Cryptogr. Commun. (2018) 10:79–99
DOI 10.1007/s12095-017-0215-z

Efficient robust secret sharing from expander graphs

Brett Hemenway[1] · **Rafail Ostrovsky**[2]

Received: 29 November 2016 / Accepted: 6 February 2017 / Published online: 7 March 2017

Abstract Threshold secret sharing allows a dealer to share a secret among n players so that any coalition of t players learns nothing about the secret, but any $t+1$ players can reconstruct the secret in its entirety. Robust secret sharing (RSS) provides the additional guarantee that even if t malicious players mangle their shares, they cannot cause the honest players to reconstruct an incorrect secret. In this work, we construct a simple RSS protocol for $t = \left(\frac{1}{2} - \epsilon\right) n$ that achieves logarithmic overhead in terms of share size and simultaneously allows efficient reconstruction. Our shares size increases by an additive term of $\mathcal{O}(\kappa + \log n)$, and reconstruction succeeds except with probability at most $2^{-\kappa}$. Previous efficient RSS protocols like that of Rabin and Ben-Or (STOC '89) and Cevallos et al. (Eurocrypt '12) use MACs to allow each player to check the shares of each other player in the protocol. These checks provide robustness, but require significant overhead in share size. Our construction identifies the n players as nodes in an expander graph, each player only checks its neighbors in the expander graph.

Keywords Robust secret sharing · Expander graphs · Secure message transmission

Mathematics Subject Classification (2010) 94A60 · 11T71 · 94C15

This article is part of the Topical Collection on Recent Trends in Cryptography.

✉ Brett Hemenway
fbrett@cis.upenn.edu

Rafail Ostrovsky
rafail@cs.ucla.edu

[1] University of Pennsylvania, Philadelphia, PA 19104, USA

[2] UCLA, Los Angeles, CA 90095, USA

1 Introduction

Robust secret sharing (RSS) is a protocol that allows a dealer to distribute a secret among n players, so that any coalition of t malicious parties learns nothing about the secret, while the honest parties can reconstruct the original secret *even if the malicious parties tamper with their shares*. An RSS protocol is δ-robust if reconstruction succeeds with probability at least $1 - \delta$. In this work, we design a δ-robust secret sharing protocol with overhead of size $\mathcal{O}\left(\log \frac{1}{\delta}\right)$ and efficient reconstruction.

Robust secret sharing can be viewed as a stand-alone protocol, providing a mechanism for secure and tamper-resistant outsourced storage, or as a means of sending messages across corrupted channels. In the setting of secure message transmission (SMT), a sender and receiver are connected via n independent channels, some fraction of which may be adversarially controlled [27]. Any RSS scheme can be made into an SMT protocol, by simply sending each share across a different channel [43]. Note that unlike Verifiable Secret Sharing (VSS) [51], in RSS the dealer is always honest. Unlike the model of Tompa and Woll [56], RSS assumes that the reconstruction algorithm receives shares from *all* players (not just a subset).

Standard secret sharing [6, 53] is extremely well understood, and there exist a wealth of protocols obtaining essentially optimal parameters in variety of different circumstances (see Section 4.2 for further discussion). Much less is known about robust secret sharing protocols, and designing RSS schemes is currently an active research area.

When the corruption threshold is low $t < \frac{n}{3}$, then simple Shamir sharing is already robust (because a Reed Solomon code of rate $\frac{1}{3}$ has relative distance $\frac{2}{3}$ and hence can recover from a $\frac{1}{3}$ fraction of errors). When $t > \frac{n}{2}$ it is not hard to see that RSS is impossible [38]. The interesting range of parameters is when $\frac{n}{3} < t < \frac{n}{2}$, in this regime, perfect RSS is impossible, but RSS is feasible if a negligible failure probability is allowed. Throughout this work, we will assume that κ is a security parameter, and reconstruction should succeed with probability at least $1 - 2^{-\kappa}$.

Early constructions of RSS protocols for $\frac{n}{3} < t < \frac{n}{2}$ fell into two categories, those with compact share size and inefficient reconstruction procedure [9, 19, 20, 39] and those with moderate share size and efficient reconstruction [14, 51].[1]

When the reconstruction procedure is allowed running time that is exponential in n, then robust secret sharing schemes have essentially no overhead in share size. On the other hand, when we require an efficient reconstruction procedure, the problem becomes much more difficult. The original scheme of Rabin and Ben-Or, had overhead[2] of $\mathcal{O}(\kappa \cdot n)$. The best existing scheme – that of Cevallos et al. – has overhead of $\tilde{\mathcal{O}}(\kappa + n)$ [14]. It is left as an open question in [14] whether the overhead can be reduced to $\mathcal{O}(\kappa)$. We exhibit a scheme

[1] The work of Lewko and Pastro [44] can be seen as interpolating between these two models.

[2] If the message space is $\mathcal{M}$, then any secret sharing scheme must have shares of size at least $\log |\mathcal{M}|$ to obtain privacy. When $\mathcal{M}$ is larger than n, then Shamir sharing achieves this bound. Since many RSS protocols (including ours) use Shamir sharing plus additional check information, we us the term "overhead" to denote the size of the check information. Thus the overhead of an RSS scheme is the share size (in bits) minus $\log |\mathcal{M}|$.

with overhead $\mathcal{O}(\kappa + \log n)$ whenever $t < \left(\frac{1}{2} - \epsilon\right) n$. Our scheme is conceptually simple and has efficient reconstruction procedure.

Our primary result is the following:

Theorem 1 *For any message space $\mathcal{M}$, and any $\epsilon, \delta > 0$, there exists an RSS protocol tolerating an $\frac{1}{2} - \epsilon$ fraction of malicious parties, with the probability of reconstruction failure bounded by δ, having shares size*

$$\log |\mathcal{M}| + \frac{4}{\epsilon^3} \log \frac{3n \log |\mathcal{M}|}{\epsilon^3 \delta}$$

This result is proven as Corollary 1.

For a t-private, $t+1$-threshold secret sharing scheme, even without robustness, the shares must be of size $\log |\mathcal{M}|$, and Carpentieri et al. [11] showed that to obtain a t-robust, $t+1$-threshold RSS with failure probability of $2^{-\kappa}$ shares of size $\log |\mathcal{M}| + \kappa$ is necessary. Our scheme, which has shares of size $\log |\mathcal{M}| + \mathcal{O}(\kappa + \log n + \log \log |\mathcal{M}|)$, comes close to achieving this overhead, although we work in a slightly weaker model where we consider a $(\frac{1}{2} - \epsilon)n$-robust, and $(\frac{1}{2} + \epsilon)n$-threshold RSS. In this model, the concurrent, independent work of Cramer et al. [18] shows how to achieve shares of size $\log |\mathcal{M}|/n$ using completely different techniques. Cheraghchi [17] also considers this model, and uses folded Reed-Solomon codes to attain RSS protocols with overhead $\mathcal{O}(\kappa)$.

2 Previous work

In this section, we use $m = \log |\mathcal{M}|$ to denote the length of the message, and n to denote the number of players.

