base (v, k, 1)-BIBD */
for each base block B in B do
    for each point x_p in B do
        for each point p in P_ℓ do
            Compute x = p − x_p (mod v)
            if x ≠ ℓ then Construct the next repair block as {p_1 + x, p_2 + x, ..., p_k + x}
return R = {R_1, R_2, ..., R_k} for P_ℓ
```

In general, we can generate all the relevant intersecting blocks for a participant P_ℓ where there are multiple base blocks in the underlying distribution design using Algorithm 6.

5.4.2 Complexity Analysis

Note that for the complexity analysis, this algorithm essentially reduces down to Algorithm 3. It performs an additional calculation in order to determine who to contact instead of storing these participant labels. The motivation here is therefore not to reduce complexity, but to reduce the storage requirements for each participant.

Worst Case:

In the worst case, it will be necessary to generate and contact all $k(r-1)$ possible participants before the repair is complete. This would therefore require time $Tk(r-1)$.

Average Case:

For our average case analysis it follows from Algorithm 3 that it will take time

$$kT\left[\frac{1-(p(r-1)+1)(1-p)^{r-1}}{p}\right].$$

Chapter 6

Beyond 2-Designs

In the schemes discussed so far, including the work of Stinson and Wei [19], all of the distribution designs produced repairable schemes such that, during a repair, each participant is able to provide at most one subshare to help repair the failed share. Each of these distribution designs was a $t-(v,k,\lambda)$ design (see Definition 6.1) with $t=2$ and $\lambda=1$. In the following, we will consider t-designs with $t \geq 3$, which will result in cases where participants can provide more than one share during a repair.

The following defines t-designs where $t>2$, outlines their use as distribution designs and compares the resulting repairable threshold schemes to the resulting schemes from distribution designs with $t=2$.

6.1 t-Designs

Definition 6.1. A $t-(v,k,\lambda)$ *design* is a design where:

1. $|X|=v$,
2. Each block is of size k,
3. Every set of t points from the set X occurs in exactly λ blocks.

Definition 6.2. A $3-(v,4,1)$ design is a *Steiner quadruple system* of order v, denoted $SQS(v)$. For all $SQS(v)$, $v \equiv 2,4 \pmod 6$.

Theorem 6.3. *[17, Thm. 9.16] If there exists an $SQS(v)$ then there exists an $SQS(2v)$.*

Theorem 6.4. *[17, Thm. 9.18] There exists an $SQS(2^n)$ for all integers $n \geq 3$.*

There are a number of Steiner quadruple systems for which we can explicitly write out the blocks. The smallest of these are the $SQS(8)$ and $SQS(10)$. These will serve as examples for using t-designs ($t \geq 3$) as distribution designs for repairable threshold schemes.

Example 6.5. For an example with $t = 3$, we can consider an $SQS(8)$, which is a $3-(8,4,1)$ design. Let $X = \{1,2,3,4,5,6,7,8\}$. Then the corresponding blocks are

$$\begin{array}{cc} 1234 & 5678 \\ 1256 & 3478 \\ 1278 & 3456 \\ 1357 & 2468 \\ 1368 & 2457 \\ 1458 & 2367 \\ 1467 & 2358. \end{array}$$

The focus going forward will be on 3-designs and comparing them to 2-designs throughout this chapter. However, first we can make note of examples for both $t = 3$ and $t = 4$.

Example 6.6. The following is a $4-(11,5,1)$ design.

2 5 7 8 10	4 5 8 10 11	1 2 7 8 11	1 4 6 8 11	1 3 4 5 10
1 3 6 9 10	1 4 5 7 8	1 5 8 9 11	2 3 4 8 11	1 2 4 5 11
2 5 7 9 11	1 2 5 9 10	1 3 5 6 11	2 4 5 8 9	5 6 7 8 9
1 3 4 7 11	1 4 6 7 10	4 7 8 9 11	1 2 3 7 10	2 3 5 6 9
2 8 9 10 11	2 4 7 10 11	1 4 5 6 9	2 3 4 9 10	2 6 7 9 10
3 4 6 10 11	1 2 4 7 9	1 5 7 10 11	1 2 3 4 6	3 5 6 7 10
3 4 5 9 11	1 3 8 10 11	1 2 4 8 10	3 4 7 8 10	1 3 5 7 9
1 2 3 9 11	1 7 8 9 10	4 5 7 9 10	1 3 6 7 8	2 5 6 8 11

3 4 5 6 8	1 6 7 9 11	4 5 6 7 11	3 5 7 8 11	1 2 6 8 9
1 3 4 8 9	6 7 8 10 11	2 3 4 5 7	3 5 8 9 10	3 6 8 9 11
1 2 6 10 11	3 4 6 7 9	4 6 8 9 10	2 4 5 6 10	
2 3 6 8 10	2 4 6 9 11	2 3 6 7 11	1 2 5 6 7	
1 2 3 5 8	5 6 9 10 11	1 4 9 10 11	1 5 6 8 10	
3 7 9 10 11	2 3 5 10 11	2 3 7 8 9	2 4 6 7 8	

Example 6.7. The following is a $5-(12,6,1)$ design.

1 2 5 7 8 10	1 2 7 8 11 12	1 3 4 5 10 12	1 3 4 6 7 9
1 3 6 9 10 12	1 5 8 9 11 12	4 5 6 10 11 12	1 2 4 6 9 11
1 2 5 7 9 11	1 3 5 6 11 12	1 5 6 7 8 9	1 5 6 9 10 11
1 3 4 7 11 12	2 4 6 7 11 12	2 4 6 8 10 11	2 3 4 7 8 12
1 2 8 9 10 11	1 4 5 6 9 12	1 2 6 7 9 10	1 4 5 6 7 11
2 4 6 9 10 12	1 5 7 10 11 12	2 5 6 8 9 10	2 3 4 6 8 9
2 3 5 6 10 12	2 4 5 6 8 12	2 3 6 9 10 11	1 4 6 8 9 10
5 7 8 9 10 11	2 5 8 10 11 12	1 2 5 6 8 11	3 4 8 9 10 11
2 5 7 8 9 12	1 4 6 8 11 12	1 3 4 5 6 8	2 3 4 6 7 10
3 4 7 9 10 12	2 3 6 7 9 12	1 3 4 8 9 12	1 2 3 7 8 9
1 2 5 9 10 12	2 3 6 8 11 12	2 3 4 5 9 12	4 5 6 7 8 10
1 4 6 7 10 12	2 7 9 10 11 12	4 5 6 8 9 11	3 4 6 8 10 12
3 5 6 8 9 12	3 4 6 9 11 12	2 6 7 8 9 11	1 2 4 5 6 10
3 4 5 8 11 12	2 3 4 10 11 12	2 5 6 7 10 11	1 2 5 6 7 12
4 6 7 8 9 12	4 6 7 9 10 11	3 5 6 8 10 11	2 4 5 6 7 9
1 7 8 9 10 12	1 3 6 7 8 12	1 6 7 8 10 11	2 3 5 7 9 10

1 2 6 8 9 12	1 2 3 5 10 11	2 3 5 6 7 8	1 2 4 8 10 12
1 3 6 8 9 11	2 3 4 5 6 11	3 4 5 6 7 12	2 4 7 8 9 10
3 5 6 7 9 11	5 6 7 8 11 12	3 4 5 7 8 9	2 4 5 7 8 11
1 2 4 6 7 8	2 5 6 9 11 12	1 2 4 5 11 12	2 4 5 7 10 12
1 5 6 8 10 12	2 3 7 8 10 11	2 3 4 5 8 10	2 3 5 7 11 12
3 6 7 10 11 12	1 6 7 9 11 12	1 3 4 7 8 10	1 4 7 8 9 11
2 6 7 8 10 12	2 4 8 9 11 12	1 2 3 4 6 12	1 3 8 10 11 12
1 3 5 8 9 10	1 3 7 9 10 11	1 2 3 4 9 10	1 2 4 7 9 12
3 4 5 6 9 10	4 5 7 9 11 12	1 2 3 7 10 12	1 2 4 7 10 11
5 6 7 9 10 12	1 2 3 5 8 12	3 5 9 10 11 12	1 4 5 7 8 12
1 3 5 7 8 11	1 2 3 6 8 10	3 5 7 8 10 12	1 4 5 8 10 11
1 4 9 10 11 12	3 4 5 7 10 11	6 8 9 10 11 12	1 2 3 9 11 12
1 2 3 6 7 11	1 2 6 10 11 12	4 5 8 9 10 12	1 3 4 5 9 11
3 6 7 8 9 10	1 3 5 7 9 12	1 2 4 5 8 9	3 7 8 9 11 12
2 3 5 8 9 11	3 4 6 7 8 11	2 3 8 9 10 12	2 4 5 9 10 11
4 7 8 10 11 12	1 3 5 6 7 10	1 2 3 4 8 11	1 3 4 6 10 11
1 2 3 4 5 7	1 2 3 5 6 9	1 4 5 7 9 10	2 3 4 7 9 11

Theorem 6.8. *[17, Thm. 9.4] The* replication number r_i *represents the number of times any set of points from X of size i is repeated in a $t-(v,k,1)$ design. It is known that*

$$r_i = \frac{\lambda\binom{v-i}{t-i}}{\binom{k-i}{t-i}}, \text{ for } 1 \leq i \leq t.$$

We previously computed the replication number $r = r_1$ using the equation from Theorem 2.14. This determined the number of times a single point occurred in the design. We can compute this same value using Theorem 6.8 with $i = 1$.

Example 6.9. The replication number r_1 for the $2-(7,3,1)$-BIBD is

$$r_1 = \frac{1\binom{7-1}{2-1}}{\binom{3-1}{2-1}} = 3.$$

In the specific case of the replication number for sets of size $t-1$, we can use another formula that reduces to the replication number $r_1 = r$ from Theorem 2.14.

Theorem 6.10. *The replication number r_{t-1} represents the number of times any $t-1$ points from the set X are repeated in the design.*

$$r_{t-1} = \frac{\lambda(v-(t-1))}{k-(t-1)}.$$

Proof. Consider Theorem 6.8 with $i = t-1$.

