Error Decodable Secret Sharing and One-Round Perfectly Secure Message Transmission for General Adversary Structures

K.M. Martin
Information Security Group
Royal Holloway, University of London
Egham, Surrey TW20 0TN, UK
keith.martin@rhul.ac.uk

M.B. Paterson
Dept. of Economics, Mathematics and Statistics
Birkbeck, University of London
Malet Street, London WC1E 7HX, UK
m.paterson@bbk.ac.uk

D.R. Stinson
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
dstinson@uwaterloo.ca

October 2, 2009

Abstract

An error decodable secret-sharing scheme is a secret-sharing scheme with the additional property that the secret can be recovered from the set of all shares, even after a coalition of participants corrupts the shares they possess. In this paper we consider schemes that can tolerate corruption by sets of participants belonging to a monotone coalition structure, thus generalising both a related notion studied by Kurosawa, and the well-known error-correction properties of threshold schemes based on Reed-Solomon codes. We deduce a necessary and sufficient condition for the existence of such schemes, and we show how to reduce the storage requirements of a technique of Kurosawa for constructing error-decodable secret-sharing schemes with efficient decoding algorithms.

In addition, we explore the connection between one-round perfectly secure message transmission (PSMT) schemes with general adversary structures and secret-sharing schemes, and we exploit this connection to investigate factors affecting the performance of one-round PSMT schemes such as the number of channels required, the communication overhead, and the efficiency of message recovery.

Keywords: secret sharing, perfectly secure message transmission, error correction

1 Introduction

A secret-sharing scheme takes a secret value and distributes related data (known as *shares*) to a set of participants so as to permit certain specified subsets of the participants to recover the secret, while preventing all other subsets from learning any information about the secret. Proposed independently in 1979 by Blakley [3] and Shamir [24], the original models considered adversaries that were passive in the sense that while they might attempt to learn the secret, they would not

otherwise interfere with the functioning of the scheme. In addition, the earliest schemes were constructed for classes of adversaries that are defined by a threshold value: any set of k or more participants is considered to be authorised to recover the secret, whereas a set of $k-1$ or fewer participants is deemed to be an adversarial coalition that should not be allowed to learn any information about the secret. Secret-sharing schemes secure against passive adversary structures defined by more general collections of subsets have received considerable attention (*e.g.* [13]).

Secret-sharing schemes have also been investigated for a number of different adversary models. Robust secret-sharing schemes allow the correct secret to be recovered even when some of the shares presented during an attempted reconstruction are incorrect. A framework for considering robust secret sharing is provided in [1], which includes schemes in both the information-theoretic and computationally secure settings. Secret-sharing schemes that either detect or identify participants who present incorrect shares during an attempted recovery have also been extensively studied, for example [17, 21, 22]. While such schemes make it apparent that cheating has occurred, they do not necessarily permit honest participants to recover the correct secret. Verifiable secret-sharing schemes have been proposed for environments where the shares given to participants may not be correct (the "dealer" of these shares may be corrupt), for example [5, 6]. These typically involve protocols that can be performed by various subsets of participants in order to check that the shares they possess are consistent in some sense. An overview of different secret-sharing adversary models can be found in [18].

In this paper we consider a type of information-theoretically secure robust secret-sharing scheme. There are two general approaches that can be taken in order to build such "robustness" into an information-theoretically secure secret-sharing scheme. The first approach involves issuing shares that contain additional information so that they can be "verified" in some way. This approach is taken by information-theoretically secure verifiable secret sharing schemes, as well as many of the secret-sharing schemes with error detection and correction capability. The second approach is to require additional shares, above and beyond what is strictly required to reconstruct the secret, in order to perform error correction [20, 23, 25, 26]. This is also the approach adopted by error decodable secret-sharing schemes, introduced by Kurosawa [16].

A secret-sharing scheme secure against a general passive adversary is said to be error decodable if it is still possible to recover the correct secret from the set of all shares even if the shares possessed by an adversarial subset are corrupted. Such schemes have previously been considered in the case of threshold adversaries [20,23,25,26]. Kurosawa demonstrates that in fact the question of whether or not a scheme is error decodable depends only on the properties of the adversary structure considered; for appropriate adversaries, the error-decoding ability essentially comes for free. What is not guaranteed, however, is an efficient algorithm for performing the decoding. In light of this, Kurosawa proposes a technique for converting any error decodable secret sharing scheme into one in which the decoding can be performed efficiently, at the cost of considerably increasing the size of the shares [16].

A primitive that turns out to be closely related to error decodable secret sharing is that of one-round perfectly secure message transmission (PSMT). The basic PSMT scenario consists of two users A and B who are connected by a number of distinct channels, some subset of which are controlled by an adversary. User A sends information to B over the channels, and the adversary can eavesdrop on the channels it controls, as well as potentially change the information that reaches B through those channels. A one-round PSMT scheme allows A to send a message to B by transmitting information over each channel in such a way that the adversary learns no information

about the message by eavesdropping on the channels it controls, and it cannot prevent B from recovering the message. As in the case for traditional secret sharing, PSMT was originally proposed for threshold adversaries [8], and has since been generalised (in the one-round case) to more general adversaries [7].

In this paper we consider both error decodable secret sharing and one-round PSMT in the even more general setting where the possible active and passive adversary coalitions are defined separately. (Such adversarial models have previously been considered in the case of verifiable secret sharing [9] and secure multi-party computation [19].) After providing background results and definitions relating to secret sharing in Section 2, in Section 3 we formally define error decodable secret sharing, and describe an adversary model that allows us to simultaneously generalise both Kurosawa's notion of error decodable secret sharing, and the error correction properties of Reed-Solomon codes. We give necessary and sufficient conditions on the adversary structures for a secret-sharing scheme to be error decodable in this model, analogous to conditions previously given in [7, 8, 11, 16, 19] (for example) for various related primitives.

In Section 4 we consider error-decodable schemes with efficient decoding algorithms. We show that Kurosawa's technique for constructing efficiently-decodable schemes can be applied in this more general setting, and we give techniques for modifying this construction in order to substantially reduce the sizes of the shares.

Section 5 contains an exploration of the precise relationship between one-round PSMT schemes for general adversary structures, and secret sharing. We describe necessary and sufficient conditions for the existence of a one-round PSMT scheme with a general adversary structure, as well as the conditions under which a secret-sharing scheme gives rise to a one-round PSMT scheme. We then consider the performance requirements of one-round PSMT, and we show how our consideration of error decodable schemes leads to a greater understanding of how to construct better one-round PSMT scheme for general adversary structures.

Finally, in Section 6 we specify some open problems in the study of error-decodable secret sharing and one-round PSMT schemes for general adversaries.