Rabin and Ben-Or [51] created a robust secret-sharing scheme, by taking a threshold secret sharing scheme and adding MACs which allows players to authenticate each other's shares. The Rabin Ben-Or scheme first shares a secret s as $(s_1, \ldots, s_n)$, then generates n^2 MAC keys k_{ij} and tags $\tau_{ij} = \mathsf{MAC}(\mathsf{k}_{ij}, s_j)$. Player i is then given $(s_i, \{\tau_{ji}\}_j, \{\mathsf{k}_{ij}\}_j)$. This results in an RSS protocol, but the drawback is the shares are now of size $m + 2n\kappa$.

The [51] protocol was improved in [14], who showed that by using the same sharing scheme, but an improved reconstruction procedure increased the probability of identifying cheating players, and allowed them to use smaller MAC keys and tags. Thus reducing the share sizes from $\mathcal{O}(\kappa n \log n)$ to $\mathcal{O}(\kappa + n \log n)$.

The work of Cabello, Padró and Sáez [9, 10] takes any secret sharing scheme over a finite field $\mathbb{F}$, and makes it robust by taking the secret s, generating a random $r \in \mathbb{F}$, and sharing the triple $(s, r, r \cdot s) \in \mathbb{F}^3$. The probability an adversary can then generate shares that correspond to a valid secret is $1/|\mathbb{F}|$. This triples the size of the shares, but recovery time is exponential, because recovery requires iterating over all possible sets. Since there are at most $\binom{n}{t+1} \leq 2^n$ subsets of size $t+1$, the error probability of the protocol is $\delta \leq \frac{2^n}{|\mathbb{F}|}$. Thus the share size is $3 \max\left(m, \log \frac{2^n}{\delta}\right)$.

Cramer, Damgård and Fehr showed that the approach of [9, 10] results in essentially optimal share size. Cramer et al. [20] showed that the "tag" $(s, r, r \cdot s)$ in [9, 10, 19] could be replaced by an Algebraic Manipulation Detection (AMD) code (see Appendix A for the full

definition of AMD codes). This generalizes the previous construction, and more importantly decouples the size of the secret from the error probability. Using the construction of [20] the share size is $(d+2)\max\left(\frac{1}{d}m, \log\frac{2^n d}{\delta}\right)$. Thus results in an improvement over [9, 10, 19] when the message space $\mathcal{M}$ is very large.

If the adversary's powers are restricted, then the running time of the previous protocols ([9, 10, 19, 20]) can be improved. Suppose the adversary is allowed to corrupt t players, but each corrupt player chooses his strategy based on the view of at most v other corrupted players. This adversary is said to be v-local. Lewko and Pastro [44] can be seen as an adaptation of the schemes of [9, 10, 19, 20]. The dealer shares the secret s, using a t private scheme to obtain shares $s_1, \ldots, s_n$. At a high-level the scheme works like this: the dealer generates a single MAC key, k, but then the dealer generates n tags, $\tau_i = \mathsf{MAC}(\mathsf{k}, s_i)$. Each player then receives three things, a t-private share of s, the tag τ_i and a v-private share of k. The reconstruction algorithm will iterate over all subsets of size $v+1$ to reconstruct k and it will accept the first value of k that successfully verifies $t+1$ of the (s_i, τ_i) pairs. To prove security, they cannot use a standard one-time MAC, since each player receives several MAC tags, and they develop special tools for this purpose.

The scheme of Jhanwar and Safavi-Naini [39] also requires exponential time for reconstruction, but takes a completely different approach. In this scheme, first the secret is shared using a $t+1$ out of $t+1$ threshold secret sharing scheme, then these shares are encoded using a $(t+1, n)$ MDS code,[3] and symbols from this codeword are given to each of the n players. In a $(t+1, n)$ MDS code, any $t+1$ symbols can be extended into a codeword, and this extension is unique. Since the adversary has control of only t symbols, if we look at any collection of $t+2$ symbols, the probability that these $t+2$ symbols are consistent with a codeword is at most $1/|\mathbb{F}|$ because the $(t+2)$nd symbol is determined by the first $t+1$. Thus the reconstructor works by finding the codeword that agrees with the n given symbols in at least $t+2$ locations. Taking a union bound over all subsets of size $t+2$, the probability that the adversary can perturb true codeword to a vector that agrees with any other codeword in at least $t+2$ locations is bounded by $\binom{n}{t+2}|\mathbb{F}|^{-1}$. By choosing $|\mathbb{F}|$ large enough, this can be made arbitrarily small. Unfortunately, the reconstruction procedure requires finding the nearest codeword which takes exponential time. This scheme also has the restriction that $n \geq 2t+2$ (instead of $n \geq 2t+1$). If $n = 2t+1$, then adversary can corrupt t locations at random, leaving only $t+1$ honest participants. Thus the honest codeword will only agree with the received codeword in $t+1$ locations, and hence will be rejected. Bishop et al. [5] pointed out a flaw in this argument, and an attack that renders the scheme insecure.

Safavi-Naini and Wang [52] constructed codes for the adversarial wiretap channel based on combining folded Reed-Solomon codes with AMD codes and subspace-evasive sets. They showed that the adversarial wiretap channel is a generalization of the Secure Message Transmission (SMT) problem, and thus their construction immediately yields an SMT protocol (and hence an RSS protocol) for $n = 2t+1$. Unfortunately, the share size is much larger than specific RSS protocols.

In concurrent, independent work, Cramer et al. [18] constructed efficient robust secret sharing schemes in the model where $t = \left(\frac{1}{2} - \epsilon\right)n$. Their scheme requires three basic building blocks: a length-reducing universal hash function h, an AMD code, and a list-recoverable error-correcting code ECC. Then their RSS scheme shares a secret, s, by

[3] A Maximum Distance Separable (MDS) code is an error correcting code that meets the singleton bound, *i.e.,* it has minimum distance $d = n - k - 1$ where k is the dimension of the code, and n is the block-length.

computing $\mathsf{ECC}(h^{-1}(\mathsf{AMD}(s)))$, and giving each player one symbol of the resulting codeword. Using the list recovery property of ECC, they can recover a list of potential candidate reconstructions, $\{y_i\}$, and using the AMD code, they can identify the correct element from the list. The length-reducing universal hash function h ensures that even if fewer than t symbols of ECC provide partial information about the message, the secret s remains statistically hidden from any adversary holding at most t symbols of $\mathsf{ECC}(h^{-1}(\mathsf{AMD}(s)))$. This scheme can share secrets of size $\mathcal{O}(n+\kappa)$ with shares of size $\mathcal{O}(1+\kappa/n)$ which is information-theoretically optimal, and asymptotically superior to our construction.