$$r_{t-1} = \frac{\lambda\binom{v-(t-1)}{t-(t-1)}}{\binom{k-(t-1)}{t-(t-1)}} = \frac{\lambda\frac{(v-t+1)!}{(v-t)!}}{\frac{(k-t+1)!}{(k-t)!}} = \frac{\lambda(v-t+1)}{k-t+1} = \frac{\lambda(v-(t-1))}{k-(t-1)}$$

□

Example 6.11. For a $2-(9,3,1)$-BIBD we can determine r_{t-1} as:

$$r_{2-1} = \frac{\lambda(v-(2-1))}{k-(2-1)},$$

which reduces to the original equation for the replication number r shown in Theorem 2.14

$$r = \frac{\lambda(v-1)}{k-1} = 4.$$

Theorem 6.12. *The number of blocks in the design is:*

$$b = \frac{\binom{v}{t}}{\binom{k}{t}} = \frac{vr_1}{k}.$$

6.2 Distribution Designs

We can use the t-designs discussed in the previous section as distribution designs in the same way we used balanced incomplete block designs. This section discusses some valid distribution designs, with $t \geq 2$, and the resulting repairable threshold schemes.

Theorem 6.13. *The repairing degree d for a $t-(v,k,1)$ design is:*

$$d = \left\lceil \frac{k}{t-1} \right\rceil.$$

Proof. We use the fact that two blocks contain at most $t-1$ common points.

Recall d is the minimum number of participants required to perform a repair. It is optimal if each participant who provides subshares provides the maximum they can, namely $t-1$.

If the maximum number of subshares provided by each participant is $t-1$ and k is a multiple of $t-1$, then we can write k as:

$$k = m(t-1), \text{ for some integer } m.$$

then the number of sufficient participants is:

$$d = \frac{m(t-1)}{t-1} = m.$$

If the maximum number of subshares are provided by each participant and k is not a multiple of $t-1$, then we can write k as:

$$k = m(t-1) + n, \text{ for some integers, } m \text{ and } n, \text{ where } 0 < n < t-1.$$

Then, the participant with the failed share could contact m other participants who would provide $t-1$ subshares and then contact an additional participant who would provide n subshares. So, the repairing degree in this case is:

$$d = m + 1.$$

Since we have $m \leq \frac{k}{t-1} < m+1$, the repairing degree in general is:

$$d = \left\lceil \frac{k}{t-1} \right\rceil.$$

□

Example 6.14. Consider the $SQS(8)$ from Example 6.5. The parameters of the design are:

$$b = 14,$$

$$r_1 = 7,$$

$$r_2 = 3.$$

Consider this example as a (τ, ℓ_1, ℓ_2) distribution design (dd) such that the union of any τ blocks produces at least ℓ_1 points and the union of any $\tau - 1$ blocks produces at most ℓ_2 points.

For this example, the union of any $\tau = 2$ blocks is going to have at least six points as any two blocks have at most $t - 1 = 2$ points in common. The union of any $\tau - 1 = 1$ blocks has at most four points. Therefore, following the definition for using a ramp scheme as a base scheme, we can use the $SQS(8)$ as a $(2, 4, 6)$-distribution design. If we use a $(4, 6, 8)$-Ramp Scheme as our base scheme with this $(2, 4, 6)$-distribution design, we can produce a $(2, 2, 14)$-Repairable threshold scheme.

See Theorem 6.13 for elaboration on repairing degree d.

Example 6.15. Consider an $SQS(10)$ and let $X = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Then the corresponding blocks are

1245	1237	1358
2356	2348	2469
3467	3459	3570
4578	4560	1468
5689	1567	2579
6790	2678	3680
1780	3789	1479
1289	4890	2580
2390	1590	1369
1340	1260	2470.

The other parameters of the design are:

$b = 30,$

$r_1 = 12,$

$r_2 = 4.$

Consider this example as a (τ, ℓ_1, ℓ_2) distribution design as in Example 6.14. Again, the union of any two blocks will contain at least six points since any two blocks has at most two points in common. Any one block will contain at most four points. This example can therefore also serve as a $(2, 4, 6)$-distribution design. If we use a $(4, 6, 10)$-Ramp Scheme as our base scheme we can produce a $(2, 2, 30)$-Repairable threshold scheme.

See Theorem 6.13 for elaboration on repairing degree d.

Theorem 6.16. *An $SQS(v)$ can be used as a $(2, 4, 6)$-distribution design to produce a $(2, 2, b)$-repairable threshold scheme, where $b = \frac{vr}{k}$ is the number of blocks in the $SQS(v)$.*

Proof. From Definition 3.11, we know that a design is a $(2, 4, 6)$-distribution design if the union of any two blocks contains at least six points, any one block contains at most four points, and $6 - 4 \geq 1$.

For an $SQS(v)$, any two blocks have at most two points in common because any three points occurs in the design exactly once by definition. Therefore, the union of any two blocks has at least $4 + 4 - (3 - 1) = 6$ points.

For an $SQS(v)$, any one block is of size four. Therefore, the union of any one block contains at most four points. Since, $6 - 4 = 2 \geq 1$, a $SQS(v)$ is a $(2, 4, 6)$-distribution design.

A $(2, 4, 6)$-distribution design produces a $(2, 2, b)$-repairable threshold scheme by Theorem 3.13 and Theorem 6.13. □

Now we show how to construct a repairable threshold scheme for $\tau = 3$ from t-designs.

Theorem 6.17. *A $t - (v, k, 1)$ design can be used as a $(3, 2k, 3k - 3(t - 1))$ distribution design to produce a $(3, \lceil \frac{k}{t-1} \rceil, b)$-repairable threshold scheme if $k \geq 3t - 2$, where b is the number of blocks in the $t - (v, k, 1)$ design.*

Proof. From Definition 3.11, we know that a design is $(3, 2k, k + 2(k - t + 1))$ distribution design if the union of any three blocks contains at least $k + 2(k - t + 1))$ points, the union of any two blocks contains at most $2k$ points, and $3k - 3(t - 1) - 2k \geq 1$.

For any $t-(v,k,1)$ design, any two blocks have at most $t-1$ points in common. If each pair has a distinct set of $t-1$ points in common, then there are the maximum number of points, namely $3(t-1)$ in common among the three of them. Therefore there are at least $3k-3(t-1)$ distinct points in the union of any three blocks.

For a $t-(v,k,1)$ design, two different blocks from the design may have zero points in common. Each block is of size k and if no points are repeated there can be up to $3k$ points in the union of any three blocks.

The remaining requirement for a $t-(v,k,1)$ design to satisfy the requirements of the specified distribution design is $3k-3(t-1)-2k \geq 1$. Consider the following:

$$3k-3(t-1)-2k \geq 1$$

$$k-3(t-1) \geq 1$$

$$k \geq 1+3(t-1)$$

$$k \geq 3t-2.$$

Therefore, a $t-(v,k,1)$ design can be used as a $(3, 2k, 3k-3(t-1))$ distribution design if $k \geq 3t-2$.

A $(3, 2k, 3k-3(t-1))$-distribution design produces a $(3, \lceil \frac{k}{t-1} \rceil, b)$-repairable threshold scheme by Theorem 3.13 and Theorem 6.13. □

In the following we present two families of designs which satisfy the requirements of Theorem 6.17.

Definition 6.18. An *inversive plane* is a $3-(q^2, q+1, 1)$ design where q is a prime number.

Theorem 6.19. *[17, Thm. 9.27] For all prime powers q, there exists a $3-(q^2, q+1, 1)$ design.*

Theorem 6.20. *An inversive plane can be used as a $(3, 2(q+1), 3q-3)$-distribution design to produce a $(3, \lceil \frac{q+1}{2} \rceil, b)$-repairable threshold scheme if $q \geq 6$ is a prime power.*

Proof. By Theorem 6.17, a $(3, 2q+2), 3(q+1)-6)$-distribution design can be used to produce a $(3, \lceil \frac{q+1}{2} \rceil, b)$-repairable threshold scheme if $k \geq 3t-2$. We have that $k = q+1$ and $t = 3$. Substituting those values appropriately into the inequality produces:

$$q+1 \geq 3 \cdot 3 - 2$$

$$q \geq 9 - 2 - 1$$
$$q \geq 6$$

□

Recall that projective planes were used in Stinson and Wei [19] to achieve better thresholds and smaller repair sets of size d. Inversive planes achieve better repairing degree d, in that they only require at least $d = \lceil \frac{k}{t-1} \rceil$ participants to perform a repair, whereas projective planes require at least $d = k$ participants to perform a repair. Unfortunately, unlike with projective planes where we can achieve many values for the threshold τ, in the case of inversive planes, we can only achieve $\tau = 3$.

Definition 6.21. A *spherical geometry* is a $3 - (q^n + 1, q + 1, 1)$ design where q is a prime number and $n \geq 2$.

As with inversive planes, the more general spherical geometry exists for all q, where q is a prime power.

Theorem 6.22. *[5, Thm. 5.11] Known infinite families of $t - (v, k, 1)$ designs include $3 - (q^n + 1, q + 1, 1)$ designs where q is a prime number and $n \geq 2$.*

Theorem 6.23. *A spherical geometry can be used as a $(3, 2(q+1), 3(q+1)-6)$-distribution design to produce a $(3, \lceil \frac{q+1}{2} \rceil, b)$-repairable threshold scheme if $q \geq 6$.*

Proof. This follows using the same reasoning as Theorem 6.20. □

Theorem 6.24. *A $t - (v, k, 1)$ design can be used as a $(\tau, (\tau - 1)k, \tau k - \binom{\tau}{2}(t - 1))$ distribution design to produce a $(\tau, \lceil \frac{k}{t-1} \rceil, b)$-repairable threshold scheme if $k \geq \binom{\tau}{2}(t-1)+1$, where b is the number of blocks in the $t - (v, k, 1)$ design.*

Proof. From Definition 3.11, we know that a design is a $(\tau, (\tau - 1)k, \tau k - \binom{\tau}{2}(t - 1))$ distribution design if the union of any τ blocks contains at least $\tau k - \binom{\tau}{2}(t-1)$ points, the union of any $\tau - 1$ blocks contains at most $(\tau - 1)k$ points, and $\tau k - \binom{\tau}{2}(t-1) - (\tau - 1)k \geq 1$.

