A Survey and Refinement of Repairable Threshold Schemes

Thalia M. Laing[1,2] and Douglas R. Stinson [†,3]

[1]HP Labs, Long Down Avenue, Stoke Gifford, Bristol, BS34 8QZ, UK
thalia.laing@hp.com
[2]Information Security Group, Royal Holloway University of London, Egham, Surrey, TW20 0EX, UK
[,3]David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada
dstinson@uwaterloo.ca

Abstract

We consider repairable threshold schemes (RTSs), which are threshold schemes that enable a player to securely reconstruct a lost share with help from their peers. We summarise and, where possible, refine existing RTSs and introduce a new parameter for analysis, called the repair metric. We then explore using secure regenerating codes as RTSs and find them to be immediately applicable. We compare all RTS constructions considered and conclude by presenting the best candidate solutions for when either communication complexity or information rate is prioritised.

1 Introduction

1.1 Threshold Schemes

We briefly introduce threshold schemes, then discuss the notation of share repairability in threshold schemes.

Definition 1.1. *Suppose t and n are positive integers such that $2 \leq t \leq n$. A* (t, n)-threshold scheme *is a method in which a* dealer *chooses a* secret s *and distributes a* share *to each of the n players $P_1, \ldots, P_n$, such that the following two properties are satisfied:*

- recoverability*: any subset of t players can compute the secret from the shares they collectively hold, and*

[†]Research supported by NSERC discovery grant RGPIN-03882

- secrecy*: no subset of fewer than t players can determine any information about the secret.*

A threshold scheme consists of two algorithms: a share *algorithm run by the dealer that receives as input the secret s and outputs n shares, and a* recover *algorithm, which receives as input at least t distinct, valid shares from the players and outputs the secret.*

All threshold schemes we consider here are *unconditionally secure*, meaning all security results are valid against adversaries with unlimited computational power. Such schemes were introduced independently by Blakley and Shamir in 1979 [3, 16] and have been extensively studied since [2, 9].

The following construction is due to Shamir and yields a (t, n)-threshold scheme:

Construction 1.2. *Let $\mathcal{P}$ be a set of n players, and let $p > n$ be a prime number. Let the secret space $\mathcal{S}$ be equal to the finite field of p elements, $\mathbb{Z}_p$. For a given secret $s \in \mathbb{Z}_p$, the share and recover algorithms comprising Shamir's threshold scheme are as follows.*

- *Share: Select $t-1$ values $r_1, r_2, \ldots, r_{t-1}$ uniformly at random from $\mathbb{Z}_p$, and let $f \in \mathbb{Z}_p[x]$ be the polynomial defined by*

$$f(x) = r_{t-1}x^{t-1} + r_{t-2}x^{t-2} + \cdots + r_1 x + s.$$

  We give each player P_i, for $1 \leq i \leq n$, the share $v_i = f(i)$.

- *Recover: A collection of t (or more) players perform polynomial interpolation on their shares in order to recover the polynomial f and hence determine the secret $s = f(0)$.*

Shamir's scheme is recoverable, as any set of t players can recover the secret via interpolation. Shamir's scheme also maintains secrecy, as for any set of $t-1$ or fewer players, and for any element $s' \in \mathbb{Z}_p$, there exists a polynomial of degree at most $t-1$ consistent with their shares and having constant term s'. Thus the shares of an unauthorised set of players yield no information about the true value of s.

1.2 Share repairability

Consider a scenario in which a player in a (t, n)-threshold scheme loses or corrupts their share and must repair it. In some settings, the player wishing to repair their share, called the *repairing player*, could communicate with the dealer and request, then receive, a copy of their share. However, the dealer may not always be accessible when a player needs to repair their share. Ideally, in this dealer-less setting, the repairing player could ask for help from its cohort of players to repair its share. A scheme in which this is possible is called a *repairable threshold scheme* (RTS).

Definition 1.3. *As before, let t and n be positive integers such that $2 \leq t \leq n$ and let $d \in \mathbb{N}$ be such that $t \leq d \leq n-1$. Call d the* repairing degree. *A (t, n, d)-*repairable threshold scheme*, denoted (t, n, d)-RTS, is a (t, n)-threshold scheme which, in addition to the share and recover algorithms, has a* repair *algorithm that allows a* repairing player P_r *to securely reconstruct their share with help from a set of d players, called the* helping players.

We must define what it means for the repair algorithm to be secure. Assume a setting in which all players execute the repair algorithm correctly. Inherited from the security definition of a (t, n)-threshold scheme, consider an adversary with access to a coalition of at most $t-1$ players, which may or may not include the repairing player. Each time the repair algorithm is executed, the coalition of at most $t-1$ players will pool their information; this includes the information stored prior to the algorithm being executed, as well as all messages sent and received during. In order to be secure, the accumulated information should yield no information about the secret distributed by the RTS.

We briefly note the bounds on the repairing degree d. First, consider the lower bound $t \leq d$. This is a necessary condition since, if a coalition of fewer than t players were able to construct a share for a player not in the coalition, then a coalition of $t-1$ players would be able to construct a t^{th} share and thus have enough shares to recover the secret s via the recover algorithm. This would contradict the privacy of the RTS, inherited from the threshold scheme, and would thus be insecure. The upper bound on the repairing degree is $d \leq n-1$. This is also obvious since, if one of the n players lost their share, there are at most $n-1$ players that could possibly help. We remark that it is desirable to have a small d, as this allows repairability to be more robust. For example, if $d = n-1$ and if two players are unavailable (they may be offline or corrupted), no players will be able to repair their shares. In contrast, if d is small, repairability would be possible even in a setting where several players are unavailable.

Finally, we introduce a notion, defined in [18], regarding the repairability of an RTS. As motivation for this definition, we note that in Definition 1.3 not every d-subset of the $n-1$ players is required to be able to help the repairing player P_r repair their share. Instead, it is necessary that at least one d-subset can help P_r. In order to distinguish between schemes in which all d-subsets, or some d-subsets, are able to help P_r, we introduce the following definition.

Definition 1.4. *A (t, n, d)-RTS has* universal repairability *if all d-subsets of the $n-1$ players are able to repair P_r's share. A (t, n, d)-RTS has* restricted repairability *if some, but not all, d-subsets are able to repair P_r's share.*

1.3 Our Contributions

This paper aims to survey the currently fragmented field of repairable threshold schemes. We summarise the existing work and, where possible, enhance the schemes and conduct a more thorough analysis, then explore applying the rich field of secure regenerating codes to RTSs and find them to be immediately applicable. We then conduct a comparison between all known schemes, finding

the best candidate solutions for RTSs with differing priorities. Besides unifying the current research, we highlight the following as explicit, novel contributions:

- We introduce a new efficiency metric which measures the repairability of an RTS scheme. We use this metric to analyse existing RTSs, in particular, the combinatorial schemes presented in [18].
- We enhance the enrolment scheme from [18] to achieve a smaller communication complexity whilst maintaining the information rate. We prove the minimality of the communication complexity.
- We explore using secure regenerating codes as RTSs and present a number of implications on the resulting RTSs.
- Based on our results, we propose the best candidate solutions for RTSs that prioritise either communication complexity or information rate.

1.4 Organisation

In Section 2, the relevant notions and definitions are introduced. This includes necessary definitions from combinatorial design theory and an overview of regenerating codes. In Section 3 we present a naïve construction for an RTS and introduce the metrics used to measure the efficiency of an RTS. Following this, Section 4 introduces, refines and analyses all known RTS constructions. This includes the both enrolment and combinatorial schemes presented in [18] and the scheme presented in [8]. In Section 5 we explore using regenerating codes as RTSs and find them to be immediately applicable. We then compare all the discussed RTS constructions in Section 6 and conclude in Section 7.

2 Preliminaries

In this section, we present some core ideas in combinatorial design theory from [17] and regenerating codes that we will need later.

2.1 Combinatorial design theory

Combinatorial design theory deals with arranging elements into finite sets with certain properties.

Definition 2.1. *A* d*esign is a pair* $(X, \mathcal{D})$ *such that* X *is a set of elements, called* points*, and* $\mathcal{D}$ *is a collection of non-empty subsets, called* blocks*, of* X*.*

There is no restriction on the collection $\mathcal{D}$ to have distinct blocks; this is why $\mathcal{D}$ is called a collection, rather than a set. If all the blocks are distinct, the design is called *simple*.

The *degree* of a point $x \in X$ is the number of blocks the point x occurs in. The design is called *regular* if all points have the same degree. The *rank*, k, of

a design is the largest block in the collection $\mathcal{D}$. If all blocks are the same size, the design is said to be *uniform*.

Balanced incomplete block designs are a widely studied type of design and are defined as follows:

Definition 2.2. *A* (m, k, λ)-balanced incomplete block design, (m, k, λ)*-BIBD, is a design such that*

1. $|X| = m$,
2. *each block in* $\mathcal{D}$ *contains exactly* k *points, and,*
3. *every pair of distinct points is contained in exactly* λ *blocks.*

For convenience, blocks will be written in the form abc, rather than $\{a, b, c\}$. Note that a BIBD is a regular, uniform, simple design.

Example 2.1. *The pair* $(X, \mathcal{D})$ *is a* $(9, 3, 1)$*-BIBD, where*

$$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}, \text{ and}$$
$$\mathcal{D} = \{123, 456, 789, 147, 258, 369, 159, 267, 348, 168, 249, 357\}.$$

The following theorem, presented without proof, shows the degree of every point x in a BIBD.

Theorem 2.3. *In an* (m, k, λ)*-BIBD, every point occurs in exactly*

$$\tau = \frac{\lambda(m-1)}{k-1}$$

blocks.

The value τ in Theorem 2.3 is called the *replication number* of the design. The following result, given without proof, provides more information about the structure of a BIBD and defines how many blocks a BIBD must have.

Theorem 2.4. *An* (m, k, λ)*-BIBD has exactly*

$$b = \frac{mr}{k} = \frac{m\lambda(m-1)}{k(k-1)}$$

blocks.

Continuing from Example 2.1, we compute the replication number τ and the number of blocks b for the BIBD.

Example 2.2. *Consider the* $(9, 3, 1)$*-BIBD in Example 2.1. The replication number of the design is* $\tau = 4$ *and* $\mathcal{D}$ *contains* $b = 12$ *blocks.*

In [18], the concept of a repairable distribution design and a basic repairing set are introduced, which will be used later to construct RTSs. We define the concept here and continue our example.

Definition 2.5. *A* (t, ℓ_1, ℓ_2)-distribution design *is a design that satisfies the following two properties:*

1. *the union of any* t *blocks contain at least* ℓ_2 *points, and*
2. *the union of any* $t-1$ *blocks contains at most* ℓ_1 *points,*

where $\ell_2 - \ell_1 \geq 1$. *The distribution design is* repairable *if every point in the distribution design occurs in at least two blocks.*

Example 2.3. *The* $(9, 3, 1)$*-BIBD in Example 2.1 is a* $(2, 3, 5)$*-distribution design, as the union of any two blocks contains at least five points, and an individual block contains at most three points. As every point occurs in* $\tau = 4$ *blocks, the distribution design is repairable.*

Finally, we consider the definition of a basic repairing set.