In concurrent, independent work, Cheraghchi [17] constructed efficient robust secret sharing schemes in the model where $t = \left(\frac{1}{2} - \epsilon\right) n$. If Shamir sharing is viewed as the analog of a Reed-Solomon code in the context of secret sharing, then Cheraghchi's scheme can be viewed as the secret sharing analog of a folded Reed-Solomon code. In Cheraghchi's scheme, first the dealer encodes the secret, s, with an AMD code [20] and then, the dealer chooses a random polynomial, f, of degree ℓt whose constant term is $\mathsf{AMD}(s)$. Finally, the dealer gives each player ℓ evaluations of the polynomial f. Thus the secret-shares form a codeword in the folded Reed-Solomon code with rate $(t+1)/n$ and folding parameter ℓ. Since a folded Reed-Solomon code of rate R is efficiently list-decodable up to a $(1-R-\epsilon)$ fraction of errors, at reconstruction time, the players can list decode the returned shares to obtain list of possible secrets. Cheraghchi then shows that if the corruptions are introduced by an adversary that only views t shares (ℓt evaluations of f) with high probability the received list will only contain one AMD-encoded value, and thus the true value of the secret can be efficiently identified in the list of possible codewords.

In concurrent, independent work, Bishop et al. [5] constructed efficient robust secret sharing schemes in the regime where $n = 2t+1$, with shares of size $m + \tilde{\mathcal{O}}(k)$. The work of [5] is the first work to achieve robustness with share size that is independent of n in the regime where $n = 2t+1$ – closing the open question posed in [14]. Both our construction and that of [5] build on the work of [51] by starting with an initial secret-sharing scheme (e.g. Shamir) and then having players authenticate each other's shares using a MAC. In the [14] scheme, each player authenticates every other player, requiring each player to store n MAC keys and tags. In the [5] scheme, each player only authenticates a small *random* subset of other players. In this setting, each player needs to store only a small number of MAC keys and tags. On the other hand, reconstruction becomes significantly more complex, because the reconstructor does not know the authentication graph – and malicious players can lie about the players they are supposed to authenticate. Nevertheless, [5] gives an efficient reconstruction procedure based on an efficient algorithm for approximating the approximate graph bisection problem (Table. 1).

When the reconstruction is required to be perfect, *i.e.,* no failure probability is allowed it is known that $n \geq 3t+1$ is a necessary condition, and this bound is achieved by Shamir sharing. Perfectly robust secret sharing has also been studied for more general access structures, and the situation is well understood [42, 46].

A separate line of work considered t-threshold secret sharing schemes, where a group of t cheating players tries to convince a *single* honest player to accept the wrong share. Thus at reconstruction time, there are only $t+1$ players, instead of all n. Since the honest player is outnumbered, there is no way to guarantee correct reconstruction, so the goal of the scheme is simply for the honest player to detect cheating on the part of the other players. This model was introduced by Tompa and Woll [56] and further studied in [42, 43, 48, 49].

Our work considers the regime $t = \left(\frac{1}{2} - \epsilon\right) n$. In this regime, $\epsilon > 0$ can be arbitrarily small, but it is fixed, independent of n and the security parameter κ. In this regime,

Table 1 Comparison of previous RSS schemes

Scheme	Share size	Reconstruction time	Restrictions
[53]	$\max(m, \log(n))$	polynomial	$n \leq 3t + 1$
[51]	$m + \mathcal{O}(\kappa n \log n)$	polynomial	
[14]	$m + \mathcal{O}(\kappa + n)$	polynomial	
[9, 10, 19]	$3\max(m, n + \log(1/\delta))$	exponential	
[20]	$m + \mathcal{O}(\kappa + n)$	exponential	
[39]	$\max(\kappa, m)$	exponential	$n \geq 2t + 2$
[44]	$m + \mathcal{O}(\kappa)$	polynomial	1-local adversary
[18]	$\mathcal{O}(m/n)$	polynomial	$t = \left(\frac{1}{2} - \mathcal{O}(1)\right) n, m > n + \kappa$
[17]	$(1 + o(1))m + \mathcal{O}(k)$	polynomial	$t = \left(\frac{1}{2} - \mathcal{O}(1)\right) n., m > n + \kappa$
[5]	$m + \mathcal{O}(k)$	polynomial	
This Work	$m + \mathcal{O}(\kappa)$	polynomial	$t = \left(\frac{1}{2} - \mathcal{O}(1)\right) n$

In this n denotes the number of players, t denotes the number of corrupted players, κ denotes the security parameter, and m denotes the bit length of the secret being shared. See Appendix B for a more detailed comparison of the share sizes of the various schemes

techniques like share packing [29] can be employed to amortize the cost of secure multiparty computation, and many extremely efficient multiparty computation protocols are known [22–25]. Protocols for secure multiparty computation in the malicious model immediately yield protocols for RSS, but these protocols are less efficient than the dedicated RSS protocols discussed above.

3 Secure message transmission (SMT)

Robust secret sharing is very closely related to the notion of Secure Message Transmission (SMT). In SMT, a sender and receiver are connected via n independent communication channels. An adversary has control over t out of n of these channels (see Fig. 1). The adversary can tamper with information sent over the channels it controls. The problem of secure message transmission (SMT) was formalized by Dolev, Dwork, Waarts and Yung [27].

An r-round n-channel SMT protocol has two guarantees. **Privacy:** an adversary eavesdropping on at most t of the channels learns no information about the secret being communicated. **Robustness:** an adversary tampering with at most t of the channels cannot

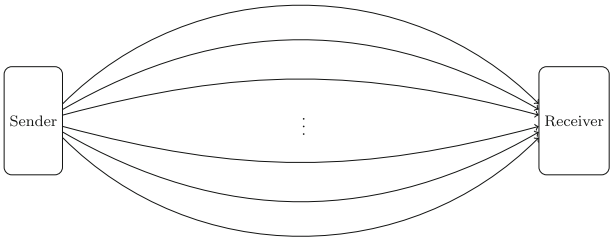

Fig. 1 An n channel SMT protocol. The sender breaks the message into n pieces. A computationally unbounded adversary has complete control over t of the n channels

cause the receiver to receive an incorrect message with probability more than ϵ. When $\epsilon = 0$, the scheme is called perfect.

In [27] it was shown that $n \geq 3t + 1$ is necessary and sufficient for perfect one round SMT protocols (using Reed Solomon codes) and $n \geq 2t + 1$ is necessary and sufficient for perfect two round SMT protocols. Since that time there has been extensive research on SMT protocols [15, 28, 31, 32, 43, 46].

By equating the message sent over channel i to the share given to player i, we can see that a one round SMT protocol is essentially equivalent to an RSS protocol. This equivalence was formalized in [43].

Throughout this work, we use the language and terminology of secret sharing, but by the above equivalence, our RSS protocol can also be viewed as an efficient one-round SMT protocol.