For any $t - (v, k, 1)$ design, any two blocks have at most $t - 1$ points in common between them. If, for the union of τ blocks, each pair of blocks has a distinct pair of points in common, then the maximum number of points in common among the τ blocks is $\binom{\tau}{2}(t - 1)$. Therefore, there are at least $\tau k - \binom{\tau}{2}(t - 1)$ distinct points in the union of any τ blocks.

For any $t-(v,k,1)$ design, τ different blocks from the design may have zero points in common. Each block is of size k and if no points are repeated there can be up to τk points in the union of any τ blocks. Therefore, the union of any $\tau-1$ blocks will have at most $(\tau-1)k$ distinct points.

The remaining requirement for a $t-(v,k,1)$ design to satisfy the requirements of the specified distribution design is $\tau k - \binom{\tau}{2}(t-1) - (\tau-1)k \geq 1$. Consider the following:

$$\tau k - \binom{\tau}{2}(t-1) - (\tau-1)k \geq 1$$

$$k - \binom{\tau}{2}(t-1) \geq 1$$

$$k \geq 1 + \binom{\tau}{2}(t-1).$$

Therefore, a $t-(v,k,1)$ design can be used as a $(\tau,(\tau-1)k,\tau k-\binom{\tau}{2}(t-1))$ distribution design if $k \geq \binom{\tau}{2}(t-1)+1$.

A $(\tau,(\tau-1)k,\tau k-\binom{\tau}{2}(t-1))$-distribution design produces a $(\tau,\lceil\frac{k}{t-1}\rceil,b)$-repairable threshold scheme by Theorem 3.13 and Theorem 6.13. □

For Theorem 6.24, there are examples of designs which satisfy the requirements for different values of τ and different values of t. However, there are some restrictions. We have designs for $\tau=4$ and $t=2$ in the family of BIBDs. We also have designs for $\tau=3$ and $t=3$ in the family of inversive planes and spherical geometries. Unfortunately, there are no known designs that satisfy the restriction of $k \geq \binom{\tau}{2}(t-1)+1$ when both $\tau>3$ and $t>2$.

Chapter 7

Repair Sets for t-designs

For distribution designs that are $2-(v,k,1)$ designs, every successful repair set consists of k participants, each of which provided a single point. For distribution designs using $t-(v,k,1)$ designs, we have repair sets of size at most k and at least $\lceil \frac{k}{t-1} \rceil$. If we consider specifically the case of Steiner quadruple systems, the repair set can be of size two, three, or four. The repair sets for an $SQS(v)$ can be any of the following forms:

1. "pair, pair"
2. "pair, point, point"
3. "point, point, point, point"

The first form "pair, pair" is the smallest possible repair set; which has size $\frac{4}{3-1} = 2$. The largest possible repair set corresponds to the last form, "point, point, point, point" which has size $k = 4$.

Example 7.1. Consider the repairable secret sharing scheme from Example 6.14

Let P_ℓ require a repair for their share 1256. Let us follow a procedure along the lines of Algorithm 2, where we contact random participants from $\mathcal{R}$.

1. Assume P_ℓ contacts the participant with 1234. This provides the pair 12.
2. The next participant, P_j, may provide the required pair 56 and the repair would be complete with a repair set of size two Alternatively, P_j might have a pair such as 16, 25, 15, or 26. This would give us only one new point, 5 or 6.

3. If the repair was not completed at the previous stage, P_ℓ will contact an additional participant P_k to receive the remaining subshare (either 5 or 6 depending on what P_j provided). If P_j provided the pair 15 and P_k provided 16, then the repair set is of size three.

Definition 7.2. A repair set of size n is *minimal* if no proper subset of its members form a repair set of size $m < n$.

Example 7.3. Continue with the repairable secret sharing scheme from Example 6.14

Let P_ℓ require a repair for their share 1256.

1. Assume P_ℓ first receives the pair 12 from participant P_i.
2. Assume P_j provides the pair 16. This would give us only one new point, 6.
3. P_ℓ will now contact participant P_k. The participant P_k could provide subshares 25, 15, or 26. If participant P_k provides the subshare 25, then from the set $R = \{P_i, P_j, P_k\}$ there exists a repair subset $\{P_j, P_k\}$ which has size 2. Therefore, this repair set R would not be minimal. However, if P_k provides the subshare 15, then there is no proper subset which also is a repair set and therefore the repair set is minimal.

7.1 Existence of Repair Sets

In this section we will determine the probability that a repair set exists when using distribution designs with $t \geq 3$ using techniques from network reliability. These techniques, as applied to network reliability, can be found in Section 1.2 of *The Combinatorics of Network Reliability* [4]. Specifically, the method that we will be using from network reliability is to employ the use of *cutsets* in calculating the reliability. Although we were able to compute the existence equations for $2 - (v, k, 1)$ designs without the use of this methodology, we can apply the use of cutsets to the analysis of 2-designs as well. As before, a repair set exists when there is at least one subset of participants which can perform a repair for a given participant who has lost their share. Recall that each participant is available with probability p and unavailable with probability $q = 1 - p$.

Definition 7.4. A *cutset*, is a set $C_i \subset \mathcal{P}$ for which if the available participants is $\mathcal{P} \setminus C_i$, a repair is not possible. This is a *failed state*.

Let participant P_ℓ have a failed share consisting of k subshares. Let C_i be the set of blocks, other than the failed block, that contain a point i required by P_ℓ. In order for there to be a possible repair set, it cannot be the case that for any C_i relevant to the failed block, all blocks within that C_i are unavailable. This is an instance of a cutset. In other words, the probability that there exists an available repair set is $1 - Pr\{\text{at least one } C_i \text{ is unavailable}\}$.

Definition 7.5. C_i is the set of blocks containing i, excluding the block requiring a repair. C_i is said to be *failed*, denoted $\overline{C_i}$, if every member of the set is unavailable.

Definition 7.6. A repair set is *available* for a block B if no C_i, $i \in B$, fails.

We can determine $Pr\{\text{at least one } C_i \text{ is unavailable}\}$ using the *inclusion-exclusion principle*. This will be applied in Example 7.8 to show its application to $2-(v,k,1)$ designs and then applied in Example 7.9 to show its application to $3-(v,4,1)$ designs.

Definition 7.7. The *inclusion-exclusion principle for cardinalities* states that given two sets A and B,

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

The generalized form of the inclusion-exclusion principle states that

$$|A_1 \cup A_2 \cup \cdots \cup A_n| =$$

$$\sum_{1 \le i \le n} |A_i| - \sum_{1 \le i_1 < i_2 \le n} |A_{i_1} \cap A_{i_2}| + \sum_{1 \le i_1 < i_2 < i_3 \le n} |A_{i_1} \cap A_{i_2} \cap A_{i_3}| - \cdots + (-1)^{n-1} |A_1 \cap A_2 \cap \cdots \cap A_n|.$$

The *inclusion-exclusion principle for probabilities*, states that for $n = 2$,

$$Pr\{A \cup B\} = Pr\{A\} + Pr\{B\} - Pr\{A \cap B\}.$$

This variation also generalizes to n as,

$$Pr\{A_1 \cup A_2 \cup \cdots \cup A_n\} =$$

$$\sum_{1 \le i \le n} Pr\{A_i\} - \sum_{1 \le i_1 < i_2 \le n} Pr\{A_{i_1} \cap A_{i_2}\} + \cdots + (-1)^{n-1} Pr\{A_1 \cap A_2 \cap \cdots \cap A_n\}.$$

Example 7.8. Consider the $STS(7)$, as in Theorem 4.4.

Let P_ℓ be the participant who requires a repair for block 123.

1. We define the set C_i for each of the points in the block as C_1, C_2, and C_3:

   $C_1 = \{145, 176\}$
   $C_2 = \{246, 257\}$
   $C_3 = \{347, 356\}$.

2. We have $|C_i| = 2$ for all i.

3. We also have $|C_i \cup C_j| = 4$ for all i, j, $i \neq j$.

4. Finally, we have $|C_1 \cup C_2 \cup C_3| = 6$.

Let $\overline{C_i}$ indicate that every member of C_i is unavailable. Then the probability that at least one C_i is unavailable is

$$Pr\{\text{at least one } \overline{C_i}\} = Pr\{\overline{C_1} \cup \overline{C_2} \cup \overline{C_3}\}.$$

Applying the inclusion-exclusion principle results in:

$$Pr\{\overline{C_1} \cup \overline{C_2} \cup \overline{C_3}\} = q^{|C_1|} + q^{|C_2|} + q^{|C_3|} - q^{|C_1 \cup C_2|} - q^{|C_1 \cup C_3|} - q^{|C_2 \cup C_3|} + q^{|C_1 \cup C_2 \cup C_3|}.$$

Applying the computations from items 2-4 results in:

$$Pr\{\text{at least one } \overline{C_i}\} = 3q^2 - 3q^4 + q^6.$$

Therefore the probability that there exists a repairing set is

$$1 - Pr\{\text{at least one } \overline{C_i}\} = 1 - 3q^2 + 3q^4 - q^6 = (1 - q^2)^3.$$

This method has produced the same equation for the probability that a repair set exists for $STS(7)$ as was obtained in Section 4.4.

7.1.1 Existence of a Repair Set for SQS

In the case of Steiner quadruple systems, as opposed to the Steiner triple system, the sets C_i for each repair will not be disjoint, as each participant may be able to provide zero, one, or two subshares.

Example 7.9. Using the $SQS(8)$:

$$A_1 = 1234 \quad A_2 = 5678$$

$$B_1 = 1256 \quad B_7 = 3478$$
$$B_2 = 1278 \quad B_8 = 3456$$
$$B_3 = 1357 \quad B_9 = 2468$$
$$B_4 = 1368 \quad B_{10} = 2457$$
$$B_5 = 1458 \quad B_{11} = 2367$$
$$B_6 = 1467 \quad B_{12} = 2358$$

Let participant P_ℓ require a repair for block $A_1 = 1234$. Note that the block 5678 does not contain any points in common with the failed block. This leaves 12 remaining blocks that contain points relevant for the repair, namely $B_1, \ldots, B_{12}$.