2 Secret Sharing

Let S be a set of n entities referred to as *participants* (for convenience of notation we will identify S with the set $\{1, 2, \ldots, n\}$). Let $\mathcal{P}(S) = \{A | A \subseteq S\}$ denote the set of all subsets of S. A collection $\Sigma \subseteq \mathcal{P}(S)$ of subsets of S is said to be a *monotone access structure* if Σ contains all sets $A' \in \mathcal{P}(S)$ satisfying $A \subseteq A'$ for some $A \in \Sigma$. The *complement* of Σ, denoted Σ^c, is the collection of subsets of S that are not in Σ, so $\Sigma^c = \mathcal{P}(S) \setminus \Sigma$. The subsets contained in Σ are *authorised sets*, and subsets contained in Σ^c are referred to as *unauthorised sets*.

Example 2.1. One widely-studied example of a monotone access structure is a (k, n)-*threshold access structure*, in which the authorised sets are all subsets of S containing at least k elements, and the unauthorised sets are those containing $k - 1$ or fewer elements.

Definition 2.2. A *secret-sharing scheme* realising an access structure Σ is an algorithm for assigning *shares* in some secret quantity s to each participant such that any authorised set of participants can use their combined shares to compute s. More formally, a secret-sharing scheme can be defined in terms of two algorithms:

- the share generation algorithm $\texttt{Gen}(s, \gamma)$ takes a secret s and randomness γ and outputs a list $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ of shares that are distributed to the n participants in S;
- the secret-recovery algorithm `Rec` takes a set of valid shares corresponding to an authorised set in Σ and outputs the secret s.

Schemes in which each unauthorised set is unable to determine any information about s are said to be *perfect*.

All further discussion of secret-sharing schemes in this paper is assumed to refer to perfect schemes unless otherwise specified.

It is known to be possible to realise any monotone access structure Σ by a *linear* secret-sharing scheme [13]:

Definition 2.3. Let $M = \{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^l\}$ be a set of l vectors in $\mathrm{GF}(q)^d$, where $l \geq d$ and $l \geq n$, and let φ be a function mapping $\{1, 2, \ldots, l\}$ to $\{1, 2, \ldots, n\}$. We say that (M, φ) is a *linear secret-sharing scheme realising the access structure* Σ if the vector $(1, 0, 0, \ldots, 0)$ is contained in the span of the vectors $\{\mathbf{m}^i | \varphi(i) \in A\}$ for any $A \in \Sigma$, yet is not contained in the span of the vectors $\{\mathbf{m}^i | \varphi(i) \in B\}$ for any $B \in \Sigma^c$.

In order to share a secret using a linear secret-sharing scheme, the secret is represented by an element $s \in \mathrm{GF}(q)$, which is used to construct a secret vector $\mathbf{r} \in \mathrm{GF}(q)^d$ by setting $r_1 = s$ then drawing r_i randomly from $\mathrm{GF}(q)$ for $i = 2, 3, \ldots, d$. The algorithm `Gen` outputs the share vector $\mathbf{v} = (v_1, v_2, \ldots, v_l) \in \mathrm{GF}(q)^l$ generated by setting $v_i = \mathbf{m}^i \cdot \mathbf{r}$; participant i is assigned a share consisting of the multiset $\{v_j | \varphi(j) = i\}$. If A is an authorised set, then there exist $\alpha_j \in \mathrm{GF}(q)$ such that $\sum_{\{j|\varphi(j)\in A\}} \alpha_j \mathbf{m}^j = (1, 0, 0, \ldots, 0)$ and then, by construction, $\sum_{\{j|\varphi(j)\in A\}} \alpha_j v_j = s$, hence the participants in A can recover the secret. Thus, for a linear secret-sharing scheme, both `Gen` and `Rec` consist solely of linear operations, and can be performed efficiently. It can be shown that for any unauthorised subset B, the participants in B obtain no information about the secret by combining their shares, and hence such schemes are perfect.

Example 2.4. A linear secret-sharing scheme realising a (k, n)-threshold access structure with $n \leq p$ and $k - 1 < p$ for some prime p can be constructed by taking $l = n$, defining $\varphi(i) = i$ for $i = 1, 2, \ldots, n$ and letting M be a subset of $\{(1, i, i^2, \ldots, i^{k-1}) | i \in \mathrm{GF}(p)^*\} \cup \{(0, 0, \ldots, 0, 1)\}$. Then each share v_j with $1 \leq j \leq p-1$ is obtained by evaluating $f(j) \pmod p$, where f is the polynomial $f = s + \sum_{i=1}^{k-1} r_i x^i \in \mathrm{GF}(p)[x]$; the value of v_p is simply given by r_{k-1} and the secret s is given by the y-intercept of f. An authorised set of shares gives the value of f at k or more points; since f has degree at most $k-1$ it can be interpolated from those values and its y-intercept recovered. The values of f at a set of $k-1$ or fewer points do not affect the likelihood of any element of $\mathrm{GF}(p)$ being the y-intercept of f, and thus an unauthorised set learns no information about s. This scheme was proposed by Shamir in 1979 [24]. We observe that the above description is readily generalised to give schemes with up to q participants defined over $\mathrm{GF}(q)$, where q is a power of a prime greater than $k - 1$.

In the scheme just described, both the secret and each of the shares consists of an element of $\mathrm{GF}(p)$, and hence have the same size. More generally, it can be shown that for any perfect secret sharing scheme, the size of each share is at least the size of the secret.

Definition 2.5. The *information rate* of a secret-sharing scheme can be defined informally as the ratio of the size of the secret to the size of the largest share. A secret-sharing scheme is said to be *ideal* if the size of each share is the same as the size of the secret, *i.e.* if it has information rate 1.

Thus Shamir's secret-sharing scheme is ideal, as is any linear secret-sharing scheme for which $l = n$ and φ is bijective.

3 Error Decodable Secret Sharing

Whereas secret sharing schemes are designed to be secure against unauthorised coalitions of participants that wish to learn the secret, an active adversary may still disrupt a scheme by corrupting some of the participants' shares, potentially making it impossible to recover the secret. In fact, it turns out that for appropriate access structures, the corresponding secret-sharing schemes do offer a degree of resistance against such attacks, which leads us to the notion of *error decodable secret sharing*. We will now examine two notions of error decodability that have appeared in the literature, before seeing how we can simultaneously unify and generalise them.

3.1 (k, n)-Threshold Schemes and Reed-Solomon Codes

It was observed as early as 1981 that an ideal (k, n)-threshold secret-sharing scheme can be interpreted as a *Reed-Solomon code* [20].

Definition 3.1. Let q be a prime power and n a positive integer with $n \leq q+1$. Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be distinct elements of $\mathrm{GF}(q) \cup \{\infty\}$. Let $\mathcal{C}$ be the length n code over $GF(q)$ defined by setting

$$\mathcal{C} = \{(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_n)) \,|\, f \in \mathrm{GF}(q)[x],\ \deg f < k\}\,,$$

where $f(\infty)$ is defined to be the coefficient of x^{k-1} in f. Then $\mathcal{C}$ is said to be an $[n, k, n-k+1]$ *Reed-Solomon code*. Such codes are known to be linear codes with dimension k and minimum distance $n-k+1$, which implies they are MDS codes [12].

