Challenging the adversary model in secret sharing schemes

Keith M. Martin[†]

[†]*Information Security Group, Department of Mathematics*
Royal Holloway, University of London
Egham, Surrey TW20 0EX
United Kingdom
Email: keith.martin@rhul.ac.uk
WWW: www.isg.rhul.ac.uk/∼martin

Abstract

Secret sharing schemes are cryptographic primitives for distributing shares of a secret amongst a set of entities in such a way that only certain coalitions can reconstruct the secret from their shares. Secret sharing schemes are highly versatile primitives that are particularly useful in applications where there is no single point of trust. Traditionally, secret sharing schemes are studied in an environment where there is a trusted dealer who initiates the scheme, passive adversaries who do not manipulate shares, and participants who either co-operate or do not co-operate in a reconstruction attempt. These assumptions are reasonable in some situations, but do not necessarily map comfortably onto many application environments. In this paper we review work on secret sharing schemes where one or more of these assumptions is challenged.

1. Introduction

A *secret sharing scheme* is a method of distributing a *secret* amongst a set of *participants* by giving each participant a *share* in such a way that only certain specified subsets of participants (defined by the *access structure* Γ) can reconstruct the secret from a pooling of their shares. Secret sharing schemes are highly versatile cryptographic primitives and, as a result, have been employed in a vast range of different applications including protection of cryptographic keys, access control, key recovery mechanisms, electronic voting, distributed certificate authorities, online auctions and secure multiparty computation. They are also objects of inherent mathematical interest and have also been researched as such.

The access structure of a secret sharing scheme normally partitions the set of all subsets of participants into *authorised sets* who are able to recover the secret and *unauthorised sets* who can not. (Some schemes feature a third class of subsets who are neither authorised or unauthorised.) The two fundamental properties of a secret sharing scheme are thus:

1. *Privacy*: Unauthorised subsets of participants should be prevented from learning the secret.

2. *Recoverability*: Authorised subsets of participants should be able to recover the secret by pooling their shares.

Secret sharing schemes also involve two functionalities that are, in many cases, carried out by a dedicated entity. The *dealer* is normally responsible for generating system parameters, generating the secret, creating initial shares and sending initial shares to participants. The *combiner* is responsible for pooling shares and reconstructing the secret. The dealer is normally a fully trusted third party, while the combiner is often left unspecified (but can be a third party or even one of the participants).

Most secret sharing schemes feature a *monotone* access structure Γ, which has the property that if $A \in \Gamma$ then all supersets A' of A are also in Γ. Where the access structure consists of all subsets of n participants of at least size k, the secret sharing scheme is normally referred to as a (k, n)*-threshold scheme*.

This article is not intended to be a tutorial on secret sharing schemes (although we provide a very basic primer). We recommend established review articles such as [37] for a more detailed mathematical treatment of the basics. The purpose of this article is to present an overview of research in one area of secret sharing, namely secret sharing under different adversarial models. We will explain these different adversarial models and provide pointers to the extensive literature on this subject, so that interested researchers can pursue this topic in greater detail.

The organisation of the remainder of the article is as follows. In Section 2, we present the "traditional" model for secret sharing schemes. In Section 3, we note the limitations of the traditional adversarial model. In the remaining sections, we review research results for a number of different adversarial models and discuss the extent to which these, at least in part, overcome the limitations of the traditional case.

2. Traditional secret sharing

Traditionally, secret sharing schemes have been studied in an *information-theoretic* security model, where the security is independent of the computing capabilities of an adversary. This can however be relaxed and some schemes have been defined for *computationally secure* models where the scheme relies on the difficulty of a mathematical problem. The information-theoretic security model permits a notion of *perfect* privacy. In a *perfect* secret sharing scheme, unauthorised sets do not learn any information about the security via their shares.

The most famous perfect secret sharing scheme is the (k, n)-threshold scheme first proposed by Shamir in 1979 and hereafter referred to as a *Shamir* threshold scheme [35]. The idea behind this construction is simple and elegant. A Shamir (k, n)-threshold scheme is defined over $\mathbb{Z}_p$. Each participant P_i is associated with a unique non-zero x_i (which is not secret). If the secret is s, the dealer randomly chooses a polynomial $f(x)$ of degree at most $k-1$ defined over $\mathbb{Z}_p$ such that $f(0) = s$. The dealer then securely issues participant P_i with share $f(x_i)$. The Shamir scheme has perfect privacy since knowledge of $k-1$ shares does not leak any information

about the secret s. It also has recoverability since any k participants can interpolate their shares to recover the polynomial $f(x)$ and hence the secret s.

There are numerous equivalent ways of modelling a general information-theoretically secure secret sharing scheme:

- **Information theory**: By representing entities as probability distributions and making statements about conditional entropy [15].
- **Combinatorially**: By defining a matrix of possible distribution rules [37].
- **Algorithmically**: As two algorithms *Share* and *Reconstruct* and defining related properties.

The information theory model is useful because it can be effectively used to prove results about the amount of information that needs to be stored by participants in a secret sharing scheme. The most well-known result of this type is that in a perfect secret sharing scheme each participant needs to store at least the amount of information that it takes to represent the secret. This leads naturally to notions of secret sharing scheme efficiency. The *information rate* of a secret sharing scheme is the ratio of the "size" of the secret over the "size" of the largest share. Schemes where this rate is precisely one are optimal in this regard and referred to as *ideal*.