4 Preliminaries

4.1 Notation

A function $\nu(\cdot)$ is called *negligible* if it vanishes faster than the inverse of any polynomial, *i.e.,* for all $c > 0$, $\nu(n) \in \mathcal{O}(n^{-c})$. Throughout this work κ will denote a security parameter. For a set X, we will use the notation $x \leftarrow X$ to denote sampling an element uniformly from X. We use the same notation $y \leftarrow \mathcal{A}(x)$ to denote the result of running the randomized algorithm $\mathcal{A}$ on input x and obtaining an output. For two distributions, X, Y, we use $\Delta(X, Y)$ to denote their statistical distance

$$\Delta(X, Y) = \frac{1}{2} \sum_x |\Pr[X = x] - \Pr[Y = x]| = \sup_A |[\Pr[X \in A] - \Pr[Y \in A]|$$

4.2 Secret sharing

Secret-sharing is a multi-party protocol that allows one player, a dealer, to distribute a secret value among a group of participants such that "authorized" subsets of participants can reconstruct the secret, while the shares held by an "unauthorized" reveal nothing about the underlying secret. In this work, we mostly focus on threshold secret sharing schemes, where there is some threshold t, and every subset of participants of size $t + 1$ is authorized, while every subset of size t (or smaller) is unauthorized. Secret-sharing was introduced by Shamir [53] and Blakley [6].

Definition 1 (Secret Sharing) A pair of randomized algorithms $(\mathsf{SS}, \mathsf{Rec})$ is called an (t, n)-threshold secret sharing protocol over a message space $\mathcal{M}$ if the following properties hold

- **Privacy:** For any $s, s' \in \mathcal{M}$, if $(s_1, \ldots, s_n) \leftarrow \mathsf{SS}(s, 1^\kappa)$ and $(s'_1, \ldots, s'_n) \leftarrow \mathsf{SS}(s', 1^\kappa)$, then for all subsets $A \subset [n]$ with $|A| \leq t$, the distributions $\{s_i\}_{i \in A}$ and $\{s'_i\}_{i \in A}$ are statistically close, *i.e.,*

  $$\Delta(\{s_i\}_{i \in A}, \{s'_i\}_{i \in A}) < \nu(\kappa)$$

  for some negligible function ν. If $\nu = 0$, then the scheme is said to have perfect privacy.
- **Reconstructability:** For all subsets $A \subset [n]$ with $|A| \geq t + 1$, if $(s_1, \ldots, s_n) \leftarrow \mathsf{SS}(s, 1^\kappa)$, then

  $$\Pr[s = \mathsf{Rec}(A, \{s_i\}_{i \in A})] > 1 - \nu(\kappa)$$

where the probability is taken over the coins of SS and Rec. If $\nu = 0$, then we say the scheme has perfect reconstruction.

Definition 2 (Ramp Secret Sharing) A pair of randomized algorithms (SS, Rec) is called an (t, g, n, ϵ)-threshold secret sharing protocol over a message space $\mathcal{M}$ if the following properties hold

- **Privacy:** For any $s, s' \in \mathcal{M}$, if $(s_1, \ldots, s_n) \leftarrow \mathsf{SS}(s, 1^\kappa)$ and $(s'_1, \ldots, s'_n) \leftarrow \mathsf{SS}(s', 1^\kappa)$, then for all subsets $A \subset [n]$ with $|A| \leq t$, the distributions $\{s_i\}_{i\in A}$ and $\{s'_i\}_{i\in A}$ are statistically close, *i.e.,*

$$\Delta(\{s_i\}_{i\in A}, \{s'_i\}_{i\in A}) < \nu(\kappa)$$

  for some negligible function ν. If $\nu = 0$, then the scheme is said to have perfect privacy.
- **Gap Reconstructability:** For all subsets $A \subset [n]$ with $|A| \geq t + g$, if $(s_1, \ldots, s_n) \leftarrow \mathsf{SS}(s, 1^\kappa)$, then

$$\Pr[s = \mathsf{Rec}(A, \{s_i\}_{i\in A})] > 1 - \nu(\kappa)$$

  where the probability is taken over the coins of SS and Rec. If $\nu = 0$, then we say the scheme has perfect reconstruction.

Note that in a ramp scheme, coalitions of between t and $t + g$ players may be able to learn some information about the secret. One of the most common secret sharing schemes is Shamir Sharing [53].

Definition 3 (Ramp Shamir Sharing) Fix a finite field $\mathbb{F}$ with $|\mathbb{F}| \geq n + g$, and distinct points $\{\alpha_1, \ldots, \alpha_n\} \in \mathbb{F} \setminus \{\beta_1, \ldots, \beta_g\}$.

– **Sharing:** To share a message $\mathbf{m} \in \mathbb{F}^g$, choose a random polynomial, f, of degree $t + g - 1$ in $\mathbb{F}[x]$ subject to the constraints that

$$f(\beta_i) = m_i \text{ for } i = 1, \ldots, g$$

  The ith share of will be $f(\alpha_i) \in \mathbb{F}$.
– **Reconstruction:** Any $t + g$ players can reconstruct the polynomial f by interpolation, and recover the secret m.

The t-privacy follows from the fact that if an adversary learns $t + g$ evaluations of f, then the remaining evaluations remain uniformly distributed.

Many general constructions of secret-sharing schemes exist [3, 4, 8, 21, 57]. Secret sharing schemes for general (non-threshold) access structures have been considered [33, 36]. Secret sharing schemes can be viewed as matroids [45] or monotone span programs [1, 30]. A survey of secret sharing schemes can be found in [2].

In this work, we focus on robust secret sharing (described in Section 4.3).

4.3 Robust secret sharing

Threshold secret sharing allows a dealer to distribute a secret among n players so that any t players learn nothing about the secret, but any $t + 1$ players can reconstruct the secret.

A secret sharing protocol is called robust, if the recovery procedure succeeds (with high probability) even if a coalition of t players maliciously tampers with their shares.

Definition 4 (RSS) An n-player secret sharing scheme $(\mathsf{SS}, \mathsf{Rec})$ is (t, δ)-robust if $(s_1, \ldots, s_n) \leftarrow \mathsf{SS}(s, 1^\kappa)$ and the following properties hold

- **Privacy:** For any $s, s' \in \mathcal{M}$, if $(s_1, \ldots, s_n) \leftarrow \mathsf{SS}(s, 1^\kappa)$ and $(s'_1, \ldots, s'_n) \leftarrow \mathsf{SS}(s', 1^\kappa)$, then for all subsets $A \subset [n]$ with $|A| \leq t$, the distributions $\{s_i\}_{i\in A}$ and $\{s'_i\}_{i\in A}$ are statistically close, *i.e.,*

$$\Delta(\{s_i\}_{i\in A}, \{s'_i\}_{i\in A}) < \nu(\kappa)$$

  for some negligible function ν. If $\nu = 0$, then the scheme is said to have perfect privacy.
- **Reconstructability** For all subsets $A \subset [n]$ with $|A| \leq t$, and any adversary $\mathcal{A}$, if $\{s'_i\}_{i\in A} \leftarrow \mathcal{A}(\{s_i\}_{i\in A}, 1^\kappa)$, and $s_i = s'_i$ for $i \in [n] \setminus A$, then

$$\Pr\left[\mathsf{Rec}(s'_1, \ldots, s'_n) \neq s\right] < \delta$$

Note that unlike Verifiable Secret Sharing (VSS) in RSS schemes the dealer is assumed to be honest.