1. We define the set C_i for each of the points in the block as C_1, C_2, C_3, and C_4.

   $$C_1 = \{B_1, B_2, B_3, B_4, B_5, B_6\}$$
   $$C_2 = \{B_1, B_2, B_9, B_{10}, B_{11}, B_{12}\}$$
   $$C_3 = \{B_3, B_4, B_7, B_8, B_{11}, B_{12}\}$$
   $$C_4 = \{B_5, B_6, B_7, B_8, B_9, B_{10}\}.$$

2. We can observe that $|C_i| = 6$, for all i.

2. We have $|C_i \cup C_j| = 10$, for all i, j, $i \neq j$.

3. We also have $|C_i \cup C_j \cup C_k| = 12$, for all i, j, k.

4. And finally, we have $|C_1 \cup C_2 \cup C_3 \cup C_4| = 12$, for all i, j, k.

Let $\overline{C_i}$ indicate that every member of C_i is unavailable. Combining the values from above and using the inclusion-exclusion principle as before, we get:

$$Pr\{\text{at least one } \overline{C_i}\} = 4(1-p)^6 - 6(1-p)^{10} + 4(1-p)^{12} - (1-p)^{12}.$$

Therefore, we have:

$$1 - Pr\{\text{at least one } \overline{C_i}\} = 1 - 4(1-p)^6 + 6(1-p)^{10} - 3(1-p)^{12}.$$

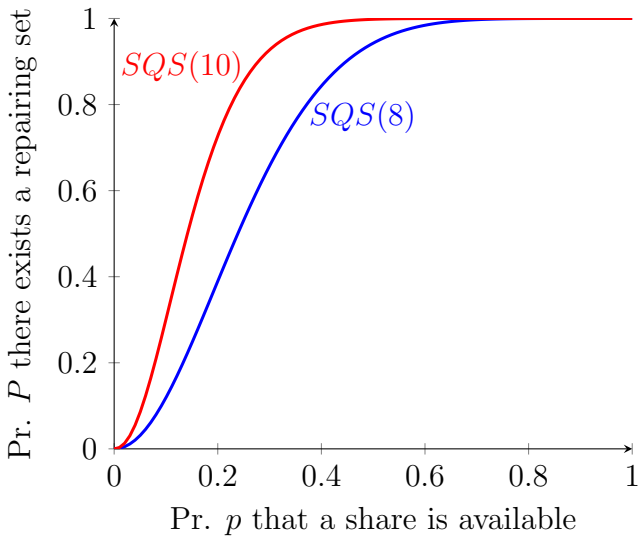


Figure 7.1: Existence of a repair set for: $SQS(8)$, $SQS(10)$

Example 7.10. Consider the $SQS(10)$ from Example 6.15

Let participant P_ℓ require a repair for block 1245.

The relevant sets for the repair are C_1, C_2, C_4, C_5. Determining the same values as for the $SQS(8)$, we have:

1. $|C_i| = 11$, for all i
2. $|C_i \cup C_j| = 19$, for all i, j, $i \neq j$
3. $|C_i \cup C_j \cup C_k| = 24$, for all i, j, k, $i \neq j \neq k$
4. $|C_1 \cup C_2 \cup C_3 \cup C_4| = 26$

$Pr\{\text{there exists a repair set}\} = 1 - 4(1-p)^{11} + 6(1-p)^{19} - 4(1-p)^{24} + (1-p)^{26}$.

We can generalize the above examples for $SQS(v)$ by applying the inclusion-exclusion principle.

Theorem 7.11. *Let $q = 1 - p$, where p is the probability that a share is available. Let $r_1 = \frac{\binom{v-1}{2}}{3}$ and $r_2 = \frac{\binom{v-2}{1}}{2}$. Then, the generalized formula for the probability of the existence of a repair set for an $SQS(v)$ is:*

$$Pr\{\textit{a repair set exists}\} = 1 - 4q^{r_1-1} + 6q^{2r_1-r_2-1} - 4q^{3r_1-3r_2} + q^{4r_1-6r_2+2}.$$

Proof. Let $\mathcal{R}(q) = Pr\{\text{a repair set exists}\}$. As per Definition 7.6, a repair set exists if no C_i fails. That is,

$$\mathcal{R}(q) = 1 - Pr\{\text{any } \overline{C_i}\}$$

For a block B in a $SQS(v)$, there are four non-disjoint sets C_i. Assume $B = abcd$.

$$Pr\{\text{any } \overline{C_i}\} = Pr\{\overline{C_a} \text{ or } \overline{C_b} \text{ or } \overline{C_c} \text{ or } \overline{C_d}\}$$

This requires the probability that one, two, three, or four different C_i fail.

1. $|C_i| = r_1 - 1$, for all $i \in B$. Therefore, $Pr\{\overline{C_i}\} = q^{r_1-1}$. There are $\binom{4}{1}$ ways to select a C_i.
2. $|C_i \cup C_j| = 2(r_1 - 1) - (r_2 - 1) = 2r_1 - r_2 - 1$, for all $i, j \in B, i \neq j$. Therefore, $Pr\{\overline{C_i} \text{ and } \overline{C_j}\} = q^{2r_1-r_2-1}$. There are $\binom{4}{2}$ ways to select C_i and C_j.
3. $|C_i \cup C_j \cup C_k| = 3(r_1 - 1) - 3(r_2 - 1) = 3r_1 - 3r_2$, for all $i, j, k \in B, i \neq j \neq k$. Therefore, $Pr\{\overline{C_i} \text{ and } \overline{C_j} \text{ and } \overline{C_k}\} = q^{3r_1-3r_2}$. There are $\binom{4}{3}$ ways to select C_i, C_j, and C_k.
4. $|C_i \cup C_j \cup C_k \cup C_\ell| = 4(r_1 - 1) - 6(r_2 - 1) = 4r_1 - 6r_2 + 2$, for all $i, j, k, \ell \in B, i \neq j \neq k \neq \ell$. Therefore, $Pr\{\overline{C_i} \text{ and } \overline{C_j} \text{ and } \overline{C_k} \text{ and } \overline{C_\ell}\} = q^{4r_1-6r_2+2}$. There are $\binom{4}{4}$ ways to select C_i, C_j, C_k, and C_ℓ.

Applying the principle of inclusion-exclusion,

$$Pr\{\overline{C_a} \text{ or } \overline{C_b} \text{ or } \overline{C_c} \text{ or } \overline{C_d}\} = \binom{4}{1}q^{r_1-1} - \binom{4}{2}q^{2r_1-r_2-1} + \binom{4}{3}q^{3r_1-3r_2} - \binom{4}{4}q^{4r_1-6r_2+2}$$

Therefore,

$$\mathcal{R}(q) = 1 - 4q^{r_1-1} + 6q^{2r_1-r_2-1} - 4q^{3r_1-3r_2} + q^{4r_1-6r_2+2}$$

□

7.1.2 Generalization for the Existence of a Repair Set

We can further generalize the formula from Theorem 7.11 to arbitrary $t-(v, k, 1)$ designs. In order to generalize, we will need to account for the fact that there are k different points in each block as opposed to stopping at four, as well as to account for the increase from triples of points occurring exactly once in the design to sets of size t occurring exactly once in the design.

Theorem 7.12. *Let $q = (1-p)$, where p is the probability that a share is available and let $r_j = \frac{\binom{v-j}{t-j}}{\binom{k-j}{t-j}}$. Then, the generalized formula for the probability of existence of a repair set for an $t-(v,k,1)$ design is:*

$$Pr\{a\ repair\ set\ exists\} = 1 - \binom{k}{1}q^{e_1} + \binom{k}{2}q^{e_2} - \binom{k}{3}q^{e_3} + \cdots + \binom{k}{k}q^{e_k}$$

where

$$e_i = \sum_{j=1}^{i}(-1)^{j+1}\binom{i}{j}(r_j - 1).$$

Proof. Let $\mathcal{R}(q) = Pr\{\text{a repair set exists}\}$. As per Definition 7.6, a repair set exists if no C_i fails. That is,

$$\mathcal{R}(q) = 1 - Pr\{\text{any } \overline{C_i}\}$$

For a block B in a $t-(v,k,1)$ design, there are k non-disjoint sets C_i required for a failed block B. Assume $B = abc\ldots k$.

$$Pr\{\text{any } \overline{C_i}\} = Pr\{\overline{C_1} \text{ or } \overline{C_2} \text{ or } \overline{C_3} \text{ or } \ldots \text{ or } \overline{C_k}\}$$

This requires the probability that one, two, three,..., or k different C_i fail.

Let e_i corresponds to $|C_{j_1} \cup C_{j_2} \cup \cdots \cup C_{j_i}|$ for $1 \leq i \leq k$. Apply the principle of inclusion-exclusion to each e_i.

$$e_1 = |C_{j_1}| = r_1 - 1$$

$$e_2 = |C_{j_1} \cup C_{j_2}| = 2(r_1 - 1) - (r_2 - 1)$$

$$e_3 = |C_{j_1} \cup C_{j_2} \cup C_{j_3}| = 3(r_1 - 1) - 3(r_2 - 1) + (r_3 - 1)$$

$$\vdots$$

$$e_k = |C_{j_1} \cup C_{j_2} \cup C_{j_3} \cup \cdots \cup C_{j_k}| = \tbinom{k}{1}(r_1 - 1) - \tbinom{k}{2}(r_2 - 1) + \tbinom{k}{3}(r_3 - 1) - \cdots + \tbinom{k}{k}(r_k - 1).$$

So, the generalized form of e_i, from the inclusion-exclusion principle, is:

$$e_i = \sum_{j=1}^{i}(-1)^{j+1}\binom{i}{j}(r_j - 1).$$

Now that we have e_i for each i, we can evaluate the probability that any number of C_i, for $1 \leq i \leq k$ fails. Recall $q = 1 - p$, where p is the probability the participant with that share is available.

1. In the case of repairing one subshare there are $\binom{k}{1}$ ways to select a C_i, maintaining a minimal repair set. For any C_i, we have $e_1 = r_1 - 1$. Therefore, $Pr\{\overline{C_i}\} = q^{e_1}$.

2. In the case of repairing two subshares there are $\binom{k}{2}$ ways to select a pair C_i and C_j, maintaining a minimal repair set. For any such pair, we have $e_2 = 2(r_1 - 1) - (r_2 - 1)$. Therefore, $Pr\{\overline{C_i} \text{ and } \overline{C_j}\} = q^{e_2}$.