The combinatorial secret sharing model is useful because it can be used to classify certain types of secret sharing scheme in the rich language of combinatorial mathematics. For example, ideal threshold schemes are known to be equivalent to a number of well-studied combinatorial objects that include *orthogonal arrays* and *maximum distance separable codes* [23].

The most studied examples of perfect secret sharing schemes for general monotone access structures are *linear* secret sharing schemes. They are so-named because the secret can be computed as a linear combination of any set of authorised shares. Linear secret sharing schemes can be equivalently defined in terms of vector spaces [7], projective geometries [24], error correcting codes [42] or monotone span programs [25]. A Shamir threshold scheme is an example of a linear secret sharing scheme.

3. Changing the adversary model

The traditional secret sharing model makes the following important assumptions about the potentially malicious behaviour of entities involved in the scheme (this behaviour is typically modelled by the idea of an *adversary*):

- *Trusted dealer*: An adversary cannot corrupt the dealer, who is fully trusted.
- *Passive*: An adversary can capture shares, but otherwise the protocol is executed correctly and shares are not corrupted.
- *Polarised participants*: Participants are either completely honest (follow the protocol) or completely malicious (they have been captured by an adversary who will attempt to subvert the protocol).

	Section	Honest users learn secret?	Honest users alerted to cheating?	Adversary learns secret?
Robust schemes	Sec. 4	Yes	Sometimes	Yes
Cheater detection	Sec. 5.1	No	Yes	Yes
Cheater identification	Sec. 5.2	No	Yes	Yes
Almost robust (fairness) schemes	Sec. 6	Sometimes	Yes	Sometimes
Cheating immune	Sec. 7	No	No	No

Table 1: Properties of schemes with respect to Tompa and Woll undesirable consequences

These assumptions are reasonable in some situations, but do not necessarily map comfortably onto many application environments. In particular, they fall short of some of the high security demands placed on computationally secure cryptographic primitives that are modelled using provable security. A substantial amount of recent work on secret sharing schemes has been devoted to looking at the problem of secret sharing in situations where one or more of these assumptions is challenged. In particular, most of this work involves *active* adversaries, who are able to take full control of participants and corrupt their shares.

3.1. The Tompa and Woll attack

The first challenge to this traditional secret sharing adversary model was a paper by Tompa and Woll [41] which showed how an active adversary can exploit the Shamir threshold scheme (in fact, this type of attack can be extended to any linear secret sharing scheme). They assessed the impact of an active adversary who takes the form of a participant who maliciously submits a false share during a reconstruction attempt. In other words, some participant P_i submits a false share λ_i instead of a correct share $f(x_i)$. This attack has several undesirable consequences:

1. it prevents the honest participants from learning the correct secret;

2. it fails to alert the other participants that they have not reconstructed the correct secret;

3. it allows the adversary to learn the correct secret (by exploiting knowledge of $f(x_i) - \lambda_i$).

In the next sections, we review a number of different proposals for overcoming the consequences of the Tompa and Woll attack. Table 3.1 indicates which of these consequences are addressed by which types of scheme.

3.2. Issues arising from new adversarial models

Changing the adversarial model (in particular moving from passive to active adversaries) for secret sharing scheme raises a number of issues that are not apparent in the traditional model.

3.2.1. WHO RECONSTRUCTS SHARES?

The recoverability property of a traditional secret sharing scheme simply states that in the presence of a set of shares corresponding to an authorised set it will be possible to reconstruct the secret. It does not place demands on precisely *who* will perform this reconstruction. This is normally assumed either to be one of the participants themselves or a third party entity (which could correspond to the participants collectively reconstructing the secret themselves "in the open"). The difference is often abstracted away by referring to a *combiner*, who could be either of these.

If, however, we change the traditional adversarial model to consider adversaries who can corrupt shares, then the identity of the combiner becomes important. This is because if the combiner is a participant then (assuming a trusted dealer) they will always know at least one valid share, whereas if the combiner is a third party entity then they will not. In [3], these two cases are referred to as recovery by an *uncorrupted player* or by an *external party*.

3.2.2. ARE SHARES REVEALED DURING RECONSTRUCTION?

Likewise, in the traditional secret sharing model the issue of whether shares are revealed to the participants during a reconstruction attempt is not explicitly addressed. When considering more adventurous adversaries, it can be important to distinguish between *open reconstructions*, where shares are revealed, and *closed reconstructions*, where they are not. Note that a reconstruction can be open or closed, regardless of who is playing the role of the combiner.

3.2.3. ARE ADVERSARIES STATIC OR DYNAMIC?

The traditional secret sharing scheme model does not explain *how an adversary might behave* but only the consequences of the adversary gaining knowledge of sets of shares. If we move to more general adversarial settings, then it can become important to distinguish between the *static adversary* case where the adversary captures a particular fixed set of participants (shares), and the *dynamic adversary* case where the adversary can choose which participants to capture one by one, as it learns their shares.