Definition 2.6. *A subset of* y *blocks contained in a* (t, ℓ_1, ℓ_2)*-distribution design is a* basic repairing set *of size* y *if every point in the design is contained in at least two blocks of the subset.*

Obviously, $y \leq b$. The following theorem lower bounds y; the proof is very similar to the proof of Theorem 2.4.

Theorem 2.7. *A basic repairing set of a* (t, ℓ_1, ℓ_2)*-distribution design, constructed from an* (m, d, λ)*-BIBD, has at least* $2m/k$ *blocks.*

Proof. Let $(X, \mathcal{D})$ be a basic repairing set of an (m, k, λ)-BIBD. Define a set

$$I = \{(y, A) : y \in X, A \in \mathcal{D}, y \in A\}.$$

We will compute $|I|$ in two different ways. First, there are m ways to choose $y \in X$. For each y, there are at least two blocks A such that $y \in A$. Hence, $|I| \geq 2m$. On the other hand, there are y ways to choose a block $A \in \mathcal{D}$. For each choice of A, there are k ways to choose $y \in A$. Hence, $|I| = yk$. Combining these two equations, we see that

$$y \geq \frac{2m}{k}, \tag{1}$$

as required. □

We illustrate the concept of basic repairing sets for our ongoing example.

Example 2.4. *For the* $(9, 3, 1)$*-BIBD in Example 2.1, we can calculate the lower bound on the basic repairing set to be* $6 \leq y$. *The set* $\{123, 456, 789, 147, 258, 369\}$ *is a basic repairing set of minimal size.*

2.2 Regenerating codes

Regenerating codes are a class of distributed storage codes introduced in 2010 by Dimakis *et al.* [5]. Regenerating codes distribute data between nodes and guarantee recoverability of the data with the cooperation of a sufficient number of nodes, and regeneration of lost or corrupted shares. Regenerating codes optimally trade the bandwidth needed for the regeneration of a failed node with the amount of data stored per node in the network.

There are no security requirements for regenerating codes. However, there exists literature exploring how to secure them. Here, we present an introduction to regenerating codes followed by a discussion on securing them.

2.2.1 Introduction to Regenerating Codes

Regenerating codes, and the notation used to describe them as in [5], is as follows.

Definition 2.8. *Let $\mathbb{F}_q$ be a finite field, and let D denote the data to be distributed, where $D \in (\mathbb{F}_q)^B$. Say B is the number of data symbols. Let $n \in \mathbb{N}$. Consider a distributed storage system consisting of n* nodes*, each with the capacity to store α symbols in $\mathbb{F}_q$. Let $t, d \in \mathbb{N}$, such that $t \leq d < n$. An* (n, t, d)-regenerating code *distributes D amongst n nodes such that each node stores a* share *of the data, where each share consists of α elements in $\mathbb{F}_q$. The distribution should be:*

- recoverable*, meaning that D can be recovered by any t of the n nodes, and*
- repairable*, meaning that any node can repair their share of the data by downloading β elements in $\mathbb{F}_q$ from each of the d repairing nodes.*

The following bound on the number of data symbols distributed by an (n, t, d)- regenerating code is:

$$B \leq \sum_{i=0}^{t-1} \min\{\alpha, (d-i)\beta\}. \tag{2}$$

Using this bound, it can be deduced that, when B, t and d are fixed, there is a trade-off between the size of the shares, α, and the bandwidth β required for repair. At one extreme of this trade-off we minimise β first and then α; this is the *minimum bandwidth regenerating (MBR) condition.* At the other extreme we minimise α and then β to get the *minimum storage regenerating (MSR) condition.*

This gives us the following parameters for the MBR condition:

$$\beta = \frac{2B}{t(2d-t+1)} \tag{3}$$

$$\alpha = \frac{2dB}{t(2d-t+1)}. \tag{4}$$

At the MSR condition, the parameters are:

$$\alpha = \frac{B}{t} \tag{5}$$

$$\beta = \frac{B}{t(d-t+1)}. \tag{6}$$

Using these extreme condition, we can define MBR and MSR codes.

Definition 2.9. *A* minimum bandwidth regenerating (MBR) code *is an* (n, t, d)*-regenerating code with parameters* (α, β, B) *satisfying the MBR conditions in* (3) *and* (4). *A* minimum storage regenerating (MSR) code *is an* (n, t, d)*-regenerating code with parameters* (α, β, B) *satisfying the MSR conditions in* (5) *and* (6).

Since MBR codes achieve the minimum possible repair bandwidth, a replacement node downloads only what it stores, so $\alpha = d\beta$. By substituting this into (2), we can see that an MBR code must satisfy

$$B = \left(td - \binom{t}{2}\right)\beta. \tag{7}$$

Similarly, MSR codes must satisfy $B = t\alpha$ and $d\beta = \alpha + (t-1)\beta$.

There exist several constructions in the literature for both MBR and MSR codes. A construction of MBR codes for all possible parameters n, t and d is given in [13]. Examples of constructions of MSR codes can be found for all parameters n, t and d in [7, 19], and for $d = 2t - 2$ in [13].

Example 2.5 shows a $(5, 2, 3)$-MBR code, constructed according to [15].

Example 2.5. *Let* $n = 5$, $t = 2$ *and* $d = 3$, *meaning that any two of the five nodes can recover the data, and any node can regenerate their share with help from three other nodes. If we let* $\beta = 1$, *then* (7) *tells us the number of message symbols that can be distributed is* $B = 5$. *Let all computations be in the field with eleven elements,* $\mathbb{Z}_{11}$, *and let*

$$u_1 = 7;\ u_2 = 3;\ u_3 = 10;\ u_4 = 6;\ u_5 = 2.$$

be the five message symbols to be distributed.

Dispersal: *Use a (public) generator matrix* Ψ, *with properties discussed in [13], and a message matrix* M *to generate the code* $C = \Psi M$ *as follows:*

$$C = \Psi M = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 1 \\ 3 & 2 & 6 \\ 4 & 2 & 1 \\ 5 & 4 & 6 \end{pmatrix} \begin{pmatrix} u_1 & u_2 & u_4 \\ u_2 & u_3 & u_5 \\ u_4 & u_5 & 0 \end{pmatrix} = \begin{pmatrix} 5 & 4 & 8 \\ 10 & 4 & 9 \\ 8 & 8 & 0 \\ 7 & 1 & 6 \\ 6 & 1 & 5 \end{pmatrix}.$$

Each node P_i *is then given row* i, *for* $1 \leq i \leq n$, *of* C. *Note that each row, and therefore each share, consists of* $\alpha = 3$ *elements in* F_{11}, *which satisfies the*

MBR conditions in (3) *and* (4).

Regeneration: *Say node* P_4 *needs to regenerate their share. This can be done with help from any* $d = 3$ *other nodes as follows. Assume nodes* P_1, P_2 *and* P_5 *are the helper nodes. Let* Ψ_i *denote row* i *of* Ψ. *Each helper node* P_i *must calculate the inner product* $(\Psi_i M)\Psi_4^T$*: note that node* P_i *knows* $(\Psi_i M)$ *as their share, and* Ψ *is a public matrix, so* Ψ_4 *is also known. Therefore, the three helper nodes* P_1, P_2 *and* P_5 *each calculate the following, respectively:*

$$(\Psi_1 M)\Psi_4^T = \begin{pmatrix} 5 & 4 & 8 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} = 3 \mod 11$$

$$(\Psi_2 M)\Psi_4^T = \begin{pmatrix} 10 & 4 & 9 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} = 2 \mod 11$$

$$(\Psi_5 M)\Psi_4^T = \begin{pmatrix} 6 & 1 & 5 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ 1 \end{pmatrix} = 9 \mod 11.$$

Each helper node then sends this value to P_4. *So* P_4 *receives the triple* $(3, 2, 9)$. *The regenerating node* P_4 *then calculates the repair matrix* Ψ_{repair}, *consisting of the rows of* Ψ *related to the helper nodes, and calculates the inverse of* Ψ_{repair}. *So, here, as* P_1, P_2 *and* P_5 *are helper nodes,* Ψ_{repair} *consists of rows* $1, 2$ *and* 5 *of* Ψ, *as follows:*

$$\Psi_{repair} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 4 & 1 \\ 5 & 4 & 6 \end{pmatrix}, \text{ and } (\Psi_{repair})^{-1} \begin{pmatrix} 3 \\ 2 \\ 9 \end{pmatrix}. \tag{8}$$

Node P_4 *then multiples* $(\Psi_{repair})^{-1}$ *with the triple* $(3, 2, 9)$ *received from the helper nodes:*

$$(\Psi_{repair})^{-1} \begin{pmatrix} 3 \\ 2 \\ 9 \end{pmatrix} = \begin{pmatrix} 9 & 9 & 8 \\ 4 & 1 & 1 \\ 10 & 1 & 2 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \\ 9 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \\ 6 \end{pmatrix}, \tag{9}$$

and thus recovers their lost share.

Recover: *Any* $t = 2$ *players can recover the data* $u_1, \ldots, u_5$. *Assume nodes* P_2 *and* P_3 *collaborate to recover the data.*

Let Ψ_{DC} *be the data collector matrix, constructed from rows corresponding to player* P_2 *and* P_3. *So:*

$$\Psi_{DC} = \begin{pmatrix} 2 & 4 & 1 \\ 3 & 2 & 6 \end{pmatrix}, \tag{10}$$

where the shares belonging to P_2 and P_3 are:

$$\Psi_{DC}M = \begin{pmatrix} 10 & 4 & 9 \\ 8 & 8 & 0 \end{pmatrix}. \tag{11}$$

We can use the properties of the message matrix M to observe that

$$\begin{pmatrix} 2 & 4 \\ 3 & 2 \end{pmatrix} \begin{pmatrix} u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} 9 \\ 0 \end{pmatrix}, \tag{12}$$

which can be solved to give $u_4 = 6$ and $u_5 = 2$. These can then be substituted into $\Psi_{DC}M$ to give:

$$\begin{aligned} \Psi_{DC}M &= \begin{pmatrix} 2 & 4 & 1 \\ 3 & 2 & 6 \end{pmatrix} \begin{pmatrix} u_1 & u_2 & 6 \\ u_2 & u_3 & 2 \\ 6 & 2 & 0 \end{pmatrix} \\ &= \begin{pmatrix} 2u_1 + 4u_2 + 6 & 2u_2 + 4u_3 + 2 & 9 \\ 3u_1 + 2u_3 + 3 & 3u_2 + 2u_3 + 1 & 0 \end{pmatrix} = \begin{pmatrix} 10 & 4 & 9 \\ 8 & 8 & 0 \end{pmatrix}, \end{aligned}$$

which gives us four equations in three variables:

$$\begin{aligned} 2u_1 + 4u_2 &= 4 \\ 2u_2 + 4u_3 &= 2 \\ 3u_1 + 2u_2 &= 5 \\ 3u_2 + 2u_3 &= 7, \end{aligned}$$

which can be solved to find $u_1 = 7, u_2 = 3$ and $u_3 = 10$. Thus, all five data symbols have been recovered by the two nodes, P_2 and P_3.