The primary concern will be the size of the shares and the efficiency of the reconstruction procedure. We also introduce the notion of a nested RSS, which slightly strengthens the notion of an RSS. In a nested RSS reconstruction can succeed even if only a subset of the shares are available at reconstruction time. Thus in a nested RSS scheme, $t + g$ shares are needed to reconstruct if the shares are all correct, and any collection of $t + g + (1 - \epsilon)c$ correct shares and ϵc incorrect shares, will also allow reconstruction with failure probability at most δ.

Definition 5 (Nested RSS) An n-player secret sharing scheme $(\mathsf{SS}, \mathsf{Rec})$ is a nested (t, g, δ, ϵ)-robust secret sharing scheme if it satisfies the following properties:

- It is a (t, g) Ramp Secret Sharing Scheme
- **Reconstructability** For all subsets $A \subset [n]$ with $|A| \leq t$, and any adversary $\mathcal{A}$, if $\{s'_i\}_{i\in A} \leftarrow \mathcal{A}(\{s_i\}_{i\in A}, 1^\kappa)$, and $s_i = s'_i$ for $i \in [n] \setminus A$, then for any $B \subset [n]$ with $|B| = \ell$, if $|A \cap B| < \epsilon(|B| - t - g)$ then

$$\Pr\left[\mathsf{Rec}(B, \{s'_i\}_{i\in B}) \neq s\right] < \delta.$$

The reconstruction procedure is described from the point of view of a single player. If all players want to reconstruct the secret, they will need to send their shares to each other player, and then separately run the reconstruction procedure. In RSS, because the dealer is assumed to be honest, even if all players want to reconstruct their secret, there is no need for a broadcast channel. If corrupt players send different, malformed shares to each party during reconstruction, the robustness ensures that each honest party will separately reconstruct the correct secret. Unlike the model of Tompa and Woll [56], in RSS shares are provided by all players (but dishonest players can provide arbitrary shares to the reconstruction procedure).

Note that because Shamir shares $(f(\alpha_1), \ldots, f(\alpha_n))$ correspond to a $[t + 1, n]$ Reed Solomon codeword, and the Reed Solomon code has minimum distance $n - t$, the original codeword (and hence the shared secret) can be recovered even if $\frac{n-t}{2}$ shares are corrupted. Thus Shamir sharing is robust as long as $\frac{n-t}{2} > t$, which means $n > 3t$. In this situation, robust reconstruction of Shamir shares can be done efficiently using the Berlekamp-Welch algorithm for decoding Reed Solomon codes. This yields the following fact

Fact 1 (Robustness of Shamir Sharing) *The Shamir sharing scheme is a* $\left(\left\lfloor \frac{n-1}{3} \right\rfloor, 0\right)$ *RSS scheme, with shares of size* $\log|\mathcal{M}|$. *The ramp Shamir sharing scheme with gap g is a*

$\left(\left\lfloor \frac{n-g}{3} \right\rfloor, 0\right)$ *RSS with shares of size* $\log|\mathcal{M}|/g$. *In fact, the error correcting properties of the Reed Solomon code mean that the ramp Shamir sharing scheme is a* $\left(t, g, 0, \frac{1}{2}\right)$ *nested RSS. This is just the statement that given any* $t + g + \ell$ *evaluations, where at most* $\frac{1}{2}\ell$ *are erroneous, you can efficiently reconstruct the unique degree* $t+g$ *polynomial going through those points.*

4.4 Message authentication codes (MACs)

Our construction relies on simple, unconditionally-secure, one-time Message Authentication Codes (MACs) [35, 54]. MACs take a message and key and output a "tag" that can be used to authenticate the message. Many types of MACs exist, but we only need a one-time MAC. In particular, we require that an adversary who sees a single valid message-tag pair, cannot generate a new, valid message-tag pair. A standard method for constructing information-theoretic one-time MACs is to use Universal hash functions [12]. Note that these information-theoretic MACs are much simpler than MACs that satisfy the stronger notion of unforgeability under chosen message attack (see Appendix A).

Definition 6 A deterministic function $\mathsf{MAC} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is called a $(\mathcal{K}, \mathcal{M}, \delta)$-MAC if for all $m_1, m_2 \in \mathcal{M}$, and for all $\tau_1, \tau_2 \in \mathcal{T}$ and $m_1, m_2 \in \mathcal{M}$

$$\Pr_{\mathsf{k}\leftarrow\mathcal{K}}[\mathsf{MAC}(\mathsf{k}, m_2) = \tau_2 | \mathsf{MAC}(\mathsf{k}, m_1) = \tau_1] < \delta$$

These MACs are easy to construct, and for concreteness, we recall a simple construction of secure MACs based on polynomials.

Theorem 2 [14, 26, 40, 55] *Let* q *be a prime power, and* $\ell > 0$ *an integer. Let* $\mathcal{M} = \mathbb{F}_q^\ell$ *and* $\mathcal{K} = \mathbb{F}_q^2$ *and* $\mathcal{T} = \mathbb{F}_q$ *then*

$$\mathsf{MAC}((\mathsf{k}_1, \mathsf{k}_2), m) = \sum_{i=1}^{\ell} m_i \mathsf{k}_1^i + \mathsf{k}_2$$

is an $\left(\mathbb{F}_q^2, \mathbb{F}_q^\ell, \frac{\ell}{q}\right)$*-MAC.*

Proof For any fixed $m \in \mathbb{F}_q^\ell$ and $\tau \in \mathbb{F}_q$ define the polynomial

$$f(x) = x^\ell + \sum_{i=1}^{\ell} m_i x^i - \mathsf{k}_2 - \tau$$

Then f is a polynomial of degree at most ℓ, so f has at most ℓ roots in $\mathbb{F}_q$. Thus

$$\Pr_{\mathsf{k}\leftarrow\mathcal{K}}[\mathsf{MAC}(\mathsf{k}, m) = \tau] = \Pr_{\mathsf{k}\leftarrow\mathcal{K}}[f(\mathsf{k}) = 0] \leq \frac{\ell}{q}$$

Now, the adversary succeeds in creating a forgery exactly when k_1 is a root of the polynomial f. Since k_1 is uniformly random conditioned on the adversary's view of a single message-tag pair, the adversary succeeds in forging a tag with probability at most $\frac{\ell}{q}$. □

4.5 Expander graphs

Our construction relies on expander graphs, and in this section we briefly review some basic concepts. See [37] for an in-depth survey.

For every d regular graph on n nodes, we can create the $n \times n$ adjacency matrix A. To normalize A, we divide each entry by d, to obtain the matrix A'. The regularity of the graph ensures that each row and column of A' has weight 1. It is straightforward to check that the all ones vector is an eigenvector of A with eigenvalue 1, and all other eigenvectors have eigenvalue bounded by 1 in absolute value. The algebraic *expansion* of the graph is determined by the size of the second largest eigenvalue of A' denoted λ.

A d regular graph, G with n nodes is called an algebraic expander with expansion λ if λ is the absolute value of the second largest eigenvalue of the normalized adjacency matrix of G. Thus $0 \leq \lambda \leq 1$, and the closer λ is to zero, the better the *expansion* of G.

One of the important properties of expanders is that for any subsets of nodes, S and T, of size $\mathcal{O}(n)$, the number of edges between S and T is essentially the expected value $\frac{d|S||T|}{n}$. This is formalized in the Expander Mixing Lemma.