Consider Shamir's (k, n)-threshold secret-sharing scheme. If we take all possible secret vectors $\mathbf{r} \in \mathrm{GF}(q)^k$, then the corresponding share vectors $\mathbf{v}$ are precisely the words of an $[n, k, n-k+1]$ Reed-Solomon code, by construction. Since a Reed-Solomon code has minimum distance $n-k+1$, then if $n - k \geq 2t$ it can be used to correct up to t errors. Therefore, given the vector of shares possessed by the participants in a (k, n) threshold scheme we can recover the corresponding secret, even if up to t of the participants' shares have been corrupted (*i.e.* replaced by a different element of $\mathrm{GF}(q)$), provided $n - k \geq 2t$ [20]. Techniques for performing such error correction for threshold secret-sharing schemes have received a certain amount of attention in the literature [23, 25, 26].

3.2 Kurosawa's Error Decodable Secret-Sharing Schemes

The notion of *error decodable secret-sharing schemes* for more general access structures was proposed by Kurosawa in [16]. According to his definition, a secret-sharing scheme realising an access structure Σ is error decodable if the secret can be recovered from the set of all participants' (possibly corrupted) shares, provided that the set of all participants whose shares have been corrupted is an unauthorised set. This is expressed more precisely in the following definition:

Definition 3.2. A secret-sharing scheme is said to be *error decodable* if the following decoding algorithm always returns the correct secret when given an input consisting of a list of shares $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ that has been corrupted in positions corresponding to the members of an unauthorised set.

Algorithm 3.3.

1. *For each possible secret s and randomness γ, compute the corresponding list $\mathbf{v}$ of shares and determine whether $\{j | v_j \neq t_j\} \in \Sigma^c$; if that is the case, we say s is a* candidate secret.
2. *If there is precisely one candidate secret s, then return s as the secret.*
3. *If there is no candidate secret, or if there exist candidate secrets s_1 and s_2 with $s_1 \neq s_2$, return $\perp$.*

Kurosawa demonstrates that a secret-sharing scheme realising an access structure Σ is error decodable precisely when the access structure satisfies a condition known as Q^3, first defined by Hirt and Maurer in the context of secure multiparty computation [11].

Definition 3.4. [11] A monotone access structure Σ for a set S of participants is said to satisfy property Q^3 if $B_1 \cup B_2 \cup B_3 \neq S$ for any $B_1, B_2, B_3 \in \Sigma^c$.

Example 3.5. A (k, n)-threshold scheme satisfies property Q^3 if and only if $n > 3(k-1)$. Equivalently, a (k, n)-threshold scheme is error decodable if and only if $n - k \geq 2(k-1)$. This agrees with the fact that an $[n, k, n-k+1]$ Reed-Solomon code can correct $k-1$ errors if and only if $n - k \geq 2(k-1)$.

3.3 General Adversary Structures and Γ-Error Decodable Secret Sharing Schemes

The case of Reed-Solomon decoding of a (k, n)-threshold scheme involves schemes that can withstand coalitions of size at most $\lfloor \frac{n-k}{2} \rfloor$ of participants who corrupt their shares. Kurosawa considered schemes secure against coalitions of players in Σ^c, where Σ is the access structure of the scheme. More generally, we define a *monotone adversary structure* for a set S of participants to be a collection $\Gamma \subseteq \mathcal{P}(S)$ with the property that if $B' \subset B$ for some $B \in \Gamma$, then $B' \in \Gamma$. We refer to elements of Γ as *coalition sets*. Then Σ^c is an example of a monotone adversary structure, as is the collection of subsets of S containing at most t participants.

Definition 3.6. Let Σ and Γ be a monotone access structure and a monotone adversary structure respectively for a set S of participants. Consider Algorithm 3.3 with Σ^c replaced by Γ in step 1. We say that a secret-sharing scheme realising the access structure Σ is Γ-*error decodable* if this algorithm always returns the correct secret when given an input consisting of some list of shares output by `Gen`, but with precisely those shares possessed by the members of some set in Γ having been corrupted. (We will understand the term 'error decodable' to mean Γ-error decodable with $\Gamma = \Sigma^c$; in this case our definition coincides with that of Kurosawa.)

In this scenario, the role played by the condition Q^3 for Kurosawa's schemes is filled by a variant of this condition, leading to the following theorem.

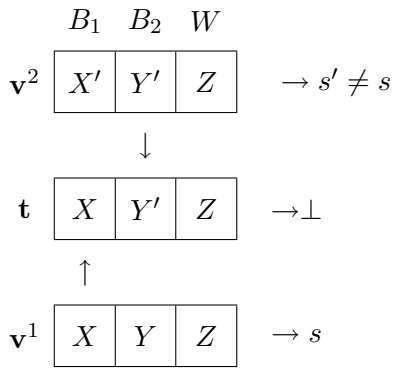

Figure 1: Decoding fails if $Q(\Gamma, \Gamma, \Sigma^c)$ is not satisfied.

Theorem 3.7. *Let Σ and Γ be a monotone access structure and monotone adversary structure respectively for a set S of participants. A secret-sharing scheme realising the access structure Σ is Γ-error decodable if and only if for all $W_1, W_2 \in \Gamma$ and all $B \in \Sigma^c$, it holds that $W_1 \cup W_2 \cup B \neq S$. We refer to this as condition $Q(\Gamma, \Gamma, \Sigma^c)$.*

Proof. Suppose that there exist $W_1, W_2 \in \Gamma$ and $B \in \Sigma^c$ with $W_1 \cup W_2 \cup B = S$. Without loss of generality, we can suppose that W_1, W_2 and B are pairwise disjoint. As B is an unauthorised set, there exist secrets s_1 and $s_2 \neq s_1$ and randomness γ_1 and γ_2 such that the share lists $\texttt{Gen}(s_1, \gamma_1) = \mathbf{v}^1$ and $\texttt{Gen}(s_2, \gamma_2) = \mathbf{v}^2$ satisfy $v_i^1 = v_i^2$ for all $i \in B$ (otherwise the shares given to members of B would uniquely determine the secret). Let $\mathbf{t} = (t_1, t_2, \ldots, t_n)$ be the corrupted list of shares defined by

$$t_i = \begin{cases} v_i^1 & i \in W_1, \\ v_i^2 & i \in W_2, \\ v_i^1 = v_i^2 & i \in B. \end{cases}$$

Then $\mathbf{t}$ differs from both $\mathbf{v}^1$ and $\mathbf{v}^2$ in positions corresponding to shares possessed by a set belonging to Γ, hence Algorithm 3.3 returns $\perp$. (This is depicted schematically in Figure 1.)

Conversely, suppose the scheme is not Γ-error decodable. Then there exists some corrupted list of shares $\mathbf{t}$, and valid lists of shares $\mathbf{v}^1$ and $\mathbf{v}^2$ whose corresponding secrets s_1 and s_1 are not equal, such that $W_1 = \{j \in S | v_j^1 \neq t_j\} \in \Gamma$ and $W_2 = \{j \in S | v_j^2 \neq t_j\} \in \Gamma$ (so the decoding algorithm outputs $\perp$ when given $\mathbf{t}$ as input).