2.2.2 Securing regenerating codes

Securing regenerating codes was first explored in [12]. We introduce the necessary definitions and notation here.

Consider an adversary who has access to (only) the data stored on ℓ_1 nodes. In addition to these ℓ_1 nodes, the adversary also has access to the data stored on, and all data downloaded during the regeneration of, ℓ_2 nodes. Suppose ℓ_1 and ℓ_2 are such that $\ell_1 + \ell_2 < t$. Call such an adversary an (ℓ_1, ℓ_2)-adversary.

It is important to establish how many regenerations an adversary can witness because regenerating codes do not have any security requirements, so each regeneration may reveal information about the data stored.

In fact, each regeneration of an MBR code is secure. This is because $\alpha = d\beta$, which means a regenerating node stores all data downloaded during a regeneration. Therefore, an eavesdropper does not obtain any extra information by having access to the data downloaded during a regeneration, as well as the data stored. Hence, when considering MBR codes, we can assume an (ℓ_1, ℓ_2)-adversary is an $(\ell_1 + \ell_2, 0)$-adversary.

In contrast, MSR codes typically have insecure regenerations. This is because $\alpha < \beta d$, which means each regenerating node downloads more data during

a regeneration than it ultimately stores, and thus an adversary would learn more information witnessing a regeneration than they would if they only had access to the data stored. This extra data learnt could leak information to the adversary about the secret. So, when discussing secure MSR codes, it is important to define how many regenerations an adversary can witness.

As a side note, a node for which an adversary has access to the data on, but cannot witness regenerate, could model the adversary gaining only momentary access. In contrast, a node for which an adversary has access to both the data on and the data downloaded during a regeneration, could model an adversary with long term access to the node.

Recall the data to be distributed in a regenerating code was denoted D, such that $D \in (\mathbb{F}_q)^B$. Let $D^{(s)}$ denote the data to be securely distributed via a regenerating code, and let $D^{(s)} \in (\mathbb{F}_q)^{B^{(s)}}$. Say $B^{(s)}$ is the number of data symbols that can be (information theoretically) securely distributed. Note that $B^{(s)} \leq B$ and, in particular, $B - B^{(s)}$ is the cost of securing the data.

Definition 2.10. *A* secure (n, t, d)-regenerating code *is an (n, t, d)-regenerating code that distributes $B^{(s)}$ secure data symbols such that an (ℓ_1, ℓ_2)-adversary, for $\ell_1 + \ell_2 < t$, learns no information about the $B^{(s)}$ secure data symbols.*

We conclude this introduction to regenerating codes by stating two theorems, each providing tighter upper bounds on $B^{(s)}$.

In [12] the authors consider a regenerating code where an adversary could witness the regeneration of every node they had access to, so $\ell_1 = 0$.

Theorem 2.11. *Consider an (n, t, d)-regenerating code with parameters α and β and a $(0, \ell_2)$-adversary. The number of data symbols that can be information theoretically secured is*

$$B^{(s)} \leq \sum_{i=\ell_2}^{t-1} \min(\alpha, (d-i)\beta).$$

In [6], the authors consider the number of data symbols that can be securely distributed by an MSR code.

Theorem 2.12. *Consider an (n, t, d)-MSR code with an (ℓ_1, ℓ_2)-adversary. The number of data symbols that can be information theoretically secured is*

$$B^{(s)} \leq (t - \ell_1 - \ell_2) \left(1 - \frac{1}{d - t + 1}\right)^{\ell_2} \alpha.$$

3 A naïve solution and efficiency metrics

In this section, we present a naïve construction for an RTS that meets all the necessary requirements. We then introduce the metrics used to measure the efficiency of an RTS; most are metrics discussed in [18], but the measure of repairability is novel. These metrics will be used going forwards to analyse other RTS constructions.

3.1 Naïve solution

Consider Construction 3.1, which presents a naïve construction for a universally repairable (t, n, d)-RTS.

Construction 3.1. *Let s be the secret to be distributed. A (t, n, d)-RTS is defined by the following three algorithms, Share, Recover and Repair.*

- *Share: Distribute the secret s via a (t, n)-threshold scheme (for example, Shamir's threshold scheme from Definition 1.2), to give n shares $v_1, \ldots, v_n$. Distribute each share v_i, for $1 \leq i \leq n$ via a (d, n)-threshold scheme, resulting in the shares $v_{i,j}$, for $1 \leq i, j \leq n$. As their share, player P_i receives the $n+1$-tuple $V_i = (v_i, v_{1,i}, v_{2,i}, \ldots, v_{n,i})$.*
- *Recover: A set of players pool their shares. The elements v_i for $1 \leq i \leq n$ are input to the recover algorithm of the (t, n)-threshold scheme. If at least t players input valid shares, s will be recovered.*
- *Repair: If player P_r needs to repair their share, they request help from a set A of at least d players, who will each send P_r the element $v_{i,r}$, for i such that $P_i \in A$. Player P_r then recovers their share via the recover algorithm of the (d, n)-threshold scheme.*

Intuitively, Construction 3.1 shares a secret s via a (t, n)-threshold scheme to give player P_i their share of the secret, v_i. In order to enable repairability, each share v_i is then shared via a (d, n)-threshold scheme to ensure d players can act as helping players. The scheme is secure as any $t-1$ players are unable to learn the secret due to the security of the (t, n)-threshold scheme, and each repair is secure due to the security of each of the $(d, n-1)$-threshold schemes.

3.2 Efficiency metrics

We are interested in the efficiency of an RTS and will consider the following metrics when analysing each scheme. The first metric, information rate, is a standard definition and was presented alongside the second metric in [18], whilst the third metric, repairability, is new.

1. *Information Rate.* The first metric we consider is the information rate of the scheme, which is defined to be the ratio $\rho = \log_2 |\mathcal{V}| / \log_2 |\mathcal{S}|$, where $\mathcal{V}$ is the set of all possible shares and $\mathcal{S}$ is the set of all possible secrets. Intuitively, this measures the amount of information each player is required to store compared to the size of the secret. The information rate is such that $0 \leq \rho \leq 1$. Call an RTS with $\rho = 1$ *ideal*.
2. *Communication Complexity.* As defined in [18], the communication complexity is the sum of the sizes (*i.e.* the bit lengths) of all messages transmitted during the repair algorithm, divided by the size of the secret. The communication complexity measures the amount of bandwidth required for each execution of the repair algorithm. We denote the communication complexity by γ.

3. *Repairability.* We define the *repairability* of an RTS, denoted by κ, to be the number of d-subsets (of the $n-1$ players) that are able to help a repairing player P_r repair their share, divided by the number of possible d-subsets (of the $n-1$ players). Note that $0 \leq \kappa \leq 1$, where $\kappa = 1$ if and only if the RTS has universal repairability, as in Definition 1.4.

4. *Computational complexity.* We briefly consider the computational complexity of the share, recover and repair algorithms of the RTS.

We will see that (t, n, d)-RTSs find a compromise between these metrics and may prioritise one at the cost of the others. In particular, there appears to be an inverse relation between the information rate and communication complexity of many schemes we consider.

We now revisit Construction 3.1 and consider how efficient the scheme is using these metrics.

Example 3.1. *Consider the share, recover and repair algorithms defining a (t, n, d)-RTS as in Construction 3.1. Assuming both underlying threshold schemes are ideal, each player is required to store n shares from the threshold scheme as their RTS share. Therefore, the information rate of the RTS is $\rho = 1/n$. An execution of the repair algorithm requires each of the d players to send one of their shares from the $(d, n-1)$-threshold scheme, so $\gamma = d$. Finally, the scheme has universal repairability, since any d-subset of players can help a repairing player repair their scheme, hence $\kappa = 1$.*

We briefly note that the share algorithm of the RTS requires the dealer to run the share algorithm of the underlying (t, n)-threshold scheme once and the share algorithm of the $(d, n-1)$-threshold scheme n times. The repair algorithm requires P_r to run the recover algorithm of the underlying $(d, n-1)$-threshold scheme once, whilst the recover algorithm requires one run of the recover algorithm for the (t, n)-threshold scheme. The exact complexity depends on the complexity of the chosen underlying threshold scheme.

4 Existing solutions

In this section, we consider three (t, n, d)-RTSs presented in the literature. The first two schemes, outlined in Section 4.1.1 and 4.2, were presented in [18]. The third scheme was presented in [8] and is introduced in Section 4.3.

4.1 The enrolment RTS

We introduce the enrolment RTS, which was originally proposed in [18]. We then refine this scheme to achieve a lower communication and computational complexity. Finally, we show that the refined scheme achieves the optimal communication complexity for an ideal RTS.

4.1.1 Definition of the enrolment RTS

A (t, n, d)-RTS with $d = t$ constructed from the NSG enrolment protocol [11, 10] is presented in [18]. (In fact, a scheme equivalent to the NSG enrolment protocol was presented much earlier by Benaloh in [1].) We refer to the (t, n, d)-RTS as the *enrolment RTS*.

Assume there exists a (t, n)-threshold scheme defined over $\mathbb{F}_q$, with shares distributed amongst n players. The share and recover algorithms of the enrolment RTS are identical to the share and recover algorithms in Shamir's threshold scheme, as in [16] and defined here in Definition 1.2. The repair algorithm of the enrolment RTS is as follows.

Suppose player P_r wishes to repair their share. Without loss of generality, assume the $d = t$ helping players are players $P_1, \ldots, P_t$, with $r \geq t$. Suppose the share for P_r is $\varphi_r = f(r)$, where $f(x) \in \mathbb{F}_q[x]$ is a random polynomial of degree at most $t - 1$ whose constant term is the secret s. The share φ_r can be expressed as

$$\varphi_r = \sum_{i=1}^{t} \zeta_i \varphi_i, \tag{13}$$

where ζ_i is the public Lagrange coefficients of P_i. In order to repair P_r's share, the protocol proceeds as follows.

1. For all $1 \leq i \leq t$, player P_i computes random values $\delta_{j,i}$ for $1 \leq j \leq t$, such that
$$\zeta_i \varphi_i = \sum_{j=1}^{t} \delta_{j,i}. \tag{14}$$

2. For all $1 \leq i \leq t$, $1 \leq j \leq t$, player P_i transmits $\delta_{j,i}$ to P_j using a secure channel.

3. For all $1 \leq j \leq t$, player P_j computes
$$\sigma_j = \sum_{i=1}^{t} \delta_{j,i}. \tag{15}$$

4. For all $1 \leq j \leq t$, player P_j transmits σ_j to player P_r using a secure channel.

5. Player P_r computes their share φ_r using the formula
$$\varphi_r = \sum_{j=1}^{t} \sigma_j. \tag{16}$$

It is straightforward to verify that player P_r constructs their share correctly; that is that the value computed using (14), (15) and (16) is the same value as in (13). This is demonstrated in [18].