3.2.4. WHAT ARE THE GOALS OF AN ADVERSARY?

With respect to privacy, in all secret sharing schemes an adversary wishes to learn information about the secret, perhaps through learning information about shares. However with respect to recoverability, the goals of different types of secret sharing schemes vary quite subtly. In the traditional model, the passive adversary can only engage in share capture and hence they can only try to prevent reconstruction of the secret through withholding shares. For each of the different types of secret sharing schemes, we will begin by identifying the main recoverability goals of an adversary.

4. Robust secret sharing schemes

Robust secret sharing is a term that is commonly used to describe schemes where, even if some participants submit incorrect shares, the correct secret can still be recovered.

4.1. Bellare and Rogaway's classification

Bellare and Rogaway [3] observed that the idea of robust secret sharing had been studied within a number of different models. They thus proposed a unifying framework for secret sharing schemes that maintain a trusted dealer and polarised participants, but for which the share capture only assumption can be relaxed.

This framework identifies three different meaningful levels of privacy (we omit their fourth category of *no privacy*):

- *Perfect* (*PSS*): no information is revealed about the secret, independent of the computing power of an adversary (the traditional notion of privacy);
- *Statistical* (*SSS*): a small amount of information is potentially revealed about the secret, independent of the computing power of an adversary (this corresponds to the traditional secret sharing scheme model when a scheme is not perfect);
- *Computational* (*CSS*): the secret is protected from an adversary with "reasonable" computing resources (computationally secure schemes).

The framework further identifies nine different levels of recoverability. These are identified by specifying:

- The extent to which an adversary can prevent an authorised set of honest participants from reconstructing the secret. This can either be:
  - *PR*: an authorised set of honest participants cannot be prevented from reconstructing the secret, independent of the computing power of an adversary (which is the case in the traditional notion of recoverability);
  - *SR*: an authorised set of honest participants can only be prevented from reconstructing the secret with a small probability, independent of the computing power of an adversary;
  - *CR*: an authorised set of honest participants can reconstruct the secret in the presence of an adversary with "reasonable" computing resources.
- The extent to which an adversary can corrupt shares. Adversaries can either be classified as:
  - *Erasure* (*0*): an adversary cannot corrupt shares (only view them and prevent them being used in a reconstruction attempt, which is the case in the traditional notion of recoverability);

– *Recoverability-1* (*1*): an adversary can corrupt all shares except one (this corresponds to the case where the combiner is an honest participant - see Section 3.2.1);
– *Recoverability* (*2*): an adversary can corrupt all shares (this corresponds to the case where the combiner is a third party - see Section 3.2.1).

Note that erasure and recoverability-1 adversaries are special cases of recoverability adversaries.

This framework accommodates 27 different cases, depending on the levels of privacy and recoverability, each of which defines a type of secret sharing scheme with a trusted dealer. These are labelled using the framework abbreviations. We have already mentioned three of these types;

- *PSS-PR0*: this corresponds to traditional perfect secret sharing schemes of Section 2;
- *SSS-PR0*: this corresponds to traditional non-perfect secret sharing schemes;
- *CSS-PR0*: this corresponds to the computationally secure secret sharing schemes.

4.2. Robust constructions

Recall that traditional ideal (k, n)-threshold schemes have a rich combinatorial structure and can be classified in a number of ways, including as maximum distance separable codes. Further, such schemes (but not all of them) can be linear, with Shamir's scheme as the most famous example of this.

Tompa and Woll's attack of Section 3.1 shows that in general traditional ideal (k, n)-threshold schemes are not secure in the presence of share-corrupting adversaries, however in certain cases limited degrees of robustness are possible:

- Tompa and Woll's attack is most devastating when the underlying threshold scheme is linear, since in this case the adversary who has corrupted up to $k - 1$ participants can not only prevent honest participants from obtaining the secret, but can also recover the correct secret, even during a closed reconstruction attempt. If the underlying scheme is non-linear, then the adversary can still disrupt honest recovery, however with respect to the recovering the secret:
  – during a closed reconstruction attempt, they may not be able to obtain the correct secret from this attack (since they will not know the honest shares);
  – during an open reconstruction attempt, they will be able to obtain the correct secret from this attack (since they will know a threshold of honest shares).
- It was first noted in [27] that in a linear ideal (k, n)-threshold scheme if $l > k$ participants, where at most $l - k$ are corrupt, attempt a recovery then the presence of corrupt shares will be detected. It follows immediately from the combinatorial classification of ideal schemes as MDS codes that this also holds for non-linear schemes (since the submitted shares will be "inconsistent"). However this does not deliver full robustness since it may not be possible to determine the correct secret.

- However it was also noted in [27] that in a linear ideal (k, n)-threshold scheme if $l > k$ participants, where at most $(l - k)/2$ are corrupt, attempt a recovery, then the corrupt shares can be identified and corrected, providing a type of robust PSS-PR2 scheme. It was observed in [32] that this also extends to non-linear schemes.

- Tompa and Woll [41] proposed a method of adapting traditional (k, n)-threshold schemes to make them robust. This required a massive increase in share size and multiple rounds (each of which involves a reconstruction attempt of a different (k, n)-threshold scheme). While fairly impractical, this scheme is an example of a robust PSS-SR1 secret sharing scheme. (As observed in [3] this is more than just SR1, since it is also simultaneously a PSS-PR0 scheme.)