Let $B = S \setminus (W_1 \cup W_2)$. Then the shares in $\mathbf{t}$ belonging to participants in B are the same as those in both $\mathbf{v}^1$ and $\mathbf{v}^2$. If B were an authorised set, then the shares possessed by participants in B would suffice to recover the secret. However, as $\mathbf{v}^1$ and $\mathbf{v}^2$ are valid share vectors corresponding to different secrets, this cannot be the case. Hence $B \in \Sigma^c$, and thus condition $Q(\Gamma, \Gamma, \Sigma^c)$ is violated. □

This approach serves to unify and also generalise Kurosawa's concept of error decodable secret sharing schemes, and the error correction properties of Reed-Solomon codes. (In the case considered by Kurosawa, the condition Q^3 is precisely condition $Q(\Sigma^c, \Sigma^c, \Sigma^c)$ in our notation.) It both allows the consideration of schemes that can withstand more powerful corruption adversaries, as well as permitting potential efficiency gains in scenarios where the adversary is more constrained.

We observe that it is not always necessary to use the entire (corrupted) list of shares in the decoding process, as illustrated by the following corollary to Theorem 3.7:

Corollary 3.8. *Suppose there exists a Γ-error decodable secret sharing scheme realising the access structure Σ on a set S of participants. Let T be a subset of S, let Σ_T consist of the collection of sets in Σ that are contained in T, and let Γ_T consist of the collection of sets in Γ that are contained in T. Then it is possible to recover the secret s by applying Algorithm 3.3 to the set of shares possessed by the participants in T, if and only if condition $Q(\Sigma_T^c, \Sigma_T^c, \Gamma_T)$ is satisfied.*

4 Efficiently Decodable Schemes

The decoding technique described in Algorithm 3.3 serves to determine when error decoding of a secret scheme is theoretically possible, but it does not represent a practical means of actually carrying out the decoding. Ideally we would like decoding techniques whose running time is polynomial in the number of participants in the scheme. In the case where Γ consists of all subsets of S of size at most t for some positive integer t, then the set of all valid share vectors in an ideal Γ-error decodable secret sharing scheme is an error-correcting code that can correct t errors. For a (k, n)-threshold scheme this code is a Reed-Solomon code, and thus there are known techniques for performing the decoding in an efficient manner [12]; however, the problem of decoding a general linear code is known to be NP-complete [2]. Thus for more general error decodable secret sharing schemes there is no guarantee that an efficient decoding technique exists.

Kurosawa has proposed a technique for transforming an error decodable secret-sharing scheme into one that can be decoded in time polynomial in l and in the bitlength of q, at the expense of significantly larger share sizes[1] [16]. We will now describe this construction, before considering techniques for reducing the storage requirements when addressing more general adversary structures.

4.1 Kurosawa's Polynomial-Time Error Decodable Scheme

Kurosawa proposes a two-level approach for converting an error decodable secret-sharing scheme into one that can be decoded in polynomial time. Kurosawa [16] used the notation of a linear secret-sharing scheme, and we present his scheme here in the same setting. However, we observe that the construction can be adapted for any secret-sharing scheme with efficient `Gen` and `Rec` algorithms.

Scheme 4.1. *Let (M, φ) be a linear secret sharing scheme realising a monotone access structure Σ that satisfies property Q^3.*

level 1 *The scheme (M, φ) is used to generate a share vector $\mathbf{v}$ corresponding to a secret $s \in \mathrm{GF}(q)$.*

level 2 *For $i = 1, 2, \ldots, l$ the element v_i is converted into a new secret vector $\mathbf{w}^i$ (with $w_1^i = v_i$), and the scheme (M, φ) is used to generate a corresponding share vector $\mathbf{u}^i$. The vector $\mathbf{w}^i$ is allocated to participant $\varphi(i)$, and u_m^i is allocated to participant $\varphi(m)$ for each $m = 1, 2, \ldots, l$. Thus participant j receives the share $\bigcup_{i=1}^{l} \{u_m^i | \varphi(m) = j\} \cup \{\mathbf{w}^m | \varphi(m) = j\}$.*

[1] Ideally we would like to be able to decode in time polynomial in n, which requires l itself to be polynomial in n. This is the case for most known dedicated constructions of linear secret-sharing schemes for specific access structures, such as Shamir's threshold secret-sharing scheme. However, generic constructions for arbitrary access structures tend to result in a value of l that is exponential in n.

decoding algorithm *For each $i = 1, 2, \ldots, l$ the level 2 share vector corresponding to the secret vector $\mathbf{w}^i$ (as possessed by user $\varphi(i)$) is computed. This vector is compared with the vector of level 2 shares u^i_j possessed by the other participants. If they differ in positions corresponding to the participants in some set that is not in Σ^c, then the level 1 share $v^i = w^i_1$ is deemed to have been corrupted. If the set of users with corrupted level 1 shares is contained in Σ^c then the remaining uncorrupted level 1 shares are used to recover s, otherwise the output is $\perp$.*

The role of the level 2 shares is essentially to provide an authenticity check for the level 1 shares. As they are assigned using (M, φ), it follows that they do not leak any additional information about the level 1 shares to any unauthorised sets. This construction can be applied to any error decodable secret sharing scheme, although at the cost of the participants having to store the random elements of the level 2 secret vectors, together with the level 2 shares. This results in the share size of each participant being $d + l$ times that of the original scheme.

4.2 Efficient Decoding for General Adversary Structures

Kurosawa's construction can in fact be applied *mutatis mutandis* to any Γ-error decodable secret sharing scheme to yield a scheme that can be decoded in polynomial time. However, the high storage requirement is something of a drawback. We now consider some ways in which this storage can be substantially reduced, in both Kurosawa's scheme and in schemes with general adversary structures.

4.2.1 Reducing the Number of Level 2 schemes Required

In Kurosawa's scheme, each element of the level 1 share vector was re-shared using a separate level 2 scheme. However, this is not strictly necessary: the level 2 schemes are used only to identify uncorrupted level 1 shares, so we simply need to guarantee that we can always find an authorised set whose shares are confirmed to be uncorrupted.

Lemma 4.2. *Let $A \subseteq S$ be the set of users whose level 1 shares are shared among the members of S by means of a level 2 scheme. Then the decoding method of Scheme 4.1 succeeds provided that set A has the property that for all $W \subseteq A$ with $W \in \Gamma$ we have $A \setminus W \in \Sigma$. (We will refer to any set with this property as a* decoding set*.)*

Proof. The set of participants in A whose shares are corrupted by a coalition is of the form $A \cap W$ for some $W \in \Gamma$. The level 1 shares possessed by users in $A \setminus W$ will be designated as uncorrupted by the decoding algorithm of Scheme 4.1, since their secret vectors $\mathbf{w}$ are unaffected, and their level 2 shares are only corrupted in positions corresponding to users in W. Thus if $A \setminus W \in \Sigma$ then there exists an authorised set of participants whose shares are known to be uncorrupted, which can thus be used to recover the desired secret. □

Note that in the case where there exists some $W \in \Gamma$ such that $A \setminus W \notin \Sigma$, then the shares of users in $A \setminus W$ do not suffice to recover the secret and the decoding algorithm of Scheme 4.1 fails.