3. In the case of repairing three subshares there are $\binom{k}{3}$ ways to select a triple C_i, C_j, and C_k, maintaining a minimal repair set. For any such triple, we have e_3. Therefore, $Pr\{\overline{C_i} \text{ and } \overline{C_j} \text{ and } \overline{C_k}\} = q^{e_3}$.

   $\vdots$

k. In the case of repairing k subshares there are $\binom{k}{k}$ ways to select a set of k, maintaining a minimal repair set. For any such set, we have e_k. Therefore,

$$Pr\{\overline{C_{i_1}} \text{ and } \overline{C_{i_2}} \text{ and } \ldots \overline{C_{i_k}}\} = q^{e_k}.$$

After applying the principle of inclusion-exclusion one last time we have,

$$Pr\{\overline{C_a} \text{ or } \overline{C_b} \text{ or } \ldots \text{ or } \overline{C_k}\} = \binom{k}{1} q^{e_1} - \binom{k}{2} q^{e_2} + \binom{k}{3} q^{e_3} - \cdots - \binom{k}{k} q^{e_k}$$

Therefore

$$\mathcal{R}(q) = 1 - \binom{k}{1} q^{e_1} + \binom{k}{2} q^{e_2} - \binom{k}{3} q^{e_3} + \cdots + \binom{k}{k} q^{e_k}.$$

□

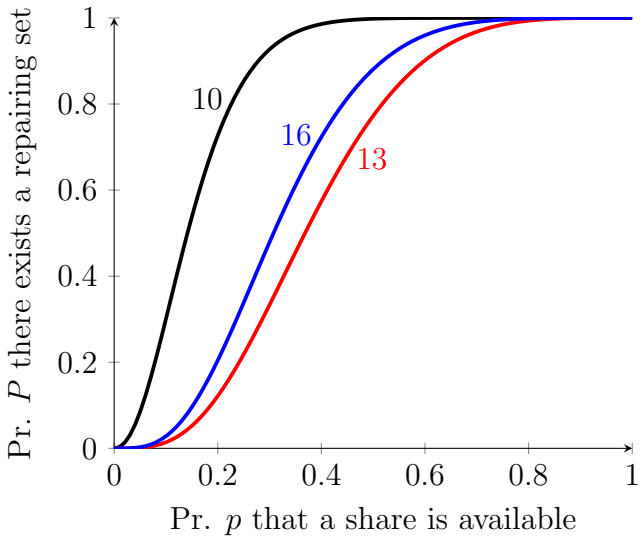


Figure 7.2: Existence of a repair set for: $2-(13,4,1)$, $2-(16,4,1)$, $SQS(10)$

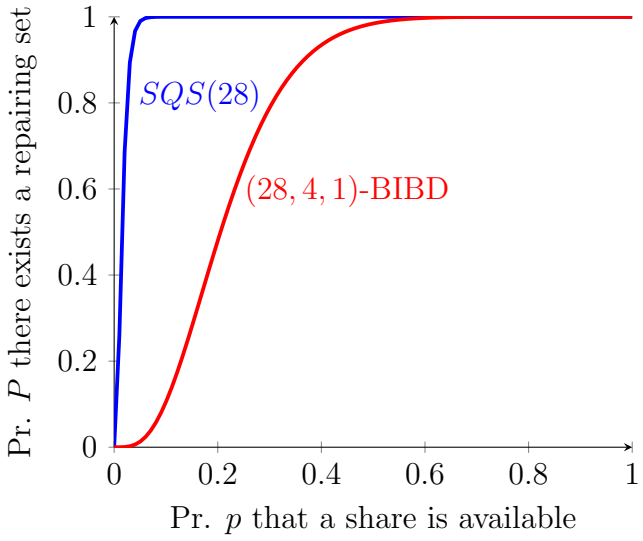


Figure 7.3: Existence of a repair set for: $SQS(28)$ and $(28,4,1)$-BIBD

Table 7.1: Repair Set Existence Comparison for $t-(v,k,1)$ Designs

t	v	k	λ	b	r_1	r_2	Value of p for $P \geq 0.99$
2	13	4	1	13	4	1	0.87
2	16	4	1	20	5	1	0.78
2	25	4	1	50	8	1	0.58
2	28	4	1	63	9	1	0.53
3	8	4	1	14	7	3	0.63
3	10	4	1	30	12	4	0.42
3	14	4	1	91	26	6	0.22
3	16	4	1	140	35	7	0.17
3	20	4	1	285	57	9	0.11
3	22	4	1	385	70	10	0.09
3	26	4	1	650	100	12	0.06
3	28	4	1	819	117	13	0.06

7.1.3 Comparing Existence for 2-designs and 3-designs

For all of the $t-(v,k,1)$ designs discussed here, the probability that a repair set exists increases with v. Larger values of v correspond to larger replication numbers, and therefore higher "slopes" in the graphs. Examples of this were shown earlier in Chapter 4 for t-designs with $t=2$. This is shown in Figure 7.1, for Steiner quadruple systems. Additionally, we can see from Figure 7.2 that the probability that there exists a repair set is increasing at a faster rate with respect to v for designs with $t=3$ than for designs with $t=2$. Within Figure 7.2, we have the probability a repair set exists for a $2-(13,4,1)$ design, for a $2-(16,4,1)$ design and finally for a $3-(10,4,1)$ design. These are labelled as 13, 16, and 10 respectively. Note that, although the $SQS(10)$ has a lower value for v, it is more likely for a repair set to exist for it than for the BIBDs when evaluated at the same value for p (the probability a share is available).

A more direct comparison of t-designs for $t=2$ and $t=3$ can be found in Figure 7.3, which shows the probability for a $2-(28,4,1)$ design and for a $3-(28,4,1)$ design. With the same value for v and for k, the t-design with $t=3$ has the probability that there exists a repair set, $P \geq 0.99$, at approximately $p=0.06$ as opposed to $p=0.53$ for $t=2$.

A summary of the parameters for all the designs discussed in this section along with a comparison of the probability p that results in $P \geq 0.99$ can be found in Table 7.1.

7.2 Expected Number of Repair Sets for SQS

Recall, from Chapter 7, that the repair set for a block belonging to a Steiner quadruple system can be of size two, three, or four. When computing the number of available minimal repair sets, we therefore must determine the number of minimal repair sets of each of the following forms:

1. pair, pair
2. pair, point, point
3. point, point, point, point

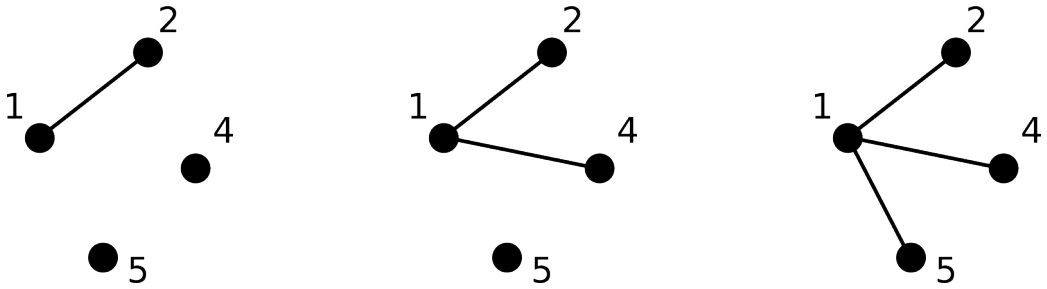
Pair, Pair

Repair sets of this form are the smallest of the three, as we only require two pairs to be provided in order to perform a repair. Each share can be considered as a set of two pairs. We know that each pair of points occurs in $r_2 - 1$ blocks in the design, other than the failed block. For a block, say 1234, there are six possible pairs, $12, 13, 14, 23, 24, 34$ which can be combined as 12 and 34, or 13 and 24, or 14 and 23. For any failed share, there exist $3(r_2 - 1)^2$ possible repairing sets of the form "pair, pair". Accounting for the availability of these repair sets, the expected number of repair sets is

$$\sum_{i=1}^{3(r_2-1)^2} p^2 = 3(r_2 - 1)^2 p^2 \tag{7.1}$$

Pair, Point, Point

Repair sets of the form "pair, point, point", are of size three and there are three different cases that may produce this form. The basic form of a "pair, point, point" repair set (shown as type (a) in Figure 7.4) has one block provide a pair while each of the two blocks contributing a single point do not have any other points in common with the failed block or the other members of the repair set.



(a) Basic (b) Mixed (c) Connected

Figure 7.4: "Pair, Point, Point" Repair Set Types

Case (a): Let $B = 1245$ be a block requiring a repair for any $SQS(v)$ and let R_1, R_2, and R_3 be the blocks forming the repair set. Assume that that $\{12\} \subset R_1$, $\{4\} \subset R_2$, and $\{5\} \subset R_3$.

Then, if the following conditions are met, we have a basic minimal repair set of the form "pair, point, point".

1. $|R_1 \cap B| = 2$
2. $|R_2 \cap B| = 1$
3. $|R_3 \cap B| = 1$

There are six possible pairs that can belong to R_1. The possible pairs to choose from are: $12, 14, 15, 24, 25, 45$. No matter which pair we choose, each of them occurs $r_2 - 1$ times in the design, other than their occurrence in the failed block requiring repair. To get an individual point, such as in condition two and in condition three, we can consider all the individual occurrences of the point while excluding all of the occurrences in a pair that occurs in the failed block as well. This leaves us with $(r_1 - 1) - 3(r_2 - 1)$ occurrences of the point. Therefore, the number of basic "pair, point, point" repair sets available can be computed as

$$\sum_{i=1}^{6(r_2-1)(r_1-3r_2+2)^2} p^3 = 6(r_2 - 1)(r_1 - 3r_2 + 2)^2 p^3. \tag{7.2a}$$

Case (b): The next form of a "pair, point, point" repair set occurs when one of the blocks contributing a single point has one other point in common with the failed block (as shown in type (b) in Figure 7.4).

Let $B = 1245$ be a block requiring a repair for any $SQS(v)$ and let R_1, R_2, and R_3 be the blocks forming the repair set. Assume that that $\{12\} \subset R_1$, $\{14\} \subset R_2$, and $\{5\} \subset R_3$.

Then if the following conditions are met, we have a basic minimal repair set of the form "pair, point, point".