5. Detecting and identifying cheaters

Recall the Tompa and Woll attack described in Section 3.1 and its three undesirable outcomes. Robust secret sharing schemes overcome this attack by eliminating the first of these outcomes and allowing successful recovery of the secret in the presence of an adversary who can corrupt shares. Solutions of this type are fairly heavy-handed and so it is worth considering weaker approaches which might be just as effective in certain application environments. We discuss two such approaches here:

1. *Secret sharing schemes with cheater detection* allow honest participants to detect any corrupt shares that have been submitted by an adversary.

2. *Secret sharing schemes with cheater identification* allow honest participants to detect and identify any corrupt shares that have been submitted by an adversary.

Secret sharing schemes with cheater identification (detection):

- Assume a trusted dealer.

- Honest participants are willing to sacrifice recovery of the secret if an adversary corrupts shares, so long as corrupt shares are identified (detected).

- The main recoverability goal of the adversary is to prevent the correct secret from being reconstructed while remaining unidentified (undetected).

- Potentially allow the adversary to obtain the correct secret while the honest participants do not.

It is widely noted that addressing these problems in computationally-secure environments could be handled by using mechanisms such as digital signatures. Thus the bulk of research on these types of secret sharing schemes has concentrated on the information-theoretic environment. This capability, not surprisingly, comes at a cost. This cost is typically that the schemes:

- *Have large shares*: Each participant is equipped with extra information that allows him to recognise malicious behaviour.

- *Require extra cooperation*: Need more than a minimum coalition of participants to cooperate in a recovery attempt.

5.1. Secret sharing schemes with cheater detection

The weakest approach is just to require detection of malicious behaviour. Indeed Tompa and Woll [41] proposed a (fairly expensive) fix for their attack (see Section 3.1) that does precisely this. We also observed in Section 4.2 that an ideal (k, n)-threshold scheme can detect t cheating participants if $k + t$ participants (at most t of whom are cheating) collaborate. As noted for cheater correction, this is more cooperation than the scheme was originally designed to support.

In [10], it was shown that in order to restrict the probability of an adversary who has corrupted $k - 1$ shares (and wants to deceive an honest participant) from escaping detection to ϵ, in a perfect (k, n)-threshold scheme, it is necessary to increase the lower bound on each participant's share size $|\mathcal{S}_i|$ from $|\mathcal{S}|$ (in traditional perfect secret sharing schemes) to $\frac{|\mathcal{S}|}{\epsilon}$. This bound was derived under the (perhaps unlikely) assumption that the adversary somehow knows the correct secret before they commence their attempt to deceive the honest participant. We will describe schemes that provide cheater detection under this assumption as *informed* and the more likely situation that cheating participants do not know the secret as *uninformed*. Note that such schemes have been rather ambiguously defined as *robust* and *secure* in some of the previous literature such as [9], and cryptically as being based on the *CDV assumption* and the *OKS assumption* in the likes of [29].

In [30], the bound for informed schemes was improved to:

$$|\mathcal{S}_i| \geq \frac{|\mathcal{S}| - 1}{\epsilon^2} + 1,$$

and a bound for uninformed schemes was given as:

$$|\mathcal{S}_i| \geq \frac{|\mathcal{S}| - 1}{\epsilon} + 1.$$

A family of uninformed (k, n)-threshold schemes with cheater detection probability $\epsilon = 1/|\mathcal{S}|$ was then constructed using combinatorial objects called *difference sets* that meet the uninformed bound. Both these bounds easily generalise to perfect secret sharing schemes that are not threshold schemes, in which case rather than being concerned with up to $k - 1$ adversaries, we are concerned with adversaries of the form $A \setminus \{P\}$ (where A is a minimal authorised set and $P \in A$) trying to cheat participant P.

Two generic techniques were given in [9] that allow a linear secret sharing scheme for any access structure to be converted into a secret sharing scheme that can detect cheating participants.

- To construct an informed scheme, each participant is given three separate shares:

  1. a share of the real secret k;
  2. a share of a random value r;
  3. a share of the product kr.

During a recovery attempt, all three values are recovered and it is checked whether the third secret is the product of the first two.

- To construct an uninformed scheme, each participant is given two separate shares:

  1. a share of the real secret k;
  2. a share of the square k^2 of the real secret.

  During a recovery attempt, both values are recovered and it is checked whether the second is the square of the first.

As for the scheme of [30], in each of these cases the chances of cheating participants getting past the check is $1/q$, where $q = |\mathcal{S}|$. The equivalent share size bounds from [30] for these cases are $q^3 - q^2 + 1$ and $q^2 - q + 1$. Thus with share sizes of q^3 and q^2, the generic schemes of [9] are close to optimal.

An alternative technique for building informed schemes that are secure against cheating for any access structure was proposed in [29] (and later corrected in [2]) which involves using a linear secret sharing scheme to issue each participant with a share of the genuine secret and a share of the key of a class of universal hash functions. Both the secret and the key are reconstructed and the latter is used to check the validity of the former. Two schemes are proposed. The first scheme has the same parameters to the informed scheme in [9]. The second scheme has the attractive property that it is configurable in the sense that the share size can be traded off against the cheating probability (unlike the previous schemes which fix this at $\epsilon = 1/|\mathcal{S}|$). More precisely, for $|\mathcal{S}| = p^N$, the second scheme has $\epsilon = (N+1)/p$ and $|\mathcal{S}_i| = p^{N+2}$. While this scheme does not meet the bound of [30], such schemes perform considerably better than any previous schemes for cases where $\epsilon > 1/|\mathcal{S}|$.