If we wish to minimise the number of level 2 schemes required, it suffices to choose a decoding set A for which $|\{i|\varphi(i) \in A\}|$ is minimised. For ideal schemes we have the following corollary:

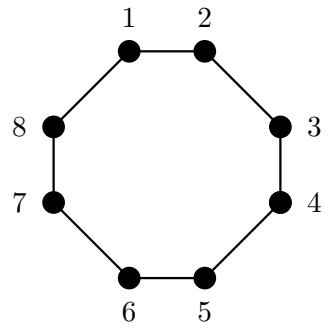

Figure 2: A graph determining an access structure on a set of eight participants

Corollary 4.3. *If (M, φ) is an ideal Γ-error decodable secret-sharing scheme then the number of level 2 schemes required for successful instantiation of Scheme 4.1 is upper bounded by*

$$1 + \max_{W \in \Gamma} |W| + \max_{B \in \Sigma^c} |B|.$$

Proof. If $1 + \max_{W\in\Gamma}|W| + \max_{B\in\Sigma^c}|B| \geq |S|$ then the result is trivially true. Suppose otherwise, and let $A \subset S$ be a set of size $1 + \max_{W\in\Gamma}|W| + \max_{B\in\Sigma^c}|B|$. For any set $W \in \Gamma$, we have that $|A \setminus W| \geq 1 + \max_{B\in\Sigma^c}|B|$, so $A \setminus W$ is not in Σ^c, and hence is a qualified set. Thus A is a decoding set, and the result follows. □

For many combinations of Σ and Γ this represents a substantial saving, as illustrated by the following examples.

Example 4.4. Suppose Σ is a (k, n)-threshold access structure for a set S of participants, and Γ is an adversary structure consisting of all subsets of S of size at most t. Kurosawa's scheme requires each participant to store at least $2t + w$ level 2 shares. However, if t shares out of any set of shares of size $t + w$ are corrupted, then the remaining set of shares is an authorised set, thus it suffices for each participant to possess $t + w$ level 2 shares.

Example 4.5. Suppose S consists of eight participants, and consider the access structure Σ consisting of all subsets of S that contain one of the two-element subsets of S given by the edges of the graph depicted in Figure 2. If we wish to protect against adversaries consisting of single participants that corrupt their shares, it suffices to provide level 2 sharings of the secrets corresponding to participants 1, 2, 3 and 4 (for example) in order to ensure that the decoding algorithm of Scheme 4.1 succeeds. For, no matter which of these participants acts as an adversary, there exist two consecutively numbered participants within this set, and their shares can be used to recover the secret since they form an authorised set. This reduces the number of level 2 schemes by a half relative to Kurosawa's technique.

Example 4.6. The scheme described in Example 4.5 can be generalised to one suitable for a set of n participants with an access structure defined by the graph consisting of a cycle on n vertices, and an adversary that compromises up to t shares, where $2t < \lfloor n/2 \rfloor$. In this case it suffices to deploy level 2 sharings of the level 1 shares corresponding to $2t + 2$ consecutive participants, rather than for all n participants.

4.2.2 Reducing the Size of the Level 2 Shares

The role of the level 2 shares is not to enable recovery of the level 1 shares (since these are already possessed by the participants), but rather to authenticate the level 1 shares so as to determine whether they have been corrupted. Therefore, it may not be necessary to use the same scheme (M, φ) for the level 2 share distribution as was used in level 1. In particular, we have the following:

Theorem 4.7. *In Scheme 4.1 it suffices for the level 2 shares to be assigned using any (possibly non-perfect) secret-sharing scheme with the following properties:*

1. *Sets of participants in Σ^c learn no information about the secret.*
2. *For any two adversary sets $W_1, W_2 \in \Gamma$, the participants in $S \setminus (W_1 \cup W_2)$ should be able to recover the secret.*

Proof. Consider a level 1 share belonging to user i that is shared by means of a level 2 scheme. If each set $B \in \Sigma^c$ with the property that $B \cup \{i\} \in \Sigma^c$ learns no information about that level 1 share then the perfectness of the level 1 scheme is not compromised. If the level 2 scheme satisfies property 1 then this will certainly hold.

Property 2 is necessary in order prevent a scenario analogous to that depicted in Figure 1: if a single corrupted share vector can be obtained from either of two different secrets by corrupting shares belonging to sets of participants in Γ, then it is not possible to determine from the level 2 shares whether the corresponding level 1 share has been corrupted. □

For many combinations of Σ and Γ, there will be sets of participants that are neither in Σ^c, nor of the form $S \setminus (W_1 \cup W_2)$ for any $W_1, W_2 \in \Gamma$. Such sets do not have to be able to obtain the secret, but there is no actual need to restrict the amount of information they can obtain about the secret. Thus it is not necessary to use a perfect scheme for distributing the level 2 shares, which potentially allows us to use more efficient schemes. Recall that in a perfect secret sharing scheme, the size of each user's share must be at least as big as the size of the secret. For schemes that are not perfect, this bound does not apply [4]; an example of this is given by the class of schemes known as *ramp schemes*:

Definition 4.8. A (t_1, t_2, n)-*ramp scheme* is a secret-sharing scheme in which any set of t_2 or more participants are able to recover the secret, yet any set of at most t_1 participants learns no information about the secret.

Example 4.9. Shamir's (k, n)-threshold scheme can be generalised to give a (t_1, t_2, n)-ramp scheme by defining the secret to consist of the coefficients of x^i in a polynomial f of degree at most t_2 for $i = 0, 1, \ldots, t_2 - t_1 - 1$, with shares generated as before (see Example 2.4). Then any set of t_2 or more users can use interpolation of f to recover the secret, and for any set of fewer than t_1 users the space of possible secrets is not decreased. Note that this scheme allows the sizes of each share to be reduced by a factor of $t_2 - t_1$ relative to the size of the secret, when compared with a (t_2, n)-threshold scheme [15].

If we are applying the construction of Scheme 4.1 to an ideal secret-sharing scheme then letting t_1 be the maximum size of a set in Σ^c and t_2 be the minimum size of a set of the form $S \setminus (W_1 \cup W_2)$ with $W_1, W_2 \in \Gamma$, whenever $t_1 < t_2$ we can use a (t_1, t_2, n)-ramp scheme to distribute the level 2 shares. If t' is the maximum size of a set in Γ then we have $t_2 \geq n - 2t'$.

Example 4.10. Suppose Σ is a (k, n)-threshold access structure for a set S of participants, and Γ is an adversary structure consisting of all subsets of S of size at most t, for some $t < k$. A perfect secret-sharing scheme realising Σ requires each user to store a share that is as least as big as the size of the secret s. However, from the above discussion we see that for the level 2 schemes in Scheme 4.1 it suffices to use a (k, t, n)-ramp scheme. Using the scheme described in Example 4.9 permits a reduction in the size of the level 2 shares by a factor of $k - t$ relative to Kurosawa's scheme.