The enrolment protocol is proven to be secure in [18]. The proof highlights two cases: the first considers a coalition of $t-1$ players in which the players are contained in $\{P_1, \ldots, P_t\}$ and excludes P_r. The second case considers a coalition of $t-1$ players which includes P_r and $t-2$ players from $\{P_1, \ldots, P_t\}$. In either case, the coalition is unable to learn anything about a t^{th} share and thus learns no information about the secret.

In the proof, for convenience, they consider a *share-exchange matrix*, E, originally defined in [10]. We note this matrix here as it will be used to define a refined scheme in Section 4.1.3:

$$E = \begin{pmatrix} \delta_{1,1} & \delta_{2,1} & \ldots & \delta_{t,1} \\ \delta_{1,2} & \delta_{2,2} & \ldots & \delta_{t,2} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{1,t} & \delta_{2,t} & \ldots & \delta_{t,t} \end{pmatrix}. \tag{17}$$

There are a number of observations to be made about the matrix E. The $(j,i)^{\text{th}}$ entry $\delta_{j,i}$ of E is the message player P_i sends P_j. From (14), we learn that the sum of the entries in the i^{th} row of E is equal to $\zeta_i \varphi_i$. Also, from (15), the sum of the entries in the j^{th} column is equal to σ_j. Finally, from (14), (15) and (16), the sum of all entries in E is equal to φ_r. Intuitively, player P_i computes and sends all values in row i and receives all the values in column i.

4.1.2 Analysis of the enrolment RTS

We evaluate the efficiency of the enrolment RTS by considering the efficiency metrics presented in Section 3.2.

Each player is required to store only one share from Shamir's threshold scheme. As this is an ideal threshold scheme, the enrolment RTS is ideal, and so $\rho = 1$. With respect to the information rate, the enrolment RTS is optimal.

Next, we consider the communication complexity of the scheme. Messages are only exchanged in Steps 2 and 4. In Step 4, each of the t helping players are required to send one message to each of the other $t-1$ helping players, which is a total of $t(t-1)$ messages. During Step 4, each of the t helping players must send one message to the repairing player P_r, which is an additional t messages. So, in total, the repair algorithm requires $t(t-1)+t$ messages to be sent. As each message is the size of the share and therefore the size of the secret (as Shamir's scheme is ideal), the communication complexity of the scheme is $\gamma = t^2$.

Inherited from Shamir's threshold scheme, the enrolment RTS has universal repairability and thus $\kappa = 1$.

Finally, we consider the computation required for the enrolment RTS. The share and recover algorithms of the enrolment RTS are identical to the share and recover algorithms of Shamir's threshold scheme and therefore have the same complexity. In Step 1, the repair algorithm requires each of the t helping players to generate t random values and compute $t-1$ modular additions over $\mathbb{F}_q$. In Step 3, each player must again compute $t-1$ modular additions, and Step 5 requires the repairing player to compute $t-1$ modular additions. So,

in total, each helping player must compute $2(t-1)$ modular additions and the repairing player must compute $t-1$ additions. This is a total of $2t^2-t-1$ modular additions.

4.1.3 Refining the enrolment RTS: the reduced enrolment RTS

We observe that not all messages in the share-exchange matrix E are necessary to enable P_r to securely repair their share. The enrolment RTS can be refined to require fewer messages being sent, and therefore achieve a lower communication complexity, whilst maintaining the optimal information rate and the security of the enrolment RTS. In fact, we can reduce the number of messages sent so that player P_i does not send P_j a message if $j > i$. We call the resulting scheme the reduced enrolment RTS. After presenting the refined scheme, the primary task is to prove it maintains the security of the enrolment RTS. The reduced enrolment RTS is as follows.

As before, let P_r be the repairing player, and let $P_1, \ldots, P_t$ be the helping players. Let $\varphi_r = f(r)$ be the share belonging to player P_r, where $f(x) \in \mathbb{F}_q[x]$ is a random polynomial of degree at most $t-1$ whose constant term is the secret s. The share φ_r can be expressed as in (13). The reduced enrolment RTS can be executed as follows:

1. For all $1 \leq i \leq t$, player P_i computes random values $\delta_{j,i}$ for $i \leq j \leq t$, such that
$$\zeta_i \varphi_i = \sum_{j=i}^{t} \delta_{j,i}.$$
2. For all $1 \leq i \leq t$, $i < j \leq t$, player P_i transmits $\delta_{j,i}$ to P_j using a secure channel.
3. For all $1 \leq j \leq t$, player P_j computes
$$\sigma_j = \sum_{i=j}^{t} \delta_{j,i}.$$
4. For all $1 \leq j \leq t$, player P_j transmits σ_j to player P_r using a secure channel.
5. Player P_r computes their share φ_r using the formula
$$\varphi_r = \sum_{j=1}^{t} \sigma_j.$$

Verifying that player P_r computes their share correctly is similar to the verification for the enrolment RTS, as in [18].

We show the reduced enrolment RTS is secure in Theorem 4.1. The proof is similar to the proof of the security of the enrolment RTS in [18].

Theorem 4.1. *The reduced enrolment RTS is information theoretically secure against a coalition A of strictly fewer than t players.*

Proof. Assume all players act honestly during the protocol.

First, we note that computing the secret, given $t-1$ shares, is equivalent to computing any additional share. This is easy to see, because any t shares allow the secret to be computed, and any $t-1$ shares along with the secret allow any other share to be computed (this is a well-known property of Shamir's threshold scheme). As in the proof of the security of the enrolment RTS, there are two cases to consider:

Case 1: The coalition A consists of a subset of $t-1$ players in $\{P_1, \ldots, P_t\}$.

Case 2: The coalition A consists of P_r along with a subset of $t-2$ players in $\{P_1, \ldots, P_t\}$.

We consider the share-exchange matrix of the reduced enrolment RTS. The matrix here is different to the share-exchange matrix of the enrolment RTS in (17), as player P_i does not send player P_j a message if $j > i$. We can adapt E and enter a '0' into the matrix to denote no message being sent. This means values $d_{j,i} = 0$, if $j > i$. This gives us the following message-exchange matrix:

$$E' = \begin{pmatrix} \delta_{1,1} & 0 & 0 & \ldots & 0 \\ \delta_{1,2} & \delta_{2,2} & 0 & \ldots & 0 \\ & & & \ddots & \\ \ldots & \ldots & \ldots & & \ldots \\ \delta_{1,t} & \delta_{2,t} & \delta_{3,t} & \ldots & \delta_{t,t} \end{pmatrix}. \tag{18}$$

As before, the sum of the entries in the i^{th} row of E' is equal to $\zeta_i\varphi_i$, the sum of the entries of the j^{th} column is equal to σ_j, and the sum of all the entries in E' is equal to φ_r.

Consider Case 1, where A consists of a subset of $t-1$ players in $\{P_1, \ldots, P_t\}$. Assume player P_i is excluded from the coalition. Now, the coalition possess all entries in E' except for $\delta_{i,i}$. The value $\delta_{i,i}$ is completely random, and knowing this value is equivalent to knowing $\zeta_i\varphi_i$, σ_i or the secret. In fact, in order for any information to be learnt, we can use the properties of E' to deduce the following equations that would need to be solved in order to learn anything about $\delta_{i,i}$.

$$\begin{aligned} \delta_{i,i} - \zeta_i\varphi_i &= w \\ \delta_{i,i} - \sigma_i &= x \\ \zeta_i\varphi_i - \varphi_r &= y \\ \sigma_i - \varphi_r &= z, \end{aligned}$$

where

$$w = \sum_{k=1}^{i-1} \delta_{k,i} \qquad x = \sum_{k=i+1}^{t} \delta_{i,k}$$

$$y = \sum_{k=1,\, k\neq i}^{t} \zeta_k \varphi_k, \qquad z = \sum_{k=1,\, k\neq i}^{t} \sigma_k$$

are all values known to the coalition.

This leads to the following system of equations:

$$\begin{pmatrix} 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} \delta_{i,i} \\ \sigma_i \\ \zeta_i \varphi_i \\ \varphi_r \end{pmatrix} = \begin{pmatrix} w \\ x \\ y \\ z \end{pmatrix}.$$

However, the columns of the matrix on the left are linearly dependent, and thus it is possible to choose any arbitrary value for $\delta_{i,i}$, which will then determine both σ_i and $\zeta_i \varphi_i$, and then φ_r. Therefore, in Case 1, the coalition learns no information about the individual shares $\zeta_i \varphi_i$ or φ_r, and therefore learns no information about the secret being distributed.

Now, consider Case 2, where A consists of P_r and a subset of $t-2$ players in $\{P_1, \ldots, P_t\}$. Assume players P_i and P_j are omitted from the coalition, where $i < j$. In this case, the coalition knows all entries in E' except $\delta_{i,i}, \delta_{i,j}$ and $\delta_{j,j}$. Note that player P_i does not send a message to P_j; this is known by the coalition and so they know that $\delta_{j,i} = 0$. For the coalition, learning $\delta_{i,i}, \delta_{i,j}$ and $\delta_{j,j}$ is equivalent to learning $\zeta_i \varphi_i$ or $\zeta_j \varphi_j$.

As $P_r \in A$ the values σ_i, σ_j and φ_r are known. This knowledge allows the coalition to compute $\delta_{j,j}$, as

$$\sum_{k=j}^{t} \delta_{j,k} = \sigma_j$$

$$\Rightarrow \delta_{j,j} = \sigma_j - \sum_{k=j+1}^{t} \delta_{j,k}.$$

Now, the coalition are left to try to compute $\delta_{i,i}$ and $\delta_{i,j}$. Note that the sum of the these two values are known, but neither value is individually known. The following equations can be formed

$$\begin{aligned} \delta_{i,i} - \zeta_i \varphi_i &= w' \\ \delta_{i,j} - \zeta_j \varphi_j &= x' \\ \delta_{i,i} + \delta_{i,j} &= \sigma_i - y' \\ \zeta_i \varphi_i + \zeta_j \varphi_j &= \varphi_r - z, \end{aligned}$$

where

$$w' = \sum_{k=1}^{i-1} \delta_{k,i} \qquad x' = \sum_{k=1,\, k\neq j}^{j-1} \delta_{k,j}$$

$$y' = \sum_{k=i+1,\, k\neq j}^{t} \delta_{i,k}, \qquad z' = \sum_{k=i,\, k\neq i,j}^{t} \zeta_k \varphi_k$$

are all known. This leads to the following system of equations, where all values in the right-hand side vector are known:

$$\begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \delta_{i,i} \\ \delta_{i,j} \\ \zeta_i \varphi_i \\ \zeta_j \varphi_j \end{pmatrix} = \begin{pmatrix} w' \\ x' \\ \sigma_i - y' \\ \varphi_r - z' \end{pmatrix}.$$

As before, the columns in the matrix on the left are linearly dependent, and thus it is possible to choose an arbitrary value for one of $\delta_{i,i}$ or $\delta_{i,j}$, which will determine the other (as the sum of $\delta_{i,i}$ or $\delta_{i,j}$ is known), which will then, in turn, determine values for $\zeta_i \varphi_i$ and $\zeta_j \varphi_j$. Similarly, we could choose arbitrary values for $\zeta_i \varphi_i$ and $\zeta_j \varphi_j$ which would determine $\delta_{i,i}$ and $\delta_{i,j}$.