Lemma 1 (Expander Mixing Lemma) *Let G be a d-regular expander on n vertices with normalized second eigenvalue λ. For any sets of vertices, S, T, in G*

$$\left| E(S,T) - \frac{d}{n}|S||T| \right| < \lambda d\sqrt{|S||T|}$$

Where $E(S,T)$ denotes the number of edges between S and T.

An expander graph is called a Ramanujan graph if $\lambda < \frac{2}{\sqrt{d}}$. See [47] for a survey of Ramanujan graphs.

Our RSS protocol will identify each of the n players with nodes in an expander graph, and each player will be relied upon to check the shares of his neighbors.

5 Construction

In this section, we explain in detail our new RSS protocol. Previous RSS protocols used MACs to allow each participant to check the shares provided by other participants in the reconstruction phase. These MACs provide robustness, but the overhead of sharing a collection of MACs causes a blowup in the size of the shares held by each player. In the protocol of Rabin and Ben-Or, each party receives a MAC of each other player's share. Since the MAC has keys and tags of size $\mathcal{O}(\kappa)$, the share size blows up by $\mathcal{O}(n\kappa)$. Cevallos et al., showed that the MAC keys and tags could be reduced from size $\mathcal{O}(\kappa)$ to $\mathcal{O}(\log n)$ by using a more complex recovery procedure. This reduced the overhead to $\mathcal{O}(n + \kappa)$. In both protocols, each player maintains a MAC to check *every* other player's share. When everyone checks everyone else, a blowup of $\mathcal{O}(n)$ is inevitable.

In this work, we change the paradigm, and each player only checks *a constant number of other players*. In particular, we consider a d-regular graph on n nodes, and we associate each node of the graph with one of the n players. Then each player will only check its d neighbors in the graph. Thus each player will only have to maintain d MAC keys and tags. When tags are of size $\mathcal{O}(\kappa)$, this results in an overhead of $\mathcal{O}(d\kappa)$ (instead of $\mathcal{O}(n\kappa)$).

Because constant degree expanders exist, we can choose d to be a constant, independent of n. An exact choice of parameters is deferred until later in this section.

Intuitively, there are many reasons for distributing MAC keys and tags according to the edges of an expander graph. First, because the degree is low, each player (node) must store only a small (constant) number of keys and tags. Second, if the graph is a vertex expander then any (small) set of malicious players (nodes) will be connected to an even larger set of honest players, who will attempt to validate the shares provided by the malicious nodes.

The idea of having computations performed by small "committees" dates back to Bracha [7], and has been used in a variety cryptography constructions including MPC [36] and SMT [28].[4] The idea of distributing keys according to expander graphs has also been used to reduce storage complexity in key predistribution schemes for wireless sensor networks [13, 34, 41].

In concurrent, independent work, [5] also designed an RSS protocol where each player only authenticates a small number of neighbors. The key difference between our schemes is that we use a fixed (expander) graph, whereas they use a random graph that is generated when the shares are distributed. Using a random graph decreases the adversary's ability to create mangled shares since the adversary only knows the neighborhood structure of the t corrupted parties. This allows them to handle the maximum number of corruptions ($n = 2t + 1$), whereas our scheme only works when $t = \left(\frac{1}{2} - \epsilon\right) n$. On other hand, since the graph is not fixed in [5], the adversary can potentially modify the neighborhood structure of the corrupted players, and thus reconstruction becomes more difficult. In our protocol, reconstruction requires simply taking a majority vote for each player, whereas in [5] reconstruction requires (efficiently) solving the approximate graph bisection problem.

Since each player only checks a small number of other players, the recovery algorithm has to be adapted to ensure that no coalition of t malicious parties can succeed in fooling the honest players into accepted mangled shares.

Let G be a d-regular graph, and let $\Gamma(i)$ denote the set of d players that are neighbors of player i. Let $\mathsf{MAC} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ a $(\mathcal{K}, \mathcal{M}, \delta')$-MAC. Our proposed scheme is presented in Fig. 2.

5.1 Expander construction

Let G be a d-regular expander on n vertices with normalized second eigenvalue λ. Let $\mathsf{MAC} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ a $(\mathcal{K}, \mathcal{M}, \delta')$-MAC. Since each of the n players will authenticate d others, we will set $\delta' < \frac{\delta}{nd}$ so the probability that an adversary can forge *any* of these tags is bounded by δ.

Theorem 3 *If* $(\mathsf{SS}_0, \mathsf{Rec}_0)$ *is a nested* $\left(\left(\frac{1}{2} - \epsilon\right) n, g, \delta, \epsilon_0\right)$ *RSS with share size* s_0 *and* G *is a* d*-regular graph on* n *vertices with normalized second eigenvalue* $\lambda < \sqrt{\frac{\epsilon^3 \epsilon_0}{\left(\frac{1}{2} - \epsilon\right)(1 - \epsilon_0)}}$

[4] In the work of Fitzi et al. the "committees" are not constructed according to nodes in an expander graph, but instead *every* committee of size d is constructed, resulting n^d committees of the n underlying players. Fitzi et al. are primarily concerned with Perfectly Secure Message Transmission, and so their construction requires two rounds of communication (a message from receiver to sender, and then a message from sender to receiver). By contrast, when viewed as a message transmission scheme, our construction has only one round, but has a negligible probability of failure.

To share a secret, $s \in \mathcal{M}$:

- Share s using a $(t, g, \delta', \epsilon_0)$ nested RSS $(\mathsf{SS}_0, \mathsf{Rec}_0)$. Let $(s_1, \ldots, s_n) \leftarrow \mathsf{SS}_0(s, 1^\kappa)$ be the resulting shares.
- For each $i \in [n]$ and $j \in \Gamma(i)$ generate a MAC key, $\mathsf{k}_{ij} \leftarrow \mathcal{K}$, and create the tag

$$\tau_{ij} = \mathsf{MAC}(\mathsf{k}_{ij}, s_j)$$

- Give player i the share s_i and the tags $\{\mathsf{k}_{ij}, \tau_{ji}\}_{j \in \Gamma(i)}$

To reconstruct a secret from shares $\{s_i, \{\mathsf{k}_{ij}, \tau_{ji}\}_{j \in \Gamma(i)}\}_i$

- Initialize $\mathsf{G} = \emptyset$.
- For each $i \in [n]$, if $|\{j : \mathsf{MAC}(\mathsf{k}_{ij}, s_j) = \tau_{ij}\}| > \frac{d}{2}$. then $\mathsf{G} = \mathsf{G} \cup \{i\}$.
- Discard all indices $i \in [n] \setminus \mathsf{G}$, and use the recovery algorithm Rec_0 of the $(t, g, \delta', \epsilon_0)$ nested RSS scheme to recover s, from the indices in G.