Note that the schemes of [9] and [29] have an advantage over the scheme of [30] in that they work for arbitrary probability distributions on the secret (whereas the scheme of [30] requires this distribution to be uniform).

All the schemes discussed thus far in this section assume that the adversary controls at most $k-1$ shares of a (k, n)-threshold scheme. An adversary in control of at least k shares is powerful enough to obtain the secret on its own, however it is argued in [1] that they might still wish to fool another shareholder into believing that the secret is a different value. An informed scheme (and any scheme under these conditions is informed by default) is proposed in [1] that involves giving each participant a share in two Shamir (k, n)-threshold schemes plus a random number, where the random numbers serves as a "secret identity" and is used as the participant's x_i value in the second Shamir scheme.

5.2. Secret sharing schemes with cheater identification

A stronger alternative to countering the effectiveness of malicious participants is to build in a mechanism that, with high probability, identifies any participants who submit incorrect shares during a secret reconstruction attempt.

As observed in Section 4.2, an ideal (k, n)-threshold scheme can identify t cheating participants, but only if $k + 2t$ participants (at most t of whom are cheating) collaborate. This is more cooperation than the scheme was originally designed to support.

All the bounds proven for share size of schemes that provide cheater detection clearly also apply to those with cheater identification. However the known constructions for schemes with cheater identification capability have significantly larger shares than these bounds. In the schemes of [33] and [8], shares are of size $|\mathcal{S}|^{f(n)}$ for a linear function of the number of participants n. In [26], a perfect (k, n)-threshold scheme was exhibited that for $k \geq 3t + 1$ allows k participants to identify up to t cheaters and has the property that:

$$|\mathcal{S}_i| \geq \frac{|\mathcal{S}|}{\epsilon^{t+2}},$$

where ϵ is the probability that up to t cheaters fail to be detected.

6. Almost robust secret sharing

It is noted in [22] that a (k, n)-threshold scheme that can identify $r < k/2$ cheaters can be used to create an "almost robust" (k, n)-threshold scheme that allows the honest participants to obtain the secret under certain circumstances. A generalised version of this conversion works by giving each participant one share in a (k, n)-threshold scheme that can identify r cheaters with secret k_1, and one share in a $(k - r, n)$-threshold scheme that can identify r cheaters with secret k_2. The real secret $s = k_1 \oplus k_2$. During a reconstruction attempt:

1. Participants submit their first shares and they are checked for the presence of cheaters (technically it suffices that a scheme with cheater *detection* is used for this stage). If cheaters are noted, then recovery is aborted.

2. If no cheaters are noted in the first stage, participants submit their second shares and they are checked for the presence of cheaters. Even if r cheaters are identified, the $k-r$ honest participants can still recover k_2.

3. The secret s is computed from k_1 and k_2.

This scheme, which was termed a *fairness* secret sharing scheme in [22], is "almost" a PSS-SR1 robust scheme. The reason that it is "almost" is that if the adversary decides to cheat during the first stage, then recovery is aborted and so nobody succeeds in reconstructing the secret.

7. Cheating immune secret sharing

Cheating immune secret sharing schemes are secret sharing schemes which adopt a subtly different approach to addressing the problem of adversaries who can corrupt shares. The idea behind cheating immune secret sharing is to remove the benefit to an adversary of submitting

corrupted shares, thus reducing the incentive of an adversary to attack the scheme solely to one of disruption without personal gain. Cheating immune secret sharing schemes:

- Assume a trusted dealer.
- Assume a third party (external) combiner.
- Honest participants are willing to sacrifice recovery of the secret if an adversary corrupts shares, so long as the adversary does not as a result have an advantage over the honest participants with respect to recovery of the genuine secret.
- The main recoverability goal of the adversary is to have more knowledge about the secret than a set of honest participants.
- If an adversary submits corrupted shares, then nobody obtains the secret.

Cheating immune secret sharing schemes were first proposed in [43]. An information-theoretic setting was adopted and two notions were proposed: t-*cheating immune* if an adversary who submits t incorrect shares gains no advantage and a more general *strictly* t-*cheating immune* if an adversary who submits *up to* t incorrect shares gains no advantage. They were subsequently investigated in greater depth in [13], where some combinatorial properties of cheating immune schemes were proven, it was shown that cheating immune secret sharing schemes cannot be perfect, it was proven that t-cheating immune (n, n)-threshold scheme must have $t < n/2$ and some constructions were given. By classifying cheating immune secret sharing schemes in terms of resilient functions, this bound is slightly improved in [6].

However the current research on cheating immune secret sharing schemes is very limited in scope since all schemes investigated thus far;

1. are (n, n)-threshold schemes;
2. have the secret and shares chosen from the same set (in this sense they are ideal but not perfect).

Using the notation of Section 4.1, the cheating immune secret sharing schemes studied to date would be classified as having SSS privacy and operate under recoverability (2) adversaries. However, they are not robust schemes since an adversary can prevent reconstruction of the secret.

8. Rational secret sharing

Rational secret sharing models a scenario which involves:

- a trusted dealer;
- open reconstruction;
- participants are neither completely honest nor completely malicious.