In either Case 1 or 2, the coalition A of $t-1$ players would be unable to learn any information about any additional share, and thus would learn no information about the secret. □

The reduced enrolment RTS maintains the information rate of the enrolment RTS, $\rho = 1$, yet manages to achieve a lower communication complexity, $\gamma = t(t+1)/2$. The reduced enrolment RTS is also universally repairable, so, as with the enrolment RTS, $\kappa = 1$. Finally, the reduced enrolment RTS has the same share and recover algorithms as the enrolment RTS, and thus has the same computational complexity for these algorithms. However, due to the reduced number of random messages generated by the helping players and the reduced number of messages sent, the repair algorithm is more efficient. Player P_i, for $1 \leq i \leq t$ must generate i random values, rather than t, and must compute $i-1$ modular additions, rather than $t-1$. As in the enrolment RTS, P_r must still compute $t-1$ additions. Therefore, the reduced enrolment protocol requires a total of $t(t+1)/2$ modular additions, rather than the $2t^2 - t - 1$ additions required in the enrolment RTS.

From the analysis, we can observe that the reduced enrolment RTS maintains or improves on all the efficiency metrics and is thus a more efficient RTS than the enrolment RTS presented in [18] and here in Section 4.1.1.

4.1.4 Optimal communication complexity of the reduced enrolment RTS

As we have reduced the communication complexity of the enrolment RTS, it is a natural question to ask whether it could be reduced any further. We show

here that the communication complexity for not only the enrolment RTS, but for any scheme securely computing the sum of shares, is in fact lower bounded by $\gamma = t(t+1)/2$. Thus, in this respect, the reduced enrolment RTS is optimal.

The reduced enrolment RTS is in the setting in which t players wish for an external player to (privately) compute the sum of their t shares. Any set of at most $t-1$ of the $t+1$ players (including the external player P_r) must learn no information about either one of the player's values, or the sum of the shares. In other words, the coalition must be prevented from learning all the inputs to the sum and the output.

All known private protocols, including the reduced enrolment RTS, are *oblivious protocols*. That is, the decision whether player P_i sends a message to P_j in round h is determined by i, j and h, and does not depend on the input and random coins. By assuming a private protocol, we are able to prove the lower bound on the communication complexity whilst making fewer assumptions on the protocol, such as the number of rounds required and the exact number of messages each player either sends or receives.

We prove a lower bound on the number of messages required by any oblivious protocol allowing a $t+1^{\text{th}}$ player to compute the sum of t player's values, such that the protocol is t-private, meaning any subset of at most $t-1$ players is unable to learn all inputs and the output.

Note that the proof of Theorem 4.2 proves the same result as in [4]. However, the proof in [4] considers a setting where the sum of the shares is computed by a player who also contributes an input and all players learn the output. Their proof does not immediately apply to our slightly different setting, where an external player with no input is must compute the sum and the output is known only to this external player. As well as achieving slightly different goals, the protocol presented here is executed in two rounds, rather than t rounds (as is achieved in [4]).

Theorem 4.2. *The lower bound on the number of messages required to be sent by any oblivious protocol that allows a $t+1^{th}$ player to compute the sum of t player's values, such that the protocol is t-private, is*

$$\binom{t+1}{2} = \frac{t(t+1)}{2}.$$

Proof. Consider a graph with $t+1$ vertices. Let each vertex v_i, for $1 \leq i \leq t$, correspond to a share $\zeta_i \varphi_i$, and let the $t+1^{\text{th}}$ vertex correspond to the sum of the shares. Let the (undirected) edges of the graph correspond to inputs and outputs to the vertices. That is, an edge between vertices i and j means that either player P_i sends player P_j a value, or player P_j sends P_i a value, or both players send values. We claim that knowledge of all inputs and outputs to a vertex will uniquely define the share relating to this vertex. For the $t+1^{\text{th}}$ vertex computing the sum of the inputs, this is obvious. For the other t vertices, we observe that each of these vertices must communicate their share to the $t+1^{\text{th}}$ player, and by learning all inputs and outputs to the vertex, the share must be

calculable, otherwise the $t+1^{\text{th}}$ player will not be able to use it as an input to the sum.

We will show that, in order to be secure, the graph must be complete. That is, there must be a total of $\binom{t+1}{2} = t(t+1)/2$ edges.

For the graph to be complete, there must be t edges connected to every vertex, meaning the degree of any vertex v in the graph is t. We show that, if there exists a vertex with degree less than t, a coalition of $t-1$ vertices will be able to learn more information than they should.

To show this, consider a vertex of the graph, v_x which has a degree of less than t. Specifically, let v_x have degree $t-1$. This means there exists no edge between v_x and one other node, which we denote as v_y. We do not need to specify whether either v_x or v_y are players with shares, or the player computing the sum; it does not matter. Now, consider a coalition of $t-1$ vertices A, which consists of all vertices excluding v_x and v_y. As there is no edge between v_x and v_y, all edges connected to these two vertices are also connected to vertices in A. Therefore A knows all inputs and outputs to both v_x and v_y and can thus determine the two vertices v_x and v_y (corresponding to either the shares or the sum). As A already knew the $t-1$ vertices included in the coalition, the additional information of v_x and v_y gives A knowledge of all t shares and the sum of all the shares.

Hence, if there exists a node with degree less than t, the scheme is insecure. Therefore, each node must have degree equal to at least t, meaning the minimum number of edges in the graph, and therefore the minimum number of messages sent during the protocol, is $\binom{t+1}{2} = t(t+1)/2$, as required. □

We have previously defined the reduced enrolment RTS which has a total of $\binom{t+1}{2}$ messages sent throughout. Thus, the reduced enrolment RTS is one construction that meets the lower bound for the communication complexity, and is thus optimal in this respect.

4.2 Combinatorial repairability

Also in [18], Stinson and Wei propose a way to construct (t, n, d)-RTSs based on combinatorial designs. These schemes achieve a reasonably high (but not optimal) information rate, and a low communication complexity. However, these schemes only achieve restricted repairability, as in Definition 1.4.

We first present the construction of these schemes, then provide an example. We then analyse their efficiency; in particular, we apply our new metric measuring the repairability to these schemes.

4.2.1 Definition of scheme

The share algorithm is as follows. Suppose there exists an (m, d, λ)-BIBD with b blocks, as in Definition 2.2, which is also a repairable (t, ℓ_1, ℓ_2)-distribution design, as in Definiton 2.5. Now, use an (ℓ_1, ℓ_2, m)-ramp scheme defined over $\mathbb{F}_q$ (for $q \geq m+1$) and call the shares output by the ramp scheme *sub-shares*. Let

$2m/d \leq n \leq b$, where the lower bound originates from the minimal size of the basic repairing set in (2.7) and b is the number of blocks in the BIBD. In [18], they construct a (t, n, d)-RTS with restricted repairability by allocating sub-shares from the ramp scheme to n players, as defined in the distribution design. Each block in the the design represents a player, and each point represents a sub-share. The recover algorithm requires t players to pool their shares and recover the shared secret via the recover algorithm of the (ℓ_1, ℓ_2, m)-ramp scheme. For the repair algorithm, a repairing player P_r must be sent d sub-shares from d players who each store one of P_r's sub-shares.

We will refer to schemes constructed in this manner as combinatorial RTSs. We illustrate this construction via an example.

Example 4.1. *The share algorithm of a* $(2, 12, 3)$*-RTS is as follows. Consider the* $(9, 3, 1)$*-BIBD, which is a repairable* $(2, 3, 5)$*-distribution design, used throughout the examples in Section 2.1. This design consists of* 12 *blocks; note that* $n = 12$ *and so* n *has been chosen to be maximal. Also consider a* $(3, 5, 9)$*-ramp scheme over* $\mathbb{F}_q$ *and, for convenience, label the nine shares output from the ramp scheme* $1, 2, \ldots, 9$*. Using the* $(2, 3, 5)$*-repairable distribution design and the* $(3, 5, 9)$*-ramp scheme, construct a* $(2, 12, 3)$*-RTS with restricted repairability by allocating sub-shares from the ramp scheme to the* 12 *players defined by the design, as follows:*

$$
\begin{array}{llll}
P_1 \leftarrow \{1,2,3\} & P_2 \leftarrow \{4,5,6\} & P_3 \leftarrow \{7,8,9\} & P_4 \leftarrow \{1,4,7\} \\
P_5 \leftarrow \{2,5,8\} & P_6 \leftarrow \{3,6,9\} & P_7 \leftarrow \{1,5,9\} & P_8 \leftarrow \{2,6,7\} \\
P_9 \leftarrow \{3,4,8\} & P_{10} \leftarrow \{1,6,8\} & P_{11} \leftarrow \{2,4,9\} & P_{12} \leftarrow \{3,5,7\}
\end{array}
$$

To recover the secret, any two players can pool their shares, which will consist of at least five distinct sub-shares from the $(3, 5, 9)$*-ramp scheme, and recover the distributed secret via the recover algorithm of the ramp scheme. Say player* P_5 *needs to repair their share. They can have assistance from players* P_1, P_2 *and* P_3*, who would each send the sub-shares* $2, 5$ *and* 8*, respectively.*

In their paper, Stinson and Wei propose a number of combinatorial RTSs relying on a range of designs. They give some specific parameters for the underlying designs and the resulting RTS.

Note that these RTS schemes are secure due to the underlying properties on the (ℓ_1, ℓ_2, m)-ramp scheme and the (t, ℓ_1, ℓ_2)-distribution design: a coalition of t players will learn at most ℓ_1 sub-shares output by the ramp scheme and will therefore learn no information about the secret.

4.2.2 Analysis of combinatorial RTSs

A theorem in [18] states the information rate and communication complexity of combinatorial RTSs. Assume there exists an (m, d, λ)-BIBD, which is a repairable (t, ℓ_1, ℓ_2)-distribution design with b blocks that contains a basic repairing set of size y, and suppose that $q \geq m + 1$. Let $y \leq n \leq b$. Then there exists a (t, n, d)-RTS with restricted repairability that has information

rate $\rho = (\ell_2 - \ell_1)/d$, and communication complexity $\gamma = d/(\ell_2 - \ell_1)$, where every share is in $(\mathbb{F}_q)^d$.