Example 4.11. Let (M, φ) be a Γ-error decodable secret-sharing scheme where S, Γ and Σ are as defined in Example 4.5. Then the largest set in Σ^c contains four elements, since any set of five vertices in the graph depicted in Figure 2 contains at least two that are connected by an edge. Furthermore, each set in Γ has size at most 1, so the smallest subset of S of the form $S \setminus (W_1 \cup W_2)$ with $W_1, W_2 \in \Gamma$ contains six elements. Thus we can use a $(4, 6, 8)$-ramp scheme (rather than the scheme (M, φ)) to distribute the level 2 shares. This requires at most half the storage of any perfect secret-sharing scheme realising Σ. Combining this with the observations of Example 4.5, we see that we can reduce the storage required for the level 2 shares by at least a factor of four relative to Kurosawa's technique for this combination of Σ and Γ.

Example 4.12. Considering the generalisation of Example 4.11 to n participants and adversaries that corrupt up to t shares (as in Example 4.6), we note that the largest set in Σ^c has $\lceil n/2 \rceil$ elements, and the smallest subset of the form $S \setminus (W_1 \cup W_2)$ with $W_1, W_2 \in \Gamma$ contains $n - 2t$ elements. Hence in this case we can use a $(n - 2t, \lceil n/2 \rceil, n)$-ramp scheme, thus cutting storage requirements by a factor of at least $\lfloor n/2 \rfloor - 2t$.

5 Secret Sharing and Perfectly Secure Message Transmission

One application of error decodable secret-sharing schemes described by Kurosawa is the construction of one-round perfectly secure message transmission (PSMT) schemes. First proposed by Dolev, Dwork, Waarts and Yung [8], the one-round PSMT primitive is closely related to secret sharing, although the underlying scenario is somewhat different. In Subsection 5.1 we give basic definitions and existence results for one-round PSMT schemes for general adversary structures. In Subsection 5.2 we examine the connection between one-round PSMT and secret sharing in more detail. Finally, in Subsection 5.3 we explore the implications of our results on error decodable secret sharing for the construction of efficient PSMT schemes.

5.1 One-Round PSMT for General Adversary Structures

Definition 5.1. A *one-round* (n, t)*-PSMT scheme* is an algorithm permitting a party A to transmit a message s to a party B by sending information over n channels such that

- B can correctly recover s even if an adversary changes the information passing through up to t of those channels;
- the adversary learns no information about s from the information that A sent over the compromised channels.

We will refer to the information $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ that is sent over the n channels as an *encoding* of the message s.

Dolev *et al.* showed that a (n, t)-PSMT scheme exists if and only if $n \geq 3t + 1$. As was the case for secret sharing, this original definition involves a threshold adversary structure (although, in contrast, the adversary is assumed to be active). Desmedt, Wang and Burmester studied the generalisation of PSMT to schemes secure against an arbitrary montone adversary structure [7]. For the adversary structure Γ, Desmedt *et al.* show that a one-round PSMT scheme is possible if and only if condition $Q(\Gamma, \Gamma, \Gamma)$ holds; in the case where Γ is a threshold access structure, this reduces to the result of Dolev *et al.*

As in the case of error decodable secret sharing, it is possible to generalise the definition of a one-round PSMT scheme further, by defining separately the adversary structures corresponding to active adversaries who can corrupt transmitted data, and passive adversaries who must be prevented from obtaining information about the message s:

Definition 5.2. Let Γ and Σ^c be monotone adversary structures defined on a set S of size n, corresponding to n channels connecting a party A to a party B. A *one-round* (Γ, Σ^c)*-PSMT* is an algorithm permitting A to transmit a message s to B by sending information over the channels such that

- B can correctly recover s even if an adversary changes the information passing through a subset $W \in \Gamma$ of the channels;
- an adversary who eavesdrops on a subset $D \in \Sigma^c$ of the channels learns no information about s from the information that A sent over the channels in D.

Such "mixed" adversary structures have previously been considered in the case of verifiable secret sharing [9], and secure multi-party computation [19]. The following theorem is an analogue for this more general case of the result obtained by Desmedt *et al.*:

Theorem 5.3. *A one-round* (Γ, Σ^c)*-PSMT scheme exists if and only if condition* $Q(\Gamma, \Gamma, \Sigma^c)$ *is satisfied.*

Proof. For any monotone access structure Σ there exists a secret-sharing scheme realising Σ, and if condition $Q(\Gamma, \Gamma, \Sigma^c)$ is satisfied then any such secret-sharing scheme is Γ-error decodable. In order to turn the secret-sharing scheme into a one-round (Γ, Σ^c)-PSMT scheme, the channels are identified with the participants in the secret-sharing scheme, and party A simply generates shares corresponding to the secret s, and sends each share over the appropriate channel. Then the shares sent along channels corresponding to a set in Σ^c do not leak any information about the secret, and party B can use a decoding algorithm for the secret-sharing scheme in order to recover s.

Now suppose condition $Q(\Gamma, \Gamma, \Sigma^c)$ does not hold. The argument used in the first part of the proof of Theorem 3.7 does not make use of the fact that authorised sets of users can recover the secret, and hence can be directly applied in this PSMT context to show that unique recovery of the message is not always possible. □

Combining Theorems 3.7 and 5.3 we obtain the following corollary:

Corollary 5.4. *A one-round* (Γ, Σ^c)*-PSMT scheme exists if and only if there exists a* Γ*-error decodable secret-sharing scheme realising the access structure* Σ*.*

Based on this result it would be tempting to conclude that one-round (Γ, Σ^c)-PSMT and Γ-error decodable secret sharing for an access structure Σ are one and the same; however, this is not quite the case. In the following section we investigate more closely the relationship between these two primitives.

5.2 Connecting One-Round PSMT and Secret Sharing

We have seen in Corollary 5.4 that the necessary and sufficient conditions for the existence of a Γ-error decodable secret-sharing scheme realising an access structure Σ are exactly the same as the necessary and sufficient conditions for the existence of a (Γ, Σ^c)-PSMT scheme, and in the proof of Theorem 5.3 we saw that every Γ-error decodable secret-sharing scheme realising an access structure Σ gives rise to a (Γ, Σ^c)-PSMT scheme. However, this does not mean that two notions are "equivalent" –although a Γ-error decodable secret sharing scheme can be transformed into a (Γ, Σ^c)-PSMT scheme, the converse is not true. A clue as to why this is the case is provided by the following lemma.

Lemma 5.5. *If a (Γ, Σ^c)-PSMT scheme encodes a message s as $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ (prior to transmission) then it is possible to recover s from $\{v_i | i \in T\}$ for any set T of the form $T = S \setminus (W_1 \cup W_2)$, where $W_1, W_2 \in \Gamma$.*

Proof. This result can be proved in a similar manner to Theorem 3.7. Assume there are sets $W_1, W_2 \in \Gamma$ (without loss of generality we suppose they are disjoint), and secrets s_1 and $s_2 \neq s_1$ encoded as $\mathbf{v}^1$ and $\mathbf{v}^2$ with $v_i^1 = v_i^2$ for all $i \in S \setminus (W_1 \cup W_2)$. Then the word $\mathbf{v}$ defined by

$$v_i = \begin{cases} v_i^1 & i \in W_1 \\ v_i^2 & i \in W_2 \\ v_i^1 = v_i^2 & i \in S \setminus (W_1 \cup W_2) \end{cases}$$

could arise from corruptions of either $\mathbf{v}^1$ or $\mathbf{v}^2$, and hence B cannot recover a unique message.