With respect to the last property, the participants are described as *rational* because they generally want to recover the secret (this is their top priority) but will take the opportunity to cheat if it is in their interest (in particular they would prefer as few people to know the secret as possible). It is observed that in a traditional (t, n)-threshold scheme a good "rational" strategy for a participant is to wait for $t - 1$ other participants to reveal their share. The participant then withholds its share and learns the secret, while the $t - 1$ who revealed their shares remain one share short. The rational secret sharing scheme proposed in [19] involves a number of rounds controlled by a dealer. In each round the dealer either:

1. with probability β generates shares of a (t, n)-threshold scheme protecting the genuine secret $s \in \mathcal{S}$,
2. with probability $1 - \beta$ generates shares of a (t, n)-threshold scheme protecting a random secret $s' \in \mathcal{S}' \setminus \mathcal{S}$ for some larger set $\mathcal{S}'$ that contains the set of genuine secrets $\mathcal{S}$.

After this has happened, participants who wish to take part broadcast their shares. It is shown in [19] that by choosing the correct parameters, this situation can be modelled by a game in which the "rational" strategy is to take part honestly. The game stops when in some round in which the dealer chose the first option, more than t participants broadcast their shares.

9. Verifiable secret sharing

The adversaries that we have modelled thus far have not been able to corrupt the dealer of the secret sharing scheme. A *verifiable secret sharing scheme* (or *VSS*) is designed to tolerate an adversary who can corrupt the dealer and some of the participants. Verifiable secret sharing schemes:

- Do not assume a trusted dealer.
- Honest participants want to recover the secret even if an adversary corrupts the dealer and some shares.
- The main recoverability goal of the adversary is to prevent the correct secret from being reconstructed.

A VSS thus requires an additional algorithm called *Verify* to be run which allows participants to verify the validity of their shares (before making any reconstruction attempt). At the end of this algorithm, each participant output either decides to *accept* or *reject* its share. The algorithm must check for:

- *Consistency*: any authorised group of participants $A \in \Gamma$ that all *accept* their shares will be able to reconstruct the same secret value u.
- *Correctness*: if the dealer was honest, then the above value u is the genuine secret.

A VSS is said to be:

1. *interactive*, if *Verify* involves participants exchanging messages between themselves.
2. *non-interactive*, if *Verify* only involves participants exchanging messages with the dealer.

9.1. Information-theoretically (interactive) secure VSSs

Given that the dealer has potentially been corrupted, it is impossible to establish an information-theoretically secure VSS without interactivity between honest participants. Hence, such schemes are proposed for interactive models, where not only there is a secure (private) channel between the (potentially corrupt) dealer and the participants, but also each pair of participants can communicate over their own secure channel.

Within this model, it is well-known that a VSS can only be established if the access structure Γ has the property that *no three subsets not in* Γ *span the entire participant set* (this is often referred to by saying that the *adversary structure*, which in this case is the compliment of the access structure, has the $Q^{(3)}$ *property*). This result was first shown for (k, n)-threshold access structures in [4], where it equates to requiring that the number of adversaries is less than $n/3$, and for general access structures in [21].

An information-theoretic verifiable (k, n)-threshold scheme based on polynomials that is almost perfect was first given in [4]. In [38], a perfect verifiable (k, n)-threshold scheme was proposed that supports $b < n/4 - 1$ adversaries. This scheme is based on *symmetric bivariate* polynomials.

In [12], a general construction for converting any linear secret sharing scheme with a $Q^{(3)}$ access structure Γ into an information theoretically secure VSS for Γ was demonstrated. This construction generalises the threshold construction of [38].

An interesting observation is made in [14] concerning the general construction in [12]. Given that the dealer is potentially corrupt, any system parameters produced by the dealer need to be checked as genuine. The construction in [12] relies on the publicly-known generator matrix G being "genuine". The VSS "proves" that the shares issues are consistent with respect to G, but does G actually realise the stated access structure Γ? In [14], it is shown that this is indeed a very hard problem. Note that this problem does not arise for the special case of (k, n)-threshold schemes since the generator matrix G in this case is easily "recognisable".

In [28], a relationship between information-theoretic secure VSSs and a class of error-correcting codes known as *error-set correcting codes* is observed. Finally, some work can been undertaken into designing interactive verifiable schemes with minimal round complexity [17, 18].

9.2. Computationally secure VSSs

To have a fully information-theoretically secure VSS places quite severe constraints on the resulting scheme. In particular:

- The scheme must necessarily be interactive.
- The number of tolerated adversaries is restricted.
- The schemes are relatively inefficient.

These problems can be overcome if we relax the security model to computational security. Note that there are two options for relaxing this security model:

1. The security of the underlying secret sharing scheme could be relaxed.

2. The security of the verifiability of the shares could be relaxed.

It is possible to relax the security model without relaxing *both* of these. This was demonstrated in two early VSS threshold schemes.

In the scheme of [16], the security of the underlying threshold scheme is relaxed but the verifiability is information-theoretically secure, whereas the underlying threshold scheme of [31] is information-theoretically secure (it is essentially the Shamir scheme), but the verifiability process is computationally secure. Both these schemes require the existence of secure (private) channels between the dealer and the participants. In [20], it is observed that the scheme of [31] generalises naturally to any linear secret sharing scheme for an arbitrary access structure.