Before calculating the repairability of the scheme, we note one disadvantage of the combinatorial RTSs that arises from the necessary condition in the aforementioned theorem: a (t, d, n)-RTS can only be constructed if there exists an (m, d, λ)-BIBD which is also a repairable (t, ℓ_1, ℓ_2)-distribution design on m points with n blocks of size d. This is not true for all possible parameters t, d and n, and so this construction may not be able to build an RTS with the desired parameters.

So, we are left to calculate the repairability κ of the combinatorial RTSs and analyse the computational complexity.

Combinatorial RTSs only have restricted repairability, so not all d-subsets of players will have the information required to help a repairing player reconstruct their share. The probability that a randomly chosen set of d players can help a repairing player repair their share is described in Theorem 4.3.

Theorem 4.3. *A randomly chosen subset of d players in a (t, n, d)-RTS, constructed using an underlying $(m, d, 1)$-BIBD with $n = b$ players, has probability*

$$\kappa = \frac{(\tau - 1)^d}{\binom{n-1}{d}}$$

of successfully repairing the share of a player P_r, where τ is the replication number of the BIBD, as defined in Theorem 2.3.

Proof. Assume an (m, d, λ)-BIBD which is also a repairable (t, ℓ_1, ℓ_2)-distribution design on m points with n blocks of size d. Say player P_r wishes to repair their share, which consists of d sub-shares. There are a potential $n - 1$ players that could play the role of helping nodes and assist P_r in reconstructing their share. As d of these players are required to repair P_r's share, there are a total of $\binom{n-1}{d}$ d-subsets that could collaborate to help P_r. Now, we have to calculate how many of the $\binom{n-1}{d}$ d-subsets have the information required to repair P_r's share. Because of the properties of the underlying (m, d, λ)-BIBD, each sub-share occurs in exactly τ blocks, where τ is the replication number of the BIBD, calculated here as $\tau = (m - 1)/(t - 1)$. Therefore, excluding P_r's share, each sub-share is contained in $\tau - 1$ of the $n - 1$ players' shares. Now, as each pair of sub-shares occurs in one player's share, each of the d helping players must send P_r exactly one sub-share. There are $\tau - 1$ players who could contribute to recovering P_r's first sub-share, then a distinct group of $\tau - 1$ players who could contribute to recovering P_r's second sub-share, and so on, until the d^{th} sub-share. Therefore, there are $(\tau - 1)^d$ d-subsets that collectively hold the information required to help P_r recover their share. Therefore, κ is given by dividing the number of d-subsets that could successfully act as helping players, $(\tau - 1)^d$, by the total number of possible d-subsets, $\binom{n-1}{d}$, as required. □

We illustrate this proof by continuing Example 4.1.

Example 4.2. *Consider the* $(2, 12, 3)$*-RTS in Example 4.1, constructed from an underlying* $(9, 3, 1)$*-BIBD. The replication number of the BIBD is* $\tau = 4$*. Each player has* $d = 3$ *sub-shares and each sub-share is stored by* $\tau - 1 = 3$ *players, excluding the repairing player. So, there are three players that can send the repairing player the first sub-share, three other players that can send the second sub-share and three other players that can send the final sub-share. This means there are* $3^3 = 27$ *sets of three players, out of the possible* $\binom{n-1}{d} = \binom{11}{3} = 165$ 3*-subsets, that have the information required to repair* P_r*'s share. So*

$$\kappa = \frac{(r-1)^d}{\binom{n-1}{d}} = \frac{27}{165} = 0.163636.$$

So, any randomly chosen set of $d = 3$ *players will have a* 16.3636% *chance of having the information required to help the repairing player. To further illustrate this, assume* P_5 *is the repairing player. There are* 27 *sets of players that can help repair* P_5*'s share, including* $\{P_1, P_2, P_3\}, \{P_7, P_8, P_9\}$ *and* $\{P_2, P_{10}, P_{11}\}$*. There are now* $165 - 27 = 138$ *sets of three players that do not have the information required to help repair* P_5*'s share, including* $\{P_4, P_6, P_7\}$*, who do not collectively know sub-shares* 5 *and* 8*, and* $\{P_1, P_8, P_{12}\}$*, who do not know sub-share* 8*.*

We make one final comment on the reparability of the combinatorial RTSs. In [18], it is not necessary to have the number of players in the scheme n to be equal to the number of blocks b. Instead, they bound n to be $y \leq n \leq b$, where y is the size of the basic repairing set and b is the number of blocks in the design. If $n < b$, the repairability of the scheme will vary depending on the number of players, and which players, are in the RTS.

Example 4.3. *Consider a* $(2, 6, 3)$*-RTS constructed from the basic repairing set in Example 2.4. Assume* P_r *wishes to repair their share. There are a possible* $\binom{5}{3} = 10$ *subsets of the remaining five players that could act as helping players. Of these* 10 *subsets, only one set has the information required to repair* P_r*'s share. Therefore,* $\kappa = 0.1$*. So, a randomly chosen set of three players will have a* 10% *probability of being able to recover the repairing player's share.*

From now on, when computing the reparability of a combinatorial RTS, we assume the number of players is maximal, so $n = b$. In Section 6.2, Table 2 shows the reparability (as well as the information rate and communication complexity) of each of the (t, n, d)-RTSs proposed in [18].

We complete the analysis of the combinatorial RTSs by commenting on the complexity of each of the RTS algorithms. Once a (t, ℓ_1, ℓ_2)-distribution design has been chosen, the RTS share algorithm requires one execution of the share algorithm of the ramp scheme. Similarly, the RTS recover algorithm requires one execution of the recover algorithm of the ramp scheme. The RTS repair algorithm requires no computation: helping players must send sub-shares to the repairing player, but no players are required to conduct any computation. In this respect, the repair algorithm is optimal.

4.3 GLF scheme

In [15], the authors (Guang, Lu and Fu) present an information theoretically secure (t, n, d)-RTS which utilises MBR codes and linearised polynomials. Intuitively, they distribute a secret via a (t, t)-threshold scheme using a random, linearised polynomial with the constant term equal to the secret, then the shares are treated as the message symbols in a (t, n, d)-MBR code. Their scheme works for all parameters n, t and d.

The GLF construction achieves an information rate of $\rho = 1/d\beta$ and a communication complexity of $\gamma = d\beta$, where β is as defined by the MBR code. The GLF RTS is universally repairable and so $\kappa = 1$.

The share algorithm of the GLF RTS requires the generation of a linearised polynomial and the evaluation of t points on this polynomial; this is followed by the computation of the necessary MBR code, where the shares are treated as message symbols. The GLF recover algorithm requires recovery of the message symbols via the MBR code, then recovering the linearised polynomial from the message symbols. The repair algorithm is identical to the regeneration of a node in the underlying MBR code.

We will see how, when considering secure regeneration codes in Section 5, MBR codes can be used to achieve schemes with a better information rate and communication complexity than is achieved by the GLF RTS.

5 Solutions using regenerating codes

Secure regenerating codes, defined in Definition 2.10, can be directly used to construct (t, n, d)-RTSs. However, the work discussing secure regenerating codes has not been presented in the framework of threshold schemes and RTSs. In this section, we briefly explore the application of secure regenerating codes to RTSs and discuss relevant parameters. Following this, we present several constructions for secure regenerating codes.

5.1 Applying regenerating codes to RTSs

Now, we translate the language used in regenerating codes into that used by repairable threshold schemes. Each node is equivalent to a player, and the data stored by the node is the player's share. A regenerating node is equivalent to a repairing player. The strongest adversary considered in an RTS is equivalent to an (ℓ_1, ℓ_2)-adversary against a secure regenerating code, where $\ell_1 = 0$ and $\ell_2 = t - 1$. So, if we wish to build a (t, n, d)-RTS, we can trivially use an information theoretically secure (n, t, d)-regenerating code.

In general, the information rate of a (t, n, d)-RTS based on a regenerating code is

$$\rho = \frac{B^{(s)}}{\alpha},$$

and the communication complexity is

$$\gamma = \frac{d\beta}{B^{(s)}}.$$

As all regenerating codes have universal repairability, all (t, n, d)-RTSs based on MBR codes have $\kappa = 1$.

Note that, because $\alpha = d\beta$ for all MBR codes, the communication complexity and information rate of these schemes are reciprocals, as was noted in [18] when discussing the combinatorial RTSs.

Prior to considering constructions of secure regenerating codes, we consider implications of the bounds given in Theorems 2.11 and 2.12 when considered in the setting of RTSs.

5.1.1 Using MBR codes as RTSs

The first corollary considers the information rate of an RTS based on an MBR code.

Corollary 5.1. *A (t, n, d)-RTS based on a secure (n, t, d)-MBR code with a $(0, t-1)$-adversary cannot be ideal.*

Proof. The information rate of the RTS is calculated as $\rho = B^{(s)}/\alpha$. The scheme is ideal if $\rho = 1$, which is true if and only if $B^{(s)} = \alpha$. Now, substitute the MBR condition from (3) and (4) into the bound given in Theorem 2.11. This gives the maximum number of messages symbols that can be securely distributed by an MBR code (as in [15]) to be

$$B^{(s)} = \left(td - \binom{t}{2}\right)\beta - \left(\ell_2 d - \binom{\ell_2}{2}\right)\beta. \tag{19}$$

By substituting in $\ell_2 = t - 1$, we learn that $B^{(s)} = \beta(d - t + 1)$. So, $B^{(s)} = \alpha$ if and only if $\beta(d-t+1) = \alpha$. But in all MBR codes, $\alpha = d\beta$, and so $d-t+1 = d$. This is only true if $t = 1$. However, in the definition of threshold schemes, given in Definition 1.1, t is defined to be such that $t \geq 2$ as, if $t = 1$, any individual player could recover the secret. Therefore, a (t, n, d)-RTS constructed from a secure (n, t, d)-MBR code cannot be ideal. □

5.1.2 Using MSR codes as RTSs

The following two corollaries consider the limitations of RTSs based on MSR codes.

Corollary 5.2. *A (t, n, d)-RTS based on an (n, t, d)-MSR code cannot securely distribute any messages if $d = t$.*

Proof. Consider the bound given in Theorem 2.12. By setting $d = t$, we can see immediately that $B^{(s)} \leq 0$. □

Corollary 5.3. *A (t, n, d)-RTS based on an (n, t, d)-MSR code with a $(0, t-1)$-adversary cannot be ideal. In fact, if $\ell_1 + \ell_2 = t - 1$, the RTS can only be ideal against a $(t-1, 0)$-adversary.*

Proof. Assume we have an optimal (in terms of $B^{(s)}$), secure (n, t, d)-MSR code. The information rate of the (t, n, d)-RTS based on this MSR code is

$$\begin{aligned}\rho &= \frac{B^{(s)}}{\alpha} \\ &= (t - \ell_1 - \ell_2)\left(1 - \frac{1}{d - t + 1}\right)^{\ell_2}. \end{aligned} \tag{20}$$

As we have assumed $\ell_1 + \ell_2 = t - 1$, we can substitute this into (20), so

$$\rho = \left(1 - \frac{1}{d - t + 1}\right)^{\ell_2}. \tag{21}$$

For the (t, n, d)-RTS to be ideal, ρ must equal 1. This is true if and only if $\ell_2 = 0$, as required. □

In fact, (21) illustrates how the information rate of a (t, n, d)-RTS based on an MSR code with an (ℓ_1, ℓ_2)- adversary, such that $\ell_1 + \ell_2 = t - 1$, decreases as ℓ_2 increases. This is because, as previously explained, the repair algorithm for MSR codes leaks information as $\alpha < \beta d$. Because of this, MSR codes may not be the best way to construct RTSs as they achieve a small information rate when considering a maximal adversary with $\ell_1 = 0$ and $\ell_2 = t - 1$.