1. $|R_1 \cap B| = 2$
2. $|R_2 \cap B| = 2$ and $|R_1 \cup R_2| = 3$
3. $|R_3 \cap B| = 1$

There are once again six possible pairs that can be chosen to start with in this form with each of them occurring $r_2 - 1$ times outside of their occurrence in the failed block. For the pair belonging to R_2, we could have chosen 14 or 24 to meet the conditions. Each of these pairs also occurs $r_2 - 1$ times excluding the failed block. Finally, R_3 is the same as in the basic case for this form. Therefore, the number of mixed "pair, point, point" repair sets available can be computed as

$$\sum_{i=1}^{12(r_2-1)^2(r_1-3r_2+2)} p^3 = 12(r_2-1)^2(r_1-3r_2+2)p^3. \tag{7.2b}$$

Case (c): The final form of a "pair, point, point" repair set occurs when both of the blocks contributing a single point has one other point in common with the failed block (as shown in type (c) in Figure 7.4).

Let $B = 1245$ be a block requiring a repair for an $SQS(v)$ and let R_1, R_2, and R_3 be the blocks forming the repair set. Assume that that $\{12\} \subset R_1$, $\{14\} \subset R_2$, and $\{15\} \subset R_3$.

Then if the following conditions are met, we have a basic minimal repair set of the form "pair, point, point".

1. $|R_1 \cap B| = 2$
2. $|R_2 \cap B| = 2$ and $|R_1 \cup R_2| = 3$
3. $|R_3 \cap B| = 1$ and $|R_1 \cup R_3| = 3$ and $|R_2 \cup R_3| = 3$

Assume that 1 is the connecting point for all three blocks as shown as type (c) in Figure 7.4. Each of the pairs connecting to 1 occur $r_2 - 1$ times outside of the failed block. We additionally could have chosen any of 1, 2, 4, or 5 as the connected point. Therefore, the number of connected "pair, point, point" repair sets available can be computed as:

$$\sum_{i=1}^{4(r_2-1)^3} p^3 = 4(r_2-1)^3 p^3. \tag{7.2c}$$

We can now combine the different forms of "pair, point, point" repair sets from equations (7.2a), (7.2b), and (7.2c) to produce, the expected number of repair sets of this type:

$$6(r_2-1)(r_1-3r_2+2)^2p^3 + 12(r_2-1)^2(r_1-3r_2+2)p^3 + 4(r_2-1)^3p^3$$

We can simplify this expression to the general "pair, point, point" repair set equation:

$$2(r_2-1)(3r_1^2 - 12r_1r_2 + 6r_1 + 11r_2^2 - 10r_2 + 2)p^3 \tag{7.3}$$

Point, Point, Point, Point

Finally, for repair sets of the form "point, point, point, point" we can follow similar reasoning as for the previous cases to produce:

$$\sum_{i=1}^{((r_1-1)-3(r_2-1))^4} p^4 = ((r_1-1)-3(r_2-1))^4 p^4 \tag{7.4}$$

We can now combine the expected number of repair sets for each form to produce the expected number of repair sets of any form in the following theorem.

Theorem 7.13. *The expected number of minimal repair sets of size two, three, or four for an $SQS(v)$ can be computed as:*

$$3(r_2-1)^2p^2 + 2(r_2-1)(3r_1^2 - 12r_1r_2 + 6r_1 + 11r_2^2 - 10r_2 + 2)p^3 + (r_1 - 3r_2 + 2)^4p^4$$

Proof. The expected number of repair sets of size two, three, or four can be computed as the sum of the expected number of repair sets of size two from equation 7.1, the expected number of repair sets of size three from equation 7.3, and the expected number of repair sets of size four from equation 7.4. □

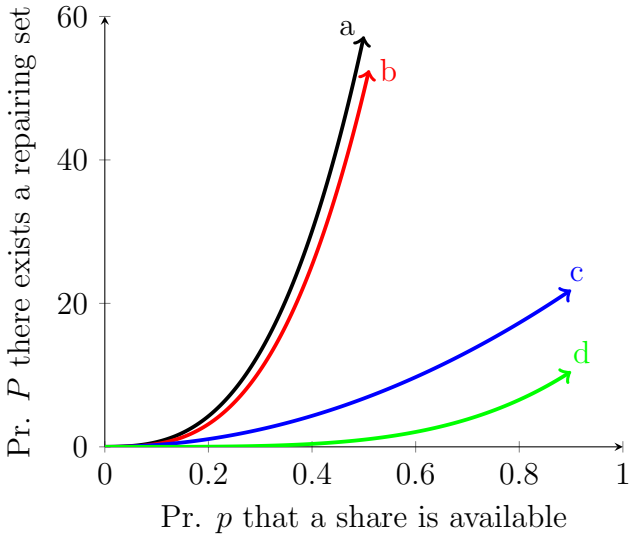


Figure 7.5: Expected number of Repair Sets By Type: $SQS(10)$

7.2.1 Comparing Expectation

When considering the expected number of available repairing sets for 2-designs and 3-designs, it is of interest to compare both the rate at which there are more repairing sets available as well as the different sized repair sets. We only have a formula for the expected number of available repair sets for Steiner quadruple systems. It seems to be infeasible to generalize to larger block size or to larger values of t (See Section 7.1.2). Therefore, for the comparisons in this section we will be evaluating $3-(v,4,1)$ designs and $2-(v,4,1)$ designs. The smallest value of v for which there exists both a $2-(v,4,1)$-design and a $3-(v,4,1)$-design is $v=28$.

Table 7.2 directly compares $2-(28,4,1)$-design and a $3-(28,4,1)$ with respect to the expected number of available repair sets of size two, three, and four for each of the two designs. The $2-(28,4,1)$-design only has repair sets of size four however, it is interesting to note the number of repair sets of size four in comparison with the $SQS(28)$. For the same value of $p=0.5$, there are 256 repair sets of size four for the $2-(28,4,1)$ design and 2560000 available repair sets of size four for the $SQS(28)$. Specifically for the $SQS(28)$, this table shows that in order to have at least one repair set of size two it is necessary to have higher values of p than for repair sets of size four. The required value of $p=0.049$ for at least one repair set of size two for the $SQS(28)$ is still lower than the required value of $p=0.130$ for at least one repair set to exist for $2-(28,4,1)$-design.

Table 7.2: Available Repair Sets of Different Sizes for $v = 28$

	Value of p for X $\geq$ 1		**X for p = 0.5**	
Repair Set Size	**2 − (28, 4, 1)**	**SQS(28)**	**2 − (28, 4, 1)**	**SQS(28)**
2	-	0.049	-	108
3	-	0.012	-	75744
4	0.130	0.013	256	2560000
any	0.130	0.010	256/4096	2635850/41566384
Ratio of expected to possible repair sets:			0.0625	0.0634

Figure 7.5 is each type for $SQS(10)$. For the same value of p, the expected number of available repairing sets increases with the value of v. This is true for both $2-(v,4,1)$ designs and $3-(v,4,1)$ designs. In the case of $3-(v,4,1)$ designs, which are Steiner quadruple systems, this increase happens at a higher rate.

Table 7.3 contains a summary of parameters for designs with $k = 4$. The table includes minimum values for p such that the expected number of available minimal repair sets is at least one. It additionally includes the expected number of available minimal repair sets when $p = 0.5$.

7.2.2 Discussion of Generalizing Expectation

In the case of existence of a repair set for $t-(v,k,1)$ designs, we were able to generalize the equations for arbitrary t, v, and k values. For expectation, it quickly gets more complicated. To see why, consider the original case of repair sets of the form "pair, point, point" for $SQS(v)$. There were three different sub-cases for this one form of a repair set. Maintaining $t = 3$, we can consider designs with other values of k. For these, there is already an increase in forms of repair sets before considering the sub-cases that may occur in each of them.

Example 7.14. Consider a $3-(v,5,1)$ design. Minimal repair sets could take the following forms:

- pair, pair, point
- pair, point, point, point

Table 7.3: Expected Number of Repair Sets Comparison for $t-(v,k,1)$ Designs

t	**v**	**k**	**λ**	**b**	$\mathbf{r_1}$	$\mathbf{r_2}$	**Value of p for X ≥ 1**	**X for p = 0.5**
2	13	4	1	13	4	1	0.340	5
2	16	4	1	20	5	1	0.250	16
2	25	4	1	50	8	1	0.150	150
2	28	4	1	63	9	1	0.130	256
3	8	4	1	14	7	3	0.230	7
3	10	4	1	30	12	4	0.12	57
3	14	4	1	91	26	6	0.049	1456
3	16	4	1	140	35	7	0.036	6247
3	20	4	1	285	57	9	0.022	75056
3	22	4	1	385	70	10	0.017	211916
3	26	4	1	650	100	12	0.012	1234590
3	28	4	1	819	117	13	0.010	2635850

- point, point, point, point, point

Additionally both the "pair, pair, point" and the "pair, point, point, point" would have sub-cases similar to that of "pair, point, point" for $SQS(v)$ which would each need to be evaluated and defined.

Example 7.15. Consider a $3-(v,6,1)$ design. Minimal repair sets could take the following forms:

- pair, pair, pair
- pair, pair, point, point
- pair, point, point, point, point
- point, point, point, point, point, point

In this case "pair, pair, point, point" and "pair, point, point, point, point" would each have sub-cases which would need to be evaluated and defined.

We see a similar expansion in complexity if we attempt to consider a generalization with respect to t. For an example of this we can consider the smallest odd and even k sizes for $t = 4$.

Example 7.16. Consider a $4-(v, 5, 1)$ design. Minimal repair sets could take the following forms:

- triple, pair
- triple, point, point
- pair, pair, point
- pair, point, point, point
- point, point, point, point, point

We see an even greater increase in potential sub-cases like that for "pair, point, point" for $SQS(v)$ with this example. All but "point, point, point, point, point" will have multiple sub-cases.

Example 7.17. Consider a $4-(v, 6, 1)$ design. Minimal repair sets could take the following forms:

- triple, triple
- triple, pair, point,
- triple point, point, point
- pair, pair, pair
- pair, pair, point, point
- pair, point, point, point, point
- point, point, point, point, point, point

Similar to the previous example we see an increase in forms which have sub-cases. In this case, all but "triple, triple" and "point, point, point, point, point, point" have sub-cases.

The number of cases to consider for the expected number of repair sets increases with both k and t. It is possible to construct equations for expectation for each $t-(v, k, 1)$ design; however, it seems that they have much variation even within the individual cases making computing each of them complex and a generalization of all of them more so.