Thus, for a (Γ, Σ^c)-PSMT scheme it must be the case that for any set of the form $T = S \setminus (W_1 \cup W_2)$, where $W_1, W_2 \in \Gamma$, the values corresponding to channels in T of any encoding of a message must uniquely determine that message. □

Thus for a one-round (Γ, Σ^c)-PSMT scheme we see that the data sent on sets of channels in Γ can be corrupted, the data sent on sets of channels in Σ^c should not yield any information about the message, and the data sent on sets of channels of the form $S \setminus (W_1 \cup W_2)$ with $W_1, W_2 \in \Gamma$ permits the message to be recovered (prior to any corruption). However, there are *no conditions placed on sets of channels that are neither in Σ^c nor of the form $S \setminus (W_1 \cup W_2)$* with $W_1, W_2 \in \Gamma$. This is in contrast to the case of Γ-error decodable secret sharing, in which every set of participants that is not in Σ^c is an authorised set and hence must be able to reconstruct the secret (provided their shares are not corrupted). Combining Lemma 5.5 and Definition 5.2 we obtain the following theorem:

Theorem 5.6. *A one-round (Γ, Σ^c)-PSMT scheme is equivalent to a (not necessarily perfect) secret-sharing scheme in which the authorised sets are those of the form $S \setminus (W_1 \cup W_2)$ with $W_1, W_2 \in \Gamma$, and the unauthorised sets belong to Σ^c.*

A Γ-error decodable secret-sharing scheme realising Σ has the same (active) adversary structure and unauthorised sets, however it satisfies the extra condition of being a perfect secret-sharing scheme. This is thus the essential difference between the two primitives. We will see in Subsection 5.3.2 that this has significant implications for the amount of information that must be transmitted in such schemes.

Example 5.7. Suppose $n \geq 3t+1$. In a one-round (n,t)-PSMT scheme the data transmitted over any set of t or fewer channels must reveal no information about the message s. Furthermore, by Lemma 5.5, the data transmitted over any set of $n-2t$ or more channels must suffice to recover the secret. Thus, when interpreted as a secret-sharing scheme, a one-round (n,t)-PSMT scheme defines a $(t, n-2t, n)$-ramp scheme. However, for Γ and Σ^c consisting of all subsets of S of size t or less, the corresponding Γ-error decodable secret-sharing scheme realising Σ is a $(t+1, n)$-threshold scheme.

5.3 Efficient One-Round PSMT Schemes

There are three significant aspects to the efficiency of a one-round PSMT scheme: the number of channels required, the amount of information transmitted through each channel, and the efficiency with which A can encode the message and B can recover it. We now consider how the above discussion casts light on each of these properties.

5.3.1 Number of Channels Required for One-Round (Γ, Σ^c)-PSMT

The focus of Desmedt *et al.* when considering one-round PSMT was on minimising the total number of channels required to implement such a scheme [7]. In our more general setting we have the following result:

Theorem 5.8. *Let Γ be an adversary structure representing passive adversaries and Σ^c an adversary structure representing passive adversaries on a set S of n channels. Then the minimum number of these channels required for the successful implementation of one-round (Γ, Σ^c)-PSMT is equal to the size of the smallest subset $T \subseteq S$ for which the property $Q(\Gamma_T, \Gamma_T, \Sigma^c_T)$ holds.*

Proof. This follows directly from Theorem 5.3. □

Desmedt *et al.* obtained the analogous result in the case where $\Gamma = \Sigma^c$ [7]. The following corollary is easily derived.

Corollary 5.9. *An upper bound on the minimum number of channels required for the successful implementation of one-round (Γ, Σ^c)-PSMT is given by*

$$1 + 2\max_{W \in \Gamma}|W| + \max_{B \in \Sigma^c}|B|.$$

In the case where Γ and Σ^c both consist of sets of size at most t this reduces to the result, proven in [8], that one-round (n,t)-PSMT is possible if and only if $n \geq 3t+1$.

5.3.2 Amount of Information Transmitted During One-Round (Γ, Σ^c)-PSMT

The amount of communication required to carry out one-round (Γ, Σ^c)-PSMT is a significant factor affecting the practical performance of a scheme. It is often specified in terms of a quantity referred to as the *overhead* of the scheme [10].

Definition 5.10. The *overhead* of a one-round (Γ, Σ^c)-PSMT scheme is defined to be the total amount of information transmitted over all the channels divided by the size of the message s.

The overhead of a one-round PSMT scheme is a direct consequence of the size of the shares in the corresponding secret-sharing scheme, although in the secret-sharing literature it is usually described in terms of the *average information rate*, which is the size of the secret divided by the mean of the sizes of the shares. (Thus the overhead of a one-round PSMT scheme is equal to n times the reciprocal of the average information rate of the corresponding secret-sharing scheme.) The problem of minimising the information rate and/or the average information rate of a secret-sharing scheme has received much attention in the secret-sharing literature. The fact that the secret-sharing schemes required for one-round PSMT do not necessarily have to be perfect can allow for additional savings to be made.

Example 5.11. In order to demonstrate that it is always possible to implement a PSMT scheme when $Q(\Gamma, \Gamma, \Gamma)$ is satisfied, Desmedt *et al.* describe a construction that is equivalent to the construction in [13] for (perfect) secret-sharing schemes realising general access structures. Kurosawa points out that in the case of a threshold adversary, this results in the need to transmit considerably more information over each of the channels in T than if an ideal threshold scheme were used [16], a result whose secret-sharing analogue is well-known (see [14], for example). If an ideal threshold scheme is used then the resulting overhead is precisely n. However, it is possible to do better than this, as illustrated by the following corollary to Theorem 5.6, which re-proves a result that appears in [10].

Corollary 5.12. *The optimal overhead of a one-round (n, t)-PSMT scheme is $n/(n - 3t)$.*

Proof. As we observed in Example 5.7 a one-round (n, t)-PSMT scheme, is precisely a $(t, n-2t, n)$-ramp scheme. The ramp scheme described in Example 4.9 leads to shares whose size is $1/((n-2t)-t)$ times the size of the secret, which is known to be optimal [15]; the result follows directly. □

For scenarios in which the cost of communication is substantial relative to the cost of computation for user B, when constructing a one-round PSMT scheme it may suffice to choose the most efficient secret-sharing scheme known for the appropriate access structure. However, in general we have also to consider the computational effort required for B to recover the message. We address this question in Subsection 5.3.3.

5.3.3 One-Round (Γ, Σ^c)-PSMT with Efficient Message Recovery

In the case of PSMT based on Reed-Solomon codes, the efficient decoding algorithms that exist for such codes provide a means for B to recover the message transmitted by A. However, for adversary structures other than the threshold structure, it is not necessarily known whether there exists a corresponding PSMT scheme with efficient message recovery (this issue was not considered in [7]).