Now, we consider secure regenerating codes as RTSs and analyse results.

5.2 Constructions of secure regenerating codes for RTSs

We consider three constructions for secure regenerating codes. The first of the three, [15], is a secure MBR code and can be constructed for all parameters t, n, d such that $t \leq d < n$. The next two constructions we consider, from [14] and [15], are secure MSR constructions and are for specified values of d.

5.2.1 SRK's secure MBR construction [15]

In [15], the authors (Shah, Rashmi and Kumar) present an information theoretically secure (n, t, d)-MBR code based on a matrix product construction for MBR codes, presented in [13] and used in Example 2.5. Without loss of generality, they construct codes for the case where $\beta = 1$, and codes for any higher value of β can be obtained by a simple concatenation of the code with $\beta = 1$ (this technique is called *striping* and is detailed in [13]). We call this construction the SRK-MBR construction.

The SRK-MBR construction achieves a secure code by replacing a carefully chosen set of $B - B^{(s)}$ message symbols with symbols which are chosen uniformly and independently at random from $\mathbb{F}_q$. If these random values are treated as

message symbols, the secure regenerating code is identical to the standard regenerating code, and so distribution, recovery and repair are as in the regeneration code defined in [13]. They prove the secure construction is information theoretically secure. The SRK-MBR scheme achieves the maximum bound for $B^{(s)}$, as in Theorem 2.11, with $\beta = 1$.

Consider this secure (n, t, d)-regenerating code as a (t, n, d)-RTS, and consider a $(0, t-1)$-adversary. As the repair algorithm of an MBR code is inherently secure, a $(0, t-1)$-adversary is equivalent to a $(t-1, 0)$-adversary. We can substitute $\ell_2 = 0$ and $\beta = 1$ into (19) to calculate

$$\begin{aligned} B^{(s)} &= \left(td - \binom{t}{2}\right) - \left((t-1)d - \binom{t-1}{2}\right) \\ &= d - t + 1. \end{aligned}$$

Now, as each player stores α elements in $\mathbb{F}_q$, where $\alpha = d\beta$ is as in (4), and as $\beta = 1$, the information rate and communication complexity metrics can be computed as follows:

$$\begin{aligned} \rho &= \frac{d-t+1}{d}, \\ \gamma &= \frac{d}{d-t+1}. \end{aligned}$$

As regenerating codes have universal repairability, $\kappa = 1$ for all secure regenerating codes treated as an RTS.

We present a brief example of this secure construction, which is a continuation from Example 2.5.

Example 5.1. *Consider the code in Example 2.5. The number of symbols that can be securely distributed via this* $(5, 2, 3)$*-MBR code is* $B^{(s)} = 2$*. Replace* u_1, u_2 *and* u_4 *with random elements in the field, and let* u_3 *and* u_4 *be the secure message symbols to be distributed. This is then a secure* $(5, 2, 3)$*-RTS against a* $(1, 0)$*-adversary and has information rate* $\rho = 2/3$*, communication complexity* $\gamma = 3/2$ *and repairability* $\kappa = 1$*.*

Finally, we briefly comment on the complexity of the scheme. All three algorithms, share, repair and recover, require all players to compute linear computations.

5.2.2 Rawat's MSR construction [14]

In [14], Rawat proposes an information theoretically secure (n, t, d)-MSR code for $d = n - 1$. The secure code is based on a construction for general MSR codes proposed in [19] which is valid for all parameters n, t and d. Rawat's MSR construction can be treated as a $(t, n, n-1)$-RTS with a $(0, t-1)$-adversary.

Rawat's construction is optimal with respect to $B^{(s)}$, as given in Theorem 2.12. Thus, by substituting in $d = n - 1$, the information rate of the

scheme can be calculated as

$$\rho = \frac{B^{(s)}}{\alpha} = \left(1 - \frac{1}{n-t}\right)^{t-1}.$$

To calculate the communication complexity, it is useful to calculate $d\beta$ using the value for β and $B = \alpha t$, given in the MSR trade-off condition. This gives that

$$d\beta = \frac{dB}{t(d-t+1)} = \frac{\alpha t(n-1)}{t(n-t)} = \frac{\alpha(n-1)}{(n-t)},$$

where α is defined to be $\alpha = (n-t)^{n-1}$ in the scheme. Then,

$$d\beta = \frac{(n-t)^{n-1}(n-1)}{n-t} = (n-t)^{n-2}(n-1).$$

Then the communication complexity of the scheme is

$$\gamma = \frac{d\beta}{B^{(s)}} = \frac{(n-t)^{n-2}(n-1)}{\left(1 - \frac{1}{n-t}\right)^{t-1}(n-t)^{n-1}} = \frac{(n-1)}{\left(1 - \frac{1}{n-t}\right)^{t-1}(n-t)}.$$

As before, $\kappa = 1$. Also, as with other schemes based on regenerating codes, all three algorithms, share, repair and recover, require computations of linear combinations.

5.2.3 SRK's secure MSR construction [15]

In [15], the authors present a secure (t, n, d)-RTS based on MSR codes, for $d = 2t - 2$. We call this the SRK-MSR construction. SRK-MSR is similar to SRK-MBR and is also based on the constructions given in [13], which are suitable for when $d = 2t-2$ (they say the schemes can be extended so $d > 2k-2$ via shortening), and thus the SRK-MBR is also for parameters $d = 2t - 2$. As in SRK-MBR, SRK-MSR consists of replacing a carefully selected subset of the message symbols with random values.

The SRK-MSR scheme is able to distribute $B^{(s)} = (t - \ell_1 - \ell_2)(\alpha - \ell_2\beta)$ messages securely. The authors claim their scheme is optimal when $\ell_2 = 0$, but say it is unknown whether it is optimal when $\ell_2 > 0$. We answer here that the MSR scheme is optimal if $\ell_2 = 0$ or 1, but is not optimal for $\ell_2 > 1$.

Corollary 5.4. *The (t, n, d)-RTS constructed from the SRK-MSR construction presented in [15] is optimal with respect to the number of messages that can be securely distributed, $B^{(s)}$, if $\ell_2 \leq 1$. If $\ell_2 > 1$, the construction is not optimal with respect to $B^{(s)}$.*

Proof. Denote the number of message symbols the SRK-MSR construction in [15] can securely distribute as

$$B^{(s)}_{SRK} = (t - \ell_1 - \ell_2)(\alpha - \ell_2\beta).$$

We can substitute in the values for $\beta = B/t(d-t+1)$ and $B = \alpha t$, given at the MSR trade-off condition in (5) and (6), to get

$$\begin{aligned} B^{(s)}_{SRK} &= (t - \ell_1 - \ell_2)\left(\alpha - \frac{\ell_2 \alpha t}{t(d-t+1)}\right) \\ &= (t - \ell_1 - \ell_2)\alpha\left(1 - \frac{\ell_2}{(d-t+1)}\right). \end{aligned}$$

Then we can compare this to the bound $B^{(s)}$ for all MSR codes given in Theorem 2.12. If we divide both values by $\alpha(t - \ell_1 - \ell_2)$, we can see that

$$\begin{aligned} \frac{B^{(s)}_{SRK}}{\alpha(t-\ell_1-\ell_2)} &= 1 - \frac{\ell_2}{d-t+1} \\ &\leq \left(1 - \frac{1}{d-t+1}\right)^{\ell_2} = \frac{B^{(s)}}{\alpha(t-\ell_1-\ell_2)}, \end{aligned}$$

with an equality when ℓ_2 equals either zero or one, but strictly greater than when $\ell_2 > 1$, as required. □

Consider the SRK-MSR construction as a $(t, n, 2t-2)$-RTS with a $(0, t-1)$-adversary. We can substitute $\beta = 1$ and $d = 2t - 2$ into the MSR condition to give that $B = \alpha(\alpha + 1)$ and $\alpha = t - 1$, so $d = 2\alpha$. Now, we can calculate, $B^{(s)} = \alpha - (t-1) = (t-1) - (t-1) = 0$. Therefore the SRK-MSR construction cannot be used to securely distribute any symbols when a $(0, t-1)$-adversary is considered.

6 Comparison of techniques

In this section, we compare the RTS constructions introduced throughout. We begin by comparing MBR and MSR based schemes. Then, we compare schemes that prioritise communication complexity above information rate, followed by schemes prioritising information rate.

6.1 Comparing MBR and MSR codes

Here, we highlight some similarities and differences between (t, n, d)-RTSs that are constructions of secure MBR and MSR codes.

Firstly, all (t, n, d)-RTSs based on either MBR or MSR codes have universal repairability, so $\kappa = 1$, always.

However, there are a number of major differences. MBR codes prioritise bandwidth and thus the (t, n, d)-RTSs based on them achieve a lower communication complexity than schemes based on MSR codes. In contrast, MSR codes prioritise storage and thus the (t, n, d)-RTSs based on MSR codes generally achieve higher information rates. However, secure MSR codes cannot be ideal RTSs unless $\ell_2 = 0$, meaning the adversary witnesses no regenerations, which may be unrealistic in the RTS setting.

Importantly, the repair algorithm for RTSs based on MBR codes is secure: the adversary is able to witness any number of regenerations and no information will be learnt. However, crucially, the repair algorithm for RTSs based on MSR codes is insecure: with each distinct regeneration, the adversary learns more information. Thus, the number of regenerations the adversary can witness affects the number of message symbols that can be securely distributed; the more regenerations witnessed, the fewer message symbols can be secured. In settings where $\ell_2 = t - 1$, which is what is considered here for an RTS, MSR codes may not be very useful.

Finally, secure MBR based (t, n, d)-RTSs exist for all valid parameters n, t and d. In the current literature, there does not appear to be any secure construction based on MSR codes for all valid parameters, and a secure construction for $d = t$ is impossible.

Thus, between MBR based and MSR based RTSs, MBR based schemes appear to be the most applicable to RTSs, mainly because of the secure repair algorithm. In particular, the secure MBR construction presented in [15] appears to be the most applicable construction from the field of regenerating codes. This is because it achieves the upper bound for the value of $B^{(s)}$ and is therefore optimal and the repair algorithm is secure, meaning the adversary can witness multiple repairs.