Fig. 2 Robust secret sharing from expander graphs

then the scheme described in Fig. 2 *is a* $\left(\frac{1}{2} - \epsilon\right)$ *n-robust secret sharing scheme with error probability* $nd\delta' + \delta$ *and share size*

$$\mathsf{s}_0 + d\,(\log|\mathcal{T}| + \log|\mathcal{K}|)$$

Proof Let B be the set of indices corrupted by the adversary, thus $|\mathsf{B}| = t$.

Notice that if player i is corrupted, and player j is honest, the probability that the adversary can generate a share s_i' and tag $\tau_{ji}' = \mathsf{MAC}(\mathsf{k}_{ji}, s_i')$ is at most δ' by the security of the MAC.

Thus with probability at least $1 - nd\delta'$ the adversary fails to generate a single forged tag. Throughout the rest of the argument, we will condition on the event that adversary fails to generate a single forged tag.

Let G be the set of indices where at least $\frac{d}{2} + 1$ tags verify (described in the reconstruction procedure in Fig. 2). Recall that B is the set of corrupted players.

Let

$$\mathsf{M}_1 = (\mathsf{G} \cup \mathsf{B})^c \qquad \text{and} \qquad \mathsf{M}_2 = \mathsf{G} \cap \mathsf{B}$$

Thus M_1 is the set of honest players (incorrectly) rejected by the reconstruction procedure and M_2 is the set of dishonest players accepted by the reconstruction procedure.

The RSS scheme $(\mathsf{SS}_0, \mathsf{Rec}_0)$ can recover from an ϵ_0 fraction of errors, Thus the reconstruction procedure will succeed if

$$|\mathsf{M}_2| < \epsilon_0\,(|\mathsf{G}| - t - g)$$

Now, we have

$$|\mathsf{G}| = \underbrace{n - |\mathsf{B}|}_{\#\text{ honest players}} - \underbrace{|\mathsf{M}_1|}_{\#\text{ honest players rejected}} + \underbrace{|\mathsf{M}_2|}_{\#\text{ dishonest players accepted}}$$

Thus the reconstruction succeeds if

$$|\mathsf{M}_2| < \epsilon_0\,(n - t - g - |\mathsf{B}| - |\mathsf{M}_1| + |\mathsf{M}_2|)$$

Which is equivalent to

$$\frac{1-\epsilon_0}{\epsilon_0}|\mathsf{M}_2| + |\mathsf{M}_1| < n - |\mathsf{B}| - t - g$$

Since $\epsilon_0 < \frac{1}{2}$, a sufficient condition for recovery is

$$\frac{1-\epsilon_0}{\epsilon_0}(|\mathsf{M}_1 \cup \mathsf{M}_2|) < n - |\mathsf{B}| - t - g$$

Let $\mathsf{M} = \mathsf{M}_1 \cup \mathsf{M}_2$ be the set of players incorrectly classified in the reconstruction phase. For a player to be incorrectly classified, at least $d/2$ of its neighbors must have provided an incorrect MAC, thus

$$\mathsf{M} \subset \{i : |\Gamma(i) \cap \mathsf{B}| > d/2\}$$

Thus $E(\mathsf{M}, \mathsf{B}) \geq \frac{d}{2}|\mathsf{M}|$. On the other hand, the Expander Mixing Lemma states that

$$|E(\mathsf{M}, \mathsf{B})| \leq \frac{d}{n}|\mathsf{M}||\mathsf{B}| + \lambda d\sqrt{|\mathsf{M}||\mathsf{B}|}$$

Thus

$$\frac{|\mathsf{M}|}{2} \leq \frac{|\mathsf{M}||\mathsf{B}|}{n} + \lambda\sqrt{|\mathsf{M}||\mathsf{B}|}$$

Rearranging, we have

$$|\mathsf{M}| < \frac{\lambda^2|\mathsf{B}|}{\left(\frac{1}{2} - \frac{|\mathsf{B}|}{n}\right)^2}$$

Thus reconstruction succeeds if

$$\frac{1-\epsilon_0}{\epsilon_0}\left(\frac{\lambda^2|\mathsf{B}|}{\left(\frac{1}{2} - \frac{|\mathsf{B}|}{n}\right)^2}\right) < n - |\mathsf{B}| - t - g$$

When $t = |\mathsf{B}| = \left(\frac{1}{2} - \epsilon\right)n$, then this means the reconstruction succeeds if

$$\frac{1-\epsilon_0}{\epsilon_0}\left(\frac{\lambda^2\left(\frac{1}{2} - \epsilon\right)n}{\epsilon^2}\right) < 2\epsilon n - g$$

If $g < \epsilon n$, then a sufficient condition is

$$\lambda^2 < \frac{\epsilon^3\epsilon_0}{\left(\frac{1}{2} - \epsilon\right)(1-\epsilon_0)}$$

If G is a Ramanujan graph, then $\lambda < \frac{2}{\sqrt{d}}$, so it suffices to take $d > \frac{4\left(\frac{1}{2}-\epsilon\right)(1-\epsilon_0)}{\epsilon^3\epsilon_0}$

Thus for this choice of d reconstruction will succeed unless the adversary successfully forges a tag (which happens with probability at most $nd\delta'$) or the inner recovery algorithm Rec_0 fails (which happens with probability at most δ). □

Although many infinite families of explicit Ramanujan graphs are known, explicit Ramanujan graphs are not known for all n, d pairs, which limits the applicability of the above construction. In the next section, we show that the same construction holds (with high probability) for a random graph.

5.2 Random graphs

It is well-known that a random d-regular graph will be an expander. In particular, [50] shows that a random d-regular graph has $\lambda < 2\sqrt{d-1}+1$ asymptotically almost surely. This does not allow us to apply the scheme in Fig. 2 directly, however, because that analysis does not give an explicit bound on the probability the graph fails to be an expander (only that it tends to zero as $n \to \infty$).

It is straightforward to show, however, that if the underlying graph is chosen at random, our construction still has only a negligible failure probability, where now the failure probability is taken over the internal randomness of the sharing algorithm, the internal randomness of the adversary and the choice of graph.

Using a random graph has the drawback that reconstruction algorithm needs to be told the graph structure in order to successfully reconstruct the secret. The concurrent, independent work of [5] uses a similar technique, where each player authenticates a random set of other players, but they do not require storing the entire graph, instead they rely on a more complex reconstruction procedure that can infer the graph structure based on each player's (possibly corrupted) neighbor sets. This has the advantage that the adversary only sees a partial view of the authentication graph, and thus cannot choose corruption patterns based on the entire graph.

5.3 Instantiations

Throughout this section, we use a standard MAC (see Theorem 2), that has keys and tags of length $\log q$, messages of length $\ell \log q$ and security $\delta' = \frac{\ell}{q}$.