The construction in Scheme 4.1 of efficiently decodable Γ-error decodable secret-sharing schemes shows that one-round (Γ, Σ^c)-PSMT with polynomial time message recovery (that is, polynomial in the number of channels) is possible provided condition $Q(\Gamma, \Gamma, \Sigma^c)$ holds and Σ can be realised by a secret-sharing scheme (M, φ) for which l is polynomial in n. We observe that the conditions placed on the level 2 schemes in Theorem 4.7 are equivalent to requiring them to be one-round (Γ, Σ^c)-PSMT schemes. For the purpose of constructing one-round (Γ, Σ^c)-PSMT schemes we can relax the conditions on the level 1 scheme to allow it to be a one-round (Γ, Σ^c)-PSMT scheme also (rather than requiring a Γ-error decodable secret-sharing scheme). However, it is not clear whether in general this will lead to schemes with smaller overhead: whereas the size of the level 1 shares

may be reduced, since the size of the minimal authorised sets in the level 1 scheme is potentially increased, the required number of level 2 schemes may increase.

It would be of interest to determine whether there exist general constructions for one-round (Γ, Σ^c)-PSMT schemes with polynomial time message recovery with communication overheads lower than those achieved by our modified version of Scheme 4.1, or whether efficient decoding techniques can be found for specific classes of adversary structure.

6 Conclusion

We have seen that the concept of error decodable secret sharing leads to useful insight into the construction of one-round PSMT schemes for general adversary structures, providing a link between one-round PSMT and the existing literature on secret-sharing. We have noted the need to consider the efficiency of message recovery when considering one-round PSMT with general adversaries, and we have shown that Kurosawa's scheme for constructing efficiently decodable secret-sharing schemes can be modified to give one-round (Γ, Σ^c)-PSMT schemes with a better trade-off between the efficiency of message recovery and the amount of information transmitted. We are left with the following open questions:

- Do there exist constructions of one-round (Γ, Σ^c)-PSMT schemes with polynomial time message recovery for general Γ, Σ with lower communication overheads than those obtained by the modifications to Scheme 4.1?
- Is it possible to determine in general which classes of Γ and Σ can be realised by schemes with efficient decoding/message recovery?
- Is it possible to find efficient decoding/message recovery techniques for specific classes of Γ and Σ?

References

[1] M. Bellare and P. Rogaway. Robust computational secret sharing and a unified account of classical secret-sharing goals. In P. Ning, S. D. C. di Vimercati, and P. F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 172–184. ACM, 2007.

[2] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, 24(3):384–386, 1978.

[3] G. R. Blakley. Safeguarding cryptographic keys. In *Proc. 1979 National Comp. Conf.*, volume 48, pages 313–317. AFIPS Press, 1979.

[4] C. Blundo, A. D. Santis, and U. Vaccaro. Efficient sharing of many secrets. In P. Enjalbert, A. Finkel, and K. W. Wagner, editors, *STACS '93*, volume 665 of *LNCS*, pages 692–703. Springer, 1993.

[5] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *FOCS '85*, pages 383–395. IEEE, 1985.

[6] R. Cramer, I. Damgard, and U. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In B. Preneel, editor, *Eurocrypt 2000*, volume 1807 of *LNCS*, pages 316–334. Springer, 2000.

[7] Y. Desmedt, Y. Wang, and M. Burmester. A complete characterization of tolerable adversary structures for secure point-to-point transmissions without feedback. In X. Deng and D.-Z. Du, editors, *ISAAC '05*, volume 3827 of *LNCS*, pages 277–287. Springer, 2005.

[8] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, 1993.

[9] S. Fehr and U. M. Maurer. Linear vss and distributed commitments based on secret sharing and pairwise checks. In M. Yung, editor, *CRYPTO '02*, volume 2442 of *LNCS*, pages 565–580. Springer, 2002.

[10] M. Fitzi, M. K. Franklin, J. A. Garay, and S. H. Vardhan. Towards optimal and efficient perfectly secure message transmission. In S. P. Vadhan, editor, *TCC 07*, volume 4392 of *LNCS*, pages 311–322. Springer, 2007.

[11] M. Hirt and U. Maurer. Player simulation and general adversary structures in perfect multi-party computation. *J. Cryptol.*, 13(1):31–60, 2000.

[12] W. C. Huffman and V. Pless. *Fundamentals of Error-correcting Codes*. Cambridge Univ. Press, 2003.

[13] M. Ito, A. Saito, and T. Nishizeki. Multiple assignment scheme for sharing secret. *J. Cryptol.*, 6(1):15–20, 1993.

[14] W.-A. Jackson and K. M. Martin. Cumulative arrays and geometric secret sharing schemes. In J. Seberry and Y. Zheng, editors, *AUSCRYPT '92*, volume 718 of *LNCS*, pages 48–55. Springer, 1992.

[15] W.-A. Jackson and K. M. Martin. A combinatorial interpretation of ramp schemes. *Australas. J. Combin.*, 14:51–60, 1996.

[16] K. Kurosawa. General error decodable secret sharing scheme and its application. Cryptology ePrint Archive, Report 2009/263, 2009. `http://eprint.iacr.org/`.

[17] K. Kurosawa, S. Obana, and W. Ogata. t-cheater identifiable (k, n) threshold secret sharing schemes. In D. Coppersmith, editor, *CRYPTO*, volume 963 of *LNCS*, pages 410–423. Springer, 1995.

[18] K. M. Martin. Challenging the adversary model in secret sharing schemes. In *Coding and Cryptography II*, Proceedings of the Royal Flemish Academy of Belgium for Science and the Arts, pages 45–63, 2008.

[19] U. M. Maurer. Secure multi-party computation made simple. *Discrete Appl. Math.*, 154(2):370–381, 2006.

[20] R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Commun. ACM*, 24(9):583–584, 1981.

[21] S. Obana and T. Araki. Almost optimum secret sharing schemes secure against cheating for arbitrary secret distribution. In X. Lai and K. Chen, editors, *ASIACRYPT*, volume 4284 of *LNCS*, pages 364–379. Springer, 2006.

[22] W. Ogata, K. Kurosawa, and D. Stinson. Optimum secret sharing scheme secure against cheating. *SIAM J. Discrete Math*, 20:79–95, 2006.

[23] R. S. Rees, D. R. Stinson, R. Wei, and G. H. J. van Rees. An application of covering designs: determining the maximum consistent set of shares in a threshold scheme. *Ars Comb.*, 53, 1999.

[24] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[25] D. R. Stinson and S. Zhang. Algorithms for detecting cheaters in threshold schemes. *J. Comb. Math. Comb. Comput.*, 61:169–191, 2007.

[26] R. Tso, Y. Miao, and E. Okamoto. A new algorithm for searching a consistent set of shares in a threshold scheme with cheaters. In J. I. Lim and D. H. Lee, editors, *ICISC '03*, volume 2971 of *LNCS*, pages 377–385. Springer, 2004.