Table 1 compares the information rate and communication complexity for all universally repairable schemes considered for an example set of parameters. Due to the restrictions of the repairing degree d for the secure MSR constructions, the table considers the metrics for a $(4, 6, 6)$-RTS using the SRK-MBR [15], Rawat [14], SRK-MSR [15] and GLF [8] constructions. The metrics for a $(4, 7, 4)$-RTS from the enrolment and reduced enrolment RTSs are also included. It is possible to see from Table 1 that, out of all RTSs based on regenerating codes, the SRK-MBR construction achieves the best information rate and the best communication complexity for the given parameters.

(t, n, d)-RTS	Construction	ρ	γ	κ
$(4, 7, 6)-$RTS	SRK-MBR [15]	1/2	2	1
	Rawat [14]	8/27	27/4	1
	SRK-MSR [15]	*Not possible*		
	GLF [8]	1/6	6	1
$(4, 7, 4)-$RTS	Enrolment [18]	1	16	1
	Reduced Enrolment	1	10	1

Table 1: Comparing metrics for universally repairable RTS constructions.

	Combinatorial Schemes [18]					MBR Schemes [15]		
(t,n,d)-RTS	$(m,d,1)$-BIBDs	n	ρ	γ	κ	ρ	γ	κ
$(2,n,3)$	$(9,3,1)$	$6 \leq n \leq 12$	2/3	3/2	0.1636	2/3	3/2	1
$(2,n,3)$	$(15,3,1)$	$10 \leq n \leq 35$	2/3	3/2	0.0361	2/3	3/2	1
$(2,n,3)$	$(21,3,1)$	$14 \leq n \leq 70$	2/3	3/2	0.0139	2/3	3/2	1
$(2,n,4)$	$(16,4,1)$	$8 \leq n \leq 20$	3/4	4/3	0.0060	3/4	4/3	1
$(2,n,4)$	$(28,4,1)$	$14 \leq n \leq 63$	3/4	4/3	0.0073	3/4	4/3	1
$(2,n,4)$	$(40,4,1)$	$20 \leq n \leq 130$	3/4	4/3	0.0019	3/4	4/3	1
$(2,n,5)$	$(25,5,1)$	$10 \leq n \leq 30$	4/5	5/4	0.0263	4/5	5/4	1
$(3,n,5)$	$(25,5,1)$	$10 \leq n \leq 30$	2/5	5/2	0.0263	3/5	5/3	1
$(2,n,5)$	$(65,5,1)$	$26 \leq n \leq 208$	4/5	5/4	0.0003	4/5	5/4	1
$(3,n,5)$	$(65,5,1)$	$26 \leq n \leq 208$	2/5	5/2	0.0003	3/5	5/3	1
$(2,n,8)$	$(64,8,1)$	$16 \leq n \leq 72$	7/8	8/7	0.0016	7/8	8/7	1
$(3,n,8)$	$(64,8,1)$	$16 \leq n \leq 72$	5/8	8/5	0.0016	3/4	4/3	1
$(4,n,8)$	$(64,8,1)$	$16 \leq n \leq 72$	1/4	4	0.0016	5/8	8/5	1
$(2,n,4)$	$(13,4,1)$	$9 \leq n \leq 13$	3/4	4/3	0.0136	3/4	4/3	1
$(3,n,4)$	$(13,4,1)$	$9 \leq n \leq 13$	1/2	2	0.0136	1/2	2	1
$(2,n,5)$	$(21,5,1)$	$12 \leq n \leq 21$	4/5	5/4	0.0660	4/5	5/4	1
$(3,n,5)$	$(21,5,1)$	$12 \leq n \leq 21$	3/5	5/3	0.0660	3/5	5/3	1
$(4,n,5)$	$(21,5,1)$	$12 \leq n \leq 21$	1/5	5	0.0660	2/5	5/2	1
$(2,n,6)$	$(31,6,1)$	$15 \leq n \leq 31$	5/6	6/5	0.0263	5/6	6/5	1
$(3,n,6)$	$(31,6,1)$	$15 \leq n \leq 31$	2/3	3/2	0.0263	2/3	3/2	1
$(4,n,6)$	$(31,6,1)$	$15 \leq n \leq 31$	1/3	3	0.0263	1/2	2	1

Table 2: Comparison of $(t,n,d)-$RTSs based on $(m,d,1)-$BIBDs, as in [18], and secure MBR codes, as in [15].

6.2 Comparison of techniques prioritising communication complexity

Both secure MBR schemes and combinatorial RTSs prioritise communication complexity over information rate. Here, we compare the SRK-MBR construction with the combinatorial RTSs given in [18].

Table 2 shows how the combinatorial RTSs in [18] compare to those based on secure MBR codes in [15] for certain parameters. The comparison shows the information rate ρ, communication complexity γ and repairability κ (assuming n is maximal for the combinatorial schemes), with highlighted rows showing the RTSs with different ρ and γ. The chosen parameters relate to proposed combinatorial RTSs in [18]; note that MBR schemes exist for all valid parameters of n, t and d, but the combinatorial RTSs rely on the existence of an underlying design with relevant parameters.

From Table 2, we can see the schemes achieve equal information rate and communication complexities in most cases. In some cases, which have been

highlighted, the SRK-MBR scheme achieves a better information rate and a better communication complexity than the combinatorial RTSs. In no case do the combinatorial RTSs achieve better results than the SRK-MBR construction. In fact, when the concept of restricted repairability was introduced in [18], it was suggested that compromising repairability may enable more efficient schemes. However, this appears to not be the case, at least so far.

As well as achieving similar or better values for ρ and γ in all defined parameters for t and d in [18], the SRK-MBR construction has two advantages over the combinatorial RTSs:

1. The MBR schemes in [15] achieve universal, rather than restricted, repairability.
2. The combinatorial RTSs in [18] depend on the existence of certain combinatorial constructions. This is in contrast to MBR schemes, which can be constructed for all valid parameters n, t and d.

However, one advantage of the combinatorial RTSs in [18] is the computational complexity of repairing a share. In schemes based on either MBR or MSR codes, every repair requires helping nodes and the repairing node to execute linear computations. The combinatorial RTSs, in contrast to this, just require the helping nodes to send the repairing node a sub-share, with no computation being required for any of the players.

In conclusion, if communication complexity if a priority and if the player's are able to compute linear combinations, RTSs based on MBR codes, such as those in in [15] are likely to be preferable. If communication complexity is a priority but players are unable to run computations, the combinatorial schemes in [18] compromise repairability and, occasionally, information rate and bandwidth, in order to achieve a repair algorithm that requires no computations from any of the players involved.

6.3 Comparison of techniques prioritising information rate

Both schemes based on MSR codes and the enrolment (and reduced enrolment) RTS prioritise information rate and offer universal repairability.

However, because of the insecure repair protocol, secure MSR codes are unable to achieve an information rate as good as the reduced enrolment RTS. Therefore, if information rate is a priority, the reduced enrolment scheme appears to be the best candidate. However, despite the improvements made to the enrolment RTS, resulting in the reduced enrolment RTS, the communication complexity remains much higher than any of the other schemes.

7 Conclusion

In this paper, we have explored RTSs. We introduced a new metric for analysis, which we used when exploring existing RTS constructions. The three constructions we considered include the enrolment scheme from [18], which we refined

to reduce the communication complexity, the combinatorial schemes from [18], which we used our new metric to analyse in more detail, and the GLF scheme in [15], which we later showed to be inefficient.

After exploring existing schemes, we studied applying the field of secure regenerating codes to RTSs. We discussed the two main types of regenerating codes, MBR and MSR codes, which result in RTSs prioritising either communication complexity or information rate, respectively. We discussed some immediate results of using these codes as RTSs, then explored constructions of both MBR and MSR codes and considered the resulting RTSs.

Finally, we compared the range of RTS constructions and discussed the results found. We concluded that, due to the insecure repair algorithm of MSR codes, MSR codes may not provide the best RTS schemes. We also concluded that the best candidate solution for an RTS prioritising communication complexity is an optimal MBR. In particular, the SRK-MBR construction [15] is one such scheme and manages to achieve information rates at least as good as the combinatorial schemes in [18], whilst maintaining universal repairability. The best candidate solution for an RTS prioritising information rate appears to be our refined version of the enrolment scheme, originally presented as an RTS in [18] and refined here in Section 4.1.3.

References

[1] J. C. Benaloh. Secret sharing homomorphisms: Keeping shares of a secret secret. In *Proceedings of Advances in Cryptology (CRYPTO): Conference on the Theory and Application of Cryptographic Techniques*, volume 263 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 1986.

[2] G. Blakley and G. Kabatianski. Ideal perfect threshold schemes and MDS codes. In *Information Theory, 1995. Proceedings., 1995 IEEE International Symposium on*, page 488. IEEE, 1995.

[3] G. R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the National Computer Conference (AFIPS)*, volume 48, pages 313–317, 1979.

[4] B. Chor and E. Kushilevitz. A communication-privacy tradeoff for modular addition. *Information Processing Letters*, 45(4):205–210, 1993.

[5] A. G. Dimakis, P. B. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.

[6] S. Goparaju, S. El Rouayheb, R. Calderbank, and H. V. Poor. Data secrecy in distributed storage systems under exact repair. In *Proceedings of International Symposium on Network Coding (NetCod)*, pages 1–6. IEEE, 2013.

[7] S. Goparaju, A. Fazeli, and A. Vardy. Minimum storage regenerating codes for all parameters. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 76–80, 2016.

[8] X. Guang, J. Lu, and F. W. Fu. Repairable threshold secret sharing schemes. *Computing Research Repository (CoRR): arXiv preprint, arXiv:1410.7190*, 2014.

[9] E. D. Karnin, J. W. Greene, and M. E. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, 29(1):35–41, 1983.

[10] M. Nojoumian. *Novel Secret Sharing and Commitment Schemes for Cryptographic Applications*. PhD thesis, University of Waterloo, 2012.

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

[12] S. Pawar, S. El Rouayheb, and K. Ramchandran. On secure distributed data storage under repair dynamics. In *Proceedings of the IEEE Symposium on Information Theory Proceedings (ISIT)*, pages 2543–2547. IEEE, 2010.

[13] K. V. Rashmi, N. B. Shah, and P. V. Kumar. Optimal exact-regenerating codes for distributed storage at the MSR and MBR points via a product-matrix construction. *IEEE Transactions on Information Theory*, 57(8):5227–5239, 2011.

[14] A. S. Rawat. A note on secure minimum storage regenerating codes. *Computing Research Repository (CoRR): arXiv preprint, arXiv:1608.01732*, 2016.

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

[16] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

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

[18] D. R. Stinson and R. Wei. Combinatorial repairability for threshold schemes. *Designs, Codes and Cryptography*, to appear.

[19] M. Ye and A. Barg. Explicit constructions of high-rate MDS array codes with optimal repair bandwidth. *IEEE Transactions on Information Theory*, 63(4):2001–2014, 2017.