Corollary 1 (Using Shamir Sharing) *Instantiating our schemes with* $(\mathsf{SS}_0, \mathsf{Rec}_0)$ *as Shamir Sharing, which is a* $(t, 1, 0, \frac{1}{2})$ *nested RSS, gives a* $\left(\left(\frac{1}{2}-\epsilon\right)n, \delta\right)$*-secure RSS scheme with shares of size*

$$\log|\mathcal{M}| + \frac{4}{\epsilon^3}\log\frac{3n\log|\mathcal{M}|}{\epsilon^3\delta}$$

Proof Thus shares are of size

$$\mathsf{s}_0 + d(\log|\mathcal{T}| + \log|\mathcal{K}|)$$

For Theorem 3 we need $d > \frac{4\left(\frac{1}{2}-\epsilon\right)(1-\epsilon_0)}{\epsilon^3\epsilon_0}$ Since $\epsilon_0 = \frac{1}{2}$, this becomes $d > \frac{4\left(\frac{1}{2}-\epsilon\right)}{\epsilon^3}$. Thus it suffices to choose $d > \frac{2}{\epsilon^3}$.

Since we need $\delta' < \frac{\delta}{nd} < \frac{\delta\epsilon^3}{2n}$, in our MAC, we need $q > \frac{2n\ell}{\epsilon^3\delta}$. The MAC supports messages of length $\ell \log q$, and the shares being signed are of size s_0, thus $\ell < \mathsf{s}_0$, and it suffices to take $q = \frac{2n\mathsf{s}_0}{\epsilon^3\delta}$. In the Shamir Sharing scheme $\mathsf{s}_0 = \log|\mathcal{M}|$, so this results in shares of size

$$\log|\mathcal{M}| + \frac{4}{\epsilon^3}\log\frac{3n\log|\mathcal{M}|}{\epsilon^3\delta}$$

□

Comparing this to [14], which has shares of size

$$\log|\mathcal{M}| + 12\log\frac{1}{\delta} + 3n\,(\log(t+1) + \log\log|\mathcal{M}| + 3)$$

Our new scheme has shares of size

$$\log|\mathcal{M}| + \frac{4}{\epsilon^3}\log\frac{1}{\delta} + \frac{4}{\epsilon^3}\left(\log 3n + \log\log|\mathcal{M}| + \log\frac{1}{\epsilon^3}\right)$$

In particular, when $\epsilon = \mathcal{O}(1)$, we have an overhead of $\mathcal{O}\left(\log\frac{1}{\delta}\right)$ instead of $\mathcal{O}\left(\log\frac{1}{\delta} + n\right)$ as in [14].

Note that in the extreme case, $n = 2t + 1$, then $\epsilon = \frac{1}{2n}$, and since $d > \frac{2}{3\epsilon^3}$, our scheme would require $\mathcal{O}(n^3)$ tags per share, which is worse than existing schemes [14, 51].

We can improve these bounds slightly by moving to Ramp Shamir Sharing

Corollary 2 (Using Ramp Shamir Sharing) *For any $g < \epsilon n$, we reduce s_0 to $\log|\mathcal{M}|/g$, and so the size of the shares becomes*

$$\frac{\log|\mathcal{M}|}{g} + \frac{4}{\epsilon^3}\log\frac{3n\log|\mathcal{M}|}{g\epsilon^3\delta}$$

Using secret sharing schemes based on Cramer and Chen's ramp-based secret sharing scheme based on algebraic-geometry codes [16], we can reduce the share size to $\mathcal{O}\left(\frac{\log n}{\epsilon^3}\right)$.

6 Conclusion

In this work, we give the first RSS protocol with efficient reconstruction and shares of size $\tilde{\mathcal{O}}(\kappa)$. Our protocol works by treating each of the n players as a node in a d-regular expander graph, and having each player check only its d neighbors. This diverges from previous protocols [14, 51] where each player must check *all* other players in the protocol. Since expander graphs exist with extremely low degree exist, our protocol makes significant gains in share size and complexity of reconstruction. In fact, when the number of malicious parties is $t = \left(\frac{1}{2} - \epsilon\right)n$, for an constant $\epsilon > 0$, our protocol achieves essentially the optimal share size. Unfortunately, when $n = 2t + 1$, then $\epsilon = \mathcal{O}\left(\frac{1}{n}\right)$ and our scheme is outperformed by existing schemes. In concurrent, independent work [5] shows how a similar technique can be adapted to create RSS protocols with $\mathcal{O}(\kappa)$ overhead when $n = 2t + 1$.

Acknowledgments This work was supported in part by NSF grants 1513671, 1619348, 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, Lockheed-Martin Corporation Research Award and by DARPA Safeware program. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

Appendix A: Authentication schemes

A.1 Message authentication codes (MACs)

In this section, we recall the notion *unforgeability under chosen message attack* for Message Authentication Codes. This is the standard notion of security for MACs. For our purposes,

we need a much weaker notion of security (see Theorem 2). We include the standard definition for reference purposes only.

Definition 7 A Message Authentication Code (MAC) is a pair of deterministic algorithms (MAC, Ver)

$$\begin{aligned} \mathsf{MAC} : \mathcal{K} \times \mathcal{M} &\to \mathcal{T} \\ (\mathsf{k}, m) &\mapsto \tau \end{aligned}$$

and

$$\mathsf{Ver} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \{0, 1\}$$

such that

$$\mathsf{Ver}(\mathsf{k}, \mathsf{MAC}(\mathsf{k}, m), m) = 1$$

for all $m \in \mathcal{M}$.

Security is defined through the following experiment

Experiment $\mathbf{exp}^{\mathsf{uf-cma}}_{\mathsf{MAC}}(\mathcal{A})$

- $\mathsf{k} \leftarrow \mathcal{K}$
- The adversary, $\mathcal{A}$ can make repeated queries to the oracles $\mathsf{MAC}(\mathsf{k}, \cdot)$ and $\mathsf{Ver}(\mathsf{k}, \cdot, \cdot)$.
- If $\mathcal{A}$ makes a query τ, m to $\mathsf{MAC}ver(\mathsf{k}, \cdot, \cdot)$ such that
  - $\mathsf{MAC}ver(\mathsf{k}, \tau, m) = 1$
  - The message, m, was never made as a query to the oracle $\mathsf{MAC}(\mathsf{k}, \cdot)$.

  then return 1, otherwise, return 0

The MAC is called secure (existentially unforgeable against a chosen message attack) if

$$\Pr\left[\mathbf{exp}^{\mathsf{uf-cma}}_{\mathsf{MAC}} = 1\right] < \nu$$

for some negligible function $\nu(\log |\mathcal{K}|)$.

A.2 Algebraic manipulation detection (AMD) codes

An Algebraic manipulation detection (AMD) code is means of encoding information so that tampering by an oblivious adversary is detectable. AMD have been widely used, but were first formalized by Cramer et al. in [20].

Definition 8 (AMD Codes) A pair of functions (AMD, Ver) is called a $(\mathcal{M}, \mathcal{T}, \delta)$-algebraic manipulation detection (AMD) code if AMD is a probabilistic map $\mathsf{AMD} : \mathcal{M} \to \mathcal{T}$, and Ver is a deterministic map $\mathsf{Ver} : \mathcal{T} \to \mathcal{M} \cup \{\perp\}$ such that for $\mathcal{T}$ is a group and for all $m \in \mathcal{M}$ and $\Delta \in \mathcal{T}$

$$\Pr\left[\mathsf{Ver}(\mathsf{AMD}(m) + \Delta) \notin \{m, \perp\}\right] < \delta$$