Note that there have been a number of computationally secure interactive VSS schemes proposed in the literature [11], [39]. Such schemes offer potential efficiency advantages over information-theoretically secure schemes, but the requirement for interactivity would appear to limit their usefulness in comparison to non-interactive schemes.

9.3. Publicly-verifiable VSSs

Note that one of the by-products of allowing interactivity in VSS schemes such as those in Section 9.1 is that at the end of the *Verify* process, a group of honest participants is not only assured of the validity of their own shares, but also those of the other honest participants. This property is lost when we move on to a non-interactive model, such as those in Section 9.2.

For this reason, *publicly verifiable* secret sharing schemes (*PVSS* schemes) were proposed in [36]. These schemes essentially replace the algorithm *Verify* with one called *Publicly-Verify*, which works by publishing asymmetrically encrypted shares and allowing the consistency check to be performed on these encrypted shares. Note that:

1. The shares are distributed to participants using asymmetric channels and so schemes are only as secure as the underlying asymmetric cryptosystem.

2. The consistency of the shares is literally *publicly* verifiable, since the consistency check can be done by entities not holding a share themselves.

Note that PVSS schemes are by definition non-interactive as previously defined. However they are often referred to as being *interactive* if *Publicly-Verify* requires interaction between participants and the dealer, and *non-interactive* if this is not required.

In fact, the very first VSS scheme proposed in [11] is actually a PVSS scheme. In [36], two PVSS schemes are proposed that are based on the ElGamal cryptosystem, but are based on unconventional security assumptions.

The PVSS scheme proposed in [34] is more efficient and has security based on the standard Diffie-Hellman assumption and its decisional variant. This scheme, which can be built onto any linear secret sharing scheme, is based on using zero-knowledge proofs of correctness of shares. An alterative PVSS scheme is proposed in [40] which is based on the VSS scheme of [31] and thus, in contrast to [34] has an underlying information-theoretically secure secret sharing scheme. Two PVSS schemes are proposed in [5], one for sharing a discrete logarithm and the other for sharing a factorisation.

10. Conclusions

We have reviewed a range of flavours of secret sharing schemes, each of which is designed to combat a type of adversary who can do more "damage" than the adversary in the traditional secret sharing model. In each case, the additional capability to withstand more active attacks on the scheme comes at a price, either in terms of information storage, compromise in security model, or additional computational or communication requirements. Several of these schemes provide elegant applications of mathematical techniques. While we have sacrificed mathematical detail in this article, we have provided extensive pointers to the wider literature and attempted to set this very varied research area into a structured context that should aid entry into the field for new researchers.

There remain several gaps in the knowledge of secret sharing schemes under different adversarial models, particularly with regard to efficiency and optimisation of schemes under different adversarial assumptions. We can also expect further developments in the formalisation of models for such schemes, as only robust secret sharing schemes have been set in a framework compatible with much of the recent theoretical formalisation of other types of cryptographic primitive. There would thus seem to remain some room in this area for further application of interesting mathematical techniques to provide secret sharing schemes with the capability of coping with sophisticated adversarial behaviour.

References

[1] T. Araki. Efficient (k, n) threshold secret sharing schemes secure against cheating from $n-1$ cheaters. In *ACISP 2007*, volume 4586 of *Lecture Notes in Computer Science*, pages 133–142. Springer-Verlag, 2007.

[2] T. Araki and S. Obana. Flaws in some secret sharing schemes against cheating. In *ACISP 2007*, volume 4586 of *Lecture Notes in Computer Science*, pages 122–132. Springer-Verlag, 2007.

[3] M. Bellare and P. Rogaway. Robust computational secret sharing and a unified account of classical secret-sharing goals. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (ACM-CCS)*, 2007.

[4] M. Ben-Or, S. Golwasser, and A. Widgerson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proceedings of ACM STOC '88*, pages 1–10, 1988.

[5] F. Boudot and J. Traore. Efficient publicly verifiable secret sharing schemes with fast or delayed recovery. In *ICICS '99*, volume 1726 of *Lecture Notes in Computer Science*, pages 87–102. Springer-Verlag, 1999.

[6] A. Braeken, V. Nikov, and S. Nikova. On cheating immune secret sharing. In *Proceedings of 25th Symposium on Information Theory in the Benelux*, pages 113–120, 2004.

[7] E.F. Brickell. Some ideal secret sharing schemes. *J. Combin. Math. Combin. Comput.*, 9:105–113, 1989.

[8] E.F. Brickell and D.R. Stinson. The detection of cheaters in threshold schemes. *SIAM Journal on Discrete Mathematics*, 4(4):502–510, 1991.

[9] S. Cabello, C. Padró, and G. Sáez. Secret sharing schemes with detection of cheaters for general access structure. *Designs, Codes and Crytography*, 25:175–188, 2002.

[10] M. Carpentieri, A. De Santis, and U. Vaccaro. Size of shares and probability of cheating in threshold schemes. In *Eurocrypt '93*, volume 765 of *Lecture Notes in Computer Science*, pages 118–125. Springer-Verlag, 1994.

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

[12] R. Cramer, I. Damgard, and U.M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In *Eurocrypt 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer-Verlag, 2000.