Chapter 8

Algorithms to Find a Repair Set

In this chapter we will discuss the original algorithms from Chapter 5 with respect to t-designs. We will mainly discuss any relevant changes required to the algorithms or to the understanding of the algorithms given the t-designs. The analysis would, of course, also be impacted, but we do not include that as part of this thesis. The following discussions cover the aspects that exist in the algorithms that are understood differently in the case of t-designs for $t \geq 3$.

8.1 The Basics

For all of the algorithms, it is the case that, unlike for the 2-designs where the responding participant can only provide one subshare when participant P_ℓ requests values for any of their k subshares, for t-designs, the participant who responds is potentially able to respond with 0 to $t-1$ subshares.

In the case of Algorithm 1, the current representation of the algorithm can remain the same. As mentioned above, the analysis would change. To consider the simplest example of this, we can consider the best case run-time of the algorithm for a Steiner quadruple system. In this case the best case would be to acquire a repair set of the form "pair, pair". This would result in a run-time of $\frac{k}{2}T$, where T is the time that a participant waits for a response from the participant they contact. In comparison, the best case for a $2-(v,k,1)$ design would be kT. Note that this is the case independent of the algorithm chosen. The probability of the best case occurring is dependent on both the algorithm being used as well as whether a 2-design or other t-design is being used.

8.2 Storing Intersecting Participants

Now that we have discussed the basics, we can consider storing repair sets. For Algorithm 2 each participant stores a set $\mathcal{R}$ consisting of all the participants which can provide a share, or in the case of t-designs with $t \geq 3$, which can provide between 1 and $t-1$ shares. As before, when a participant is contacted to help perform a repair, they will then respond with the relevant subshares they possess.

8.3 Grouping Intersecting Participants

In previous chapters, storing intersecting participants consisted of storing k sets of $r_1 - 1$ participants as in Algorithm 3. It is more complicated for general $t-(t,k,1)$ designs. When we think of grouping intersecting participants for t-designs when $t \geq 3$, we need to consider how we want to address the minimal repair sets of each size. Let us first look at an example of an $SQS(v)$ and its corresponding potential repair participants.

Example 8.1. Let the underlying distribution design be an $SQS(10)$ and let participant P_ℓ require a repair for their share corresponding to block 1245.

Following the original algorithm description of how the grouped sets are stored we would have $k = 4$ sets as follows:

$$R_1 = \{1237, 1358, 1468, 1567, 1780, 1479, 1289, 1590, 1369, 1340, 1260\}$$

$$R_2 = \{1237, 2356, 2348, 2469, 2579, 2678, 1289, 2580, 2390, 1260, 2470\}$$

$$R_4 = \{2348, 2469, 3467, 2469, 4578, 4560, 1468, 1479, 4890, 1340, 2470\}$$

$$R_5 = \{1358, 2356, 3459, 3570, 4578, 4560, 5689, 1567, 2579, 2580, 1590\}$$

If we were to simply store k sets of size $r_1 - 1$ as we did previously, the algorithm could work much the same. However, it could result in unnecessary work being done if P_ℓ had already received a pair. Perhaps it makes more sense to attempt to find the corresponding pair that completes the repair at once than to continue through each of the point repair sets. We could in fact outline more than one algorithm, depending on what kind of repair sets we want to prioritize, through grouping your repair sets in different ways.

We can make different choices by prioritizing "pair, pair" repair sets. One way of doing this is to store the "pair" repair sets as follows:

$$R_{12} = \{1237, 1260, 1289\}$$

$$R_{45} = \{3459, 4578, 4560\}$$

$$R_{14} = \{1340, 1468, 1479\}$$

$$R_{15} = \{1358, 1567, 1590\}$$

$$R_{24} = \{2348, 2469, 2470\}$$

$$R_{25} = \{2356, 2579, 2580\}$$

With the "pair, pair" stored intersecting participants, after the first success the participant with the failed share would then contact participants from the corresponding "pair, pair" repair list. That is, if they received a pair from R_{12} they would contact 45. If that failed, and we were only storing "pair, pair" lists, we would then attempt to acquire the points by contacting participants from other lists. We could, however, also considering repair sets of only points which do not occur within a "pair, pair" case.

We could have participants stored grouped by which blocks contain points in isolation as follows:

$$R_1^* = \{1780, 1369\}$$

$$R_2^* = \{2678, 2390\}$$

$$R_4^* = \{3467, 4890\}$$

$$R_5^* = \{3570, 5689\}$$

If we stored both the "pair, pair" groupings as well as the points in isolation (points relevant to the failed block which do not occur in a relevant pair) we could then modify the algorithm to attempt to do any or all of minimizing the amount of participants required to repair, to only use minimal repairing sets, and to choose which type of stored participants list to use is dependent on the previous results.

8.4 Generating Grouped Intersecting Participants

Using generated intersecting participants is also possible for at least some $t - (v, k, 1)$ designs.

Theorem 8.2. *[16, Thm. 2] A cyclic $SQS(2 \cdot 5^n)$ exists for all $n \in \mathbb{N}$.*

Theorem 8.3. *[16, Thm. 3] A cyclic $SQS(v)$ exists for*

$$v = 122, 170, 194, 314, 338, 386, 458, 578.$$

For necessary and sufficient conditions for the existence of cyclic $t-(v,k,\lambda)$ designs, as well as some constructions, see Kohler [10] and Brand [3].

The $SQS(10)$ we have been working with can be generated using a cyclic automorphism. We can define the set of base blocks $\mathcal{B} = \{1245, 1237, 1358\}$. Then, all 30 blocks can be generated using the base blocks and the mapping $x \mapsto x + 1 \pmod{10}$

We can apply the same strategy as we described for Algorithm 4 earlier if we want to generate a block which has a specific point. An interesting generalization is to define a similar algorithm to find blocks with pairs.

Example 8.4. Let P_ℓ have the failed block 4890 requiring repair. It is sufficient to consider pairs from the base blocks that have a difference of 4 (mod 10) between them. Generate participants who have the pair $\ell_1\ell_2 = 48$ for participant P_ℓ as follows:

From first base block 1245:

1. Begin with the first pair $x_1x_2 = 12$ from 1245. The difference between the two points in the pair is $2 - 1 = 1 \neq 4$ and $1 - 2 = 9 \neq 4$. Therefore, this pair will not generate a block with a desired pair, and we can discard it.
2. The difference between the points for the pairs $x_1x_2 = 14$, $x_1x_2 = 25$, $x_1x_2 = 24$, and $x_1x_2 = 45$ is not equal to four and so we can discard all of these pairs.
3. Consider the final pair, $x_1x_2 = 15$ from 1245. The difference between the points in the pair is $5 - 1 = 4$, and therefore we can attempt to generate a pair block.
4. Compute $\ell_1 - x_1 = 4 - 1 \pmod{10} = 3$ and $8 - 5 \pmod{10} = 3$. We have a relevant index to get a repair block $\{1 + 3, 2 + 3, 4 + 3, 5 + 3\} = 4578$.

From second base block 1237:

1. The difference between the points for the pairs $x_1x_2 = 12$, $x_1x_2 = 13$, $x_1x_2 = 23$, and $x_1x_2 = 27$ is not equal to four and so we can discard all of these pairs.
2. Consider the next pair $x_1x_2 = 17$ from 1237. The difference between the points in the pair is $1 - 7 = 4$, and so we can generate a block.

3. Compute $\ell_1 - x_2 = 4 - 7 \pmod{10} = 7$. We have a relevant index to get a repair block $\{1+7, 2+7, 3+7, 7+7\} = 8904$. However, this is our original block for participant P_ℓ, so we ignore it.
4. Consider the final pair $x_1x_2 = 37$ from 1237. The difference is $7 - 3 = 4$, and so we can find an index to generate an appropriate block.
5. Compute $\ell_1 - x_1 = 4 - 3 \pmod{10} = 1$. We have a relevant index to get a repair block $\{1+1, 2+1, 3+1, 7+1\} = 2348$.

From third base block 1358:

1. The difference between the points for the pairs $x_1x_2 = 13$, $x_1x_2 = 15$, $x_1x_2 = 35$, $x_1x_2 = 38$, and $x_1x_2 = 18$ is not equal to four and so we can discard all of these pairs.
2. Consider the final pair $x_1x_2 = 15$ from 1358. The difference between the points in the pair is $5 - 1 = 4$ and therefore we can generate a block with a pair.
3. Compute $\ell_1 - x_1 = 4 - 1 \pmod{10} = 3$ and $8 - 5 \pmod{10} = 3$. We have a relevant index to get a repair block $\{1+3, 3+3, 5+3, 8+3\} = 4681$.

The same process can be applied to any of the other pairs from the block 4890.

Algorithm 7 RepairPairBlocksFromMultipleBaseBlocks($P_\ell, \mathcal{B}$)

```
/* Generates all relevant blocks with pairs for P_ℓ using B */
/* Assume base (v, k, 1)-BIBD */
for each base block B in B do
    for each distinct pair x_1x_2 in B do
        for each distinct pair ℓ_1ℓ_2 in P_ℓ do
            Compute d = ℓ_2 − ℓ_1
            if x_1 − x_2 ≠ d (mod v) and x_2 − x_1 ≠ d (mod v) then Break to next ℓ_1ℓ_2 pair
            if x_1 − x_2 = d then Compute e = ℓ_1 − x_2
                if The block {p_1 + e, p_2 + e, ..., p_k + e} ≠ P_ℓ then We have a repair block
            if x_2 − x_1 = d then Compute e = ℓ_1 − x_1
                if The block {p_1 + e, p_2 + e, ..., p_k + e} ≠ P_ℓ then We have a repair block
return R = {R_1, R_2, ..., R_k} for P_ℓ
```

In general, we can generate all the relevant intersecting blocks containing a pair for a participant P_ℓ using Algorithm 7.

8.4.1 Efficiency Metrics

When considering communication complexity and information rate, Theorem 3.13 still holds for designs with $t > 2$. We can compare $3-(v,k,1)$ designs to $2-(v,k,1)$ designs with respect to their information rate and their communication complexity as defined in Section 3.2. In doing so, we can see that the $2-(v,k,1)$ designs achieve the same communication complexity and information rate as the $3-(v,k,1)$ designs.

In earlier sections of this thesis, as well as in the repairable threshold scheme due to Guang et al. [8] and the repairable threshold scheme due to Stinson and Wei [19], the assumption was that a participant contributes a constant number β of subshares when participating in a repair. This assumption does not necessarily hold for $t-(v,k,1)$ designs where $t > 2$ and is dependent on the algorithm chosen. The relevant choice to determine whether β is constant is similar to choosing how to prioritize intersecting participants in the discussion on storing lists of participants for repairs. For instance, if we assume the algorithm not only prioritizes the smallest sized repair set, but instead only uses the smallest sized repair sets then β is constant. More precisely, each participant will always provide $t-1$ subshares. However, if we allow for different sized (but still minimal) repair sets it is possible for a participant to send any number of subshares from one up to and including $t-1$ subshares. Therefore, β would no longer be constant. In the case of a Steiner quadruple system β could take on the values of 2, 3, or 4.

Chapter 9

Conclusion

We have shown that combinatorial repairable threshold schemes can be analyzed using methods found in network reliability to demonstrate their robustness with respect to performing a repair. The results of our analysis can by found summarized in Table 9.2). Notably, the reliability of the scheme was improved by using t-designs where $t > 2$ in Section 7.1.3 for existence and in Section 7.2.1 for expectation. We also designed efficient algorithms (see Table 9.1) for performing a repair in a repairable threshold scheme and we developed equations for evaluating which designs are able to produce different thresholds[5].

Table 9.1: Summary of the Algorithms for $t = 2$

Notation: T is the time waiting for a response from another participant, k is the size of the blocks in the distribution design, b is the number of blocks in the design, r is the replication number, p is the probability a participant is available. Note that all algorithms require the indices corresponding to the participants share to be stored.

	Storage	**Expected Complexity**	**Fault Model**
Algorithm 1	Only share indices	$T\frac{b}{p(r-1)}\ln k$	Transient
Algorithm 2	Intersecting blocks	$\frac{k\ln k}{p}$	Transient
Algorithm 3	Grouped blocks by index	$kT\left[\frac{1-(p(r-1)+1)(1-p)^{r-1}}{p}\right]$	Permanent
Algorithm 4	Base blocks	$kT\left[\frac{1-(p(r-1)+1)(1-p)^{r-1}}{p}\right]$	Permanent

[5]See Theorem 6.16, Theorem 6.17, and Theorem 6.24.

Table 9.2: Repairable Threshold Schemes and Reliability: Summary of Results

Notation: τ is the threshold, n is the number of participants, v is the number of points in the design, k is the number of points in a block, r is the replication number, p is the probability a participant is available. The column heading "Existence" refers to the probability a repair set exists for the design. The column heading "Expectation" refers to the expected number of available repair sets for the design.

Design	$\boldsymbol{\tau}$	**n**	**Existence**	**Expectation**
STS, $\boldsymbol{\lambda}$**= 1**				
$v \equiv 1, 3 \pmod 6$, $v \geq 7$	2	$\frac{2v}{3} \leq n \leq \frac{v(v-1)}{6}$	$(1-(1-p)^{r-1})^3$	$(r-1)^3p^3$
BIBD, $\boldsymbol{\lambda}$**= 1**				
$k = 4$	2	$\frac{v}{2} \leq n \leq \frac{v(v-1)}{12}$	$(1-(1-p)^{r-1})^4$	$(r-1)^4p^4$
$k = 5$, $v \equiv 5 \pmod{20}$	2	$\frac{2v}{5} \leq n \leq \frac{v(v-1)}{20}$	$(1-(1-p)^{r-1})^5$	$(r-1)^5p^5$
$k = 8$, $v \equiv 8 \pmod{56}$	2	$\frac{v}{4} \leq n \leq \frac{v(v-1)}{56}$	$(1-(1-p)^{r-1})^8$	$(r-1)^8p^8$
$k = 5$, $v \equiv 5 \pmod{20}$	3	$\frac{2v}{5} \leq n \leq \frac{v(v-1)}{20}$	$(1-(1-p)^{r-1})^5$	$(r-1)^5p^5$
$k = 8$, $v \equiv 8 \pmod{56}$	3	$\frac{v}{4} \leq n \leq \frac{v(v-1)}{56}$	$(1-(1-p)^{r-1})^8$	$(r-1)^8p^8$
$k = 8$, $v \equiv 8 \pmod{56}$	4	$\frac{v}{4} \leq n \leq \frac{v(v-1)}{56}$	$(1-(1-p)^{r-1})^8$	$(r-1)^8p^8$
SQS, $\boldsymbol{\lambda}$**= 1**				
$v \equiv 2, 4 \pmod 6$	2	$\frac{vr_1}{k}$	See Thm. 7.11	See Thm. 7.13
t-Designs, $\mathbf{t > 2}$, $\boldsymbol{\lambda}$**= 1**				
$t = 3$, $k > 4$	3	$\frac{vr_1}{k}$	See Thm. 7.12	-

We designed efficient algorithms with trade-offs between storage and the expected communication and computation requirements for repairable threshold schemes with 2-designs as their underlying distribution design. The choice of combinatorial design was shown to have implications with respect to the algorithms used for contacting participants to perform a repair. For example, whether Algorithm 4 can be used is dependent on whether the design is able to generate all of the blocks in the design from a set of base blocks. Further implications are dependent on the value of t in the design and the choices you make for prioritizing different repair set forms (discussed in Chapter 8). To account for these implications we discussed modifications required when using $t-(v,k,1)$ designs with $t > 2$. The algorithms were analyzed under one of two probability models. The first model,

which is a transient fault, meant that if a participant is unavailable when we contact them, we could continue to contact them and they would eventually respond at some point in the future. For our pirates, this is an example where we can assume they are cursed and cannot be permanently removed by any of the many risks they face; however, they may be taking an exceptionally long voyage and be out of communication range for some time. However, if we wait long enough and they return in range, they will eventually reply and be able to provide an appropriate subshare if they have one. The other model is the case of a permanent fault. In this case, if a participant fails to reply, it is safe to assume that they have lost a battle to the sea, to insufficient supplies of citrus, or to any of the other risks they face. In this model, there is no point in attempting to contact a participant who does not reply as after a failed response: they will not respond, no matter how many attempts to contact them are made.

The equations for evaluating the repairable threshold schemes included equations for the probability a repair can occur. This resulted in a generalized formula for the probability that there exists at least one repairing set when the underlying distribution design is a $t-(v,k,1)$ design, where $t \geq 2$. We additionally determined generalized formulas for the expected number of available repair sets when the underlying distribution design is either a $2-(v,k,1)$ design or a $SQS(v)$. Finally, we included general formulas which specify the necessary restrictions on the block size k in order for a $t-(v,k,1)$ design to yield a repairable threshold scheme with threshold τ.

Given the formulas and algorithms presented in this thesis, a group of pirates (or other participants) can effectively choose appropriate distribution designs depending on the desired reliability and the preferred optimization with respect to storage or time. If the pirates have minimal storage available to them but can perform some simple computations, then they would like want to choose designs that can use Algorithm 4. If they have as much storage available to them as they need, and they cannot perform computations, they would choose to use Algorithm 3. In terms of reliability, if they want greater guarantees that a repair set exists, and they only require threshold $\tau = 2$ or $\tau = 3$, they would choose to use t-designs with $t > 2$. If they require thresholds such that $\tau > 3$, at this time they will need to choose a t-design with $t = 2$ since constructions for suitable t-designs are not currently known in the literature. With this thesis at their side, the pirates will now be able to determine which algorithms and which designs are best for ensuring the secrecy and recoverability of their treasure.

References

[1] T. Beth, D. Jungnikel, and H. Lenz. *Design Theory Volume 1 Second Ed.* Cambridge University Press, Cambridge, United Kingdom, 1999.

[2] G. R. Blakley. Safeguarding cryptographic keys. *Federal Information Processing Standard Conference Proceedings*, 48:313–317, 1979.

[3] N. Brand. Design invariants. *Geometriae Dedicata*, 21(2):169–179, 1986.

[4] C.J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, Inc., 1987.

[5] C.J. Colbourn and J. H. Dinitz. *CRC Handbook of Combinatorial Designs.* CRC press, 2010.

[6] C.J. Colbourn and A. Rosa. *Triple Systems.* Oxford University Press, Oxford, New York, 1999.

[7] S. El Rouayheb and K. Ramchandran. Fractional repetition codes for repair in distributed storage systems. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1510–1517. IEEE, 2010.

[8] X. Guang, J. Lu, and F. Fu. Repairable threshold secret sharing schemes. *arXiv preprint arXiv:1410.7190*, 2014.

[9] Captain Charles Johnson. *A General History of the Robberies and Murders of the most notorious Pyrates.* Ch. Rivington, J. Lacy, and J. Stone, Britain, 1724.

[10] E. Köhler. k-difference-cycles and the construction of cyclic t-designs. In *Geometries and Groups*, pages 195–203. Springer, 1981.

[11] T.M. Laing and D.R. Stinson. A survey and refinement of repairable threshold schemes. *Journal of Mathematical Cryptology*, 12(1):57–81, 2 2018.

[12] C.C. Lindner and C.A. Rodger. *Design Theory*. CRC Press, Boca Raton, New York, 1997.

[13] M. Nojoumian, D.R. Stinson, and M. Grainger. Unconditionally secure social secret sharing scheme. *IET Information Security*, 4(4):202–211, 2010.

[14] N.B. Shah, K. Rashmi, and P.V. Kumar. Information-theoretically secure regenerating codes for distributed storage. *Global Telecommunications Conference (GLOBECOM 2011)*, pages 1–5, 2011.

[15] A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, 1979.

[16] H. Siemon. Some remarks on the construction of cyclic steiner quadruple systems. *Archiv der Mathematik*, 49(2):166–178, 1987.

[17] D.R. Stinson. *Combinatorial Designs Constructions and Anlaysis*. Springer-Verlag New York Inc., New York, New York, 2004.

[18] D.R. Stinson and M.B.Paterson. *Cryptography Theory and Practice Fourth Edition*. Chapman & Hall CRC, Boca Raton, Florida, 2019.

[19] D.R. Stinson and R. Wei. Combinatorial repairability for threshold schemes. *Designs, Codes and Cryptography*, 86(1):195–210, 2018.