[13] P. D'Arco, W. Kishimoto, and D.R. Stinson. Properties and constraints of cheating-immune secret sharing schemes. *Discrete Applied Mathematics*, 154:219–233, 2006.

[14] Y. Desmedt, K. Kurosawa, and T. Van Le. Error correcting and complexity aspects of linear secret sharing schemes. In *ISC 2003*, volume 2851 of *Lecture Notes in Computer Science*, pages 396–407. Springer-Verlag, 2003.

[15] J. Greene, E. Karnin, and M. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, IT-29:35–41, 1983.

[16] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the 28th IEEE Symposium on the Foundations of Computer Science*, pages 427–437, 1987.

[17] M. Fitzi, J. Garay, S. Gollakota, C. Pandu Rangan, and K. Srinathan. Round-optimal and efficient verifiable secret sharing. In *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 329–342. Springer-Verlag, 2006.

[18] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. The round complexity of verifiable secret sharing and secure multicast. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC '01)*, pages 580–589, 2001.

[19] S. Dov Gordon and J. Katz. Rational secret sharing, revisited. In *Security and Cryptography for Networks 2006*, volume 4116 of *Lecture Notes in Computer Science*, pages 229–241. Springer-Verlag, 2006.

[20] J. Herranz and G. Saez. Verifiable secret sharing for general access structures, with application to fully distributed proxy signatures. In *Financial Cryptography 2003*, volume 2742 of *Lecture Notes in Computer Science*, pages 286–302. Springer-Verlag, 2003.

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

[22] R.J. Hwang and C.-C. Chang. Enhancing the efficiency of (v, r, n)-fairness secret sharing schemes. In *Proceedings of 18th International Conference on Advanced Networking and Applications, AINA 2004*, pages 208–211, 2004.

[23] W.-A. Jackson and K.M. Martin. Combinatorial models for perfect secret sharing schemes. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 28:249–265, 1998.

[24] W.-A. Jackson, K.M. Martin, and C.M. O'Keefe. Geometrical contributions to secret sharing theory. *Journal of Geometry*, 79:102–133, 2004.

[25] M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the 8th Annual Structure in Complexity Theory*, pages 102–111, 1993.

[26] Kaoru Kurosawa, Satoshi Obana, and Wakaha Ogata. t-cheater identifiable (k, n) threshold secret sharing schemes. *Lecture Notes in Computer Science*, 963:410–423, 1995.

[27] R.J. McEliece and D.V. Sarwate. On sharing secrets and reed-solomon codes. *Communications of the ACM*, 24:583–584, 1981.

[28] V. Nikov and S. Nikova. On a relation between verifiable secret sharing and a class of error-correcting codes. In *Proceedings of WCC 2005*, Electronic Notes in Discrete Mathematics, pages 372–382. Elsevier, 2005.

[29] S. Obana and T. Araki. Almost optimum secret sharing shemes secure against cheating for arbitrary secret distribution. In *Asiacrypt '06*, volume 4284 of *Lecture Notes in Computer Science*, pages 364–379. Springer-Verlag, 2006.

[30] W. Ogata, K. Kurosawa, and D.R. Stinson. Optimum secret sharing schemes secure against cheating. *SIAM Journal on Discrete Mathematics*, 20:79–95, 2006.

[31] T.P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Crypto '91*, volume 547 of *Lecture Notes in Computer Science*, pages 129–140. Springer-Verlag, 1992.

[32] J. Pieprzyk and X.-M. Zhang. Ideal threshold schemes and mds codes. In *ICISC 2002*, volume 2587 of *Lecture Notes in Computer Science*, pages 253–263. Springer-Verlag, 2003.

[33] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of 21st ACM Symposium on Theory of Computing*, pages 73–85, 1989.

[34] B. Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In *Crypto '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 148–164. Springer-Verlag, 1999.

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

[36] M. Stadler. Publicly verifiable secret sharing. In *Eurocrypt '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 190–199. Springer-Verlag, 1996.

[37] D.R. Stinson. An explication of secret sharing schemes. *Designs, Codes and Cryptography*, 2:357–390, 1992.

[38] D.R. Stinson and R. Wei. Unconditionally secure proactive secret sharing schemes with combinatorial structures. In *SAC '99*, volume 1758 of *Lecture Notes in Computer Science*, pages 200–214. Springer-Verlag, 2000.

[39] C. Tang, Z. Liu, and M. Wang. A verifiable secret sharing scheme with statistical zero-knowledge. Cryptology ePrint Archive, Report 2003/222, 2003. http://eprint.iacr.org/.

[40] C. Tang, D. Pei, Z. Liu, and Y. He. Non-interactive and information-theoretic secure publicly verifiable secret sharing. Cryptology ePrint Archive, Report 2004/201, 2004. http://eprint.iacr.org/.

[41] M. Tompa and H. Woll. How to share a secret with cheaters. *Journal of Cryptology*, 1:133–138, 1988.

[42] M. van Dijk. A linear construction for perfect secret sharing schemes. In *Adv. in Cryptology - EUROCRYPT '94*, volume 950 of *Lecture Notes in Computer Science*, pages 23–34. Springer-Verlag, 1995.

[43] X.M. Zhang and J. Pieprzyk. Cheating immune secret sharing. In *ICICS '01*, volume 2229 of *Lecture Notes in Computer Science*, pages 144–149. Springer-Verlag, 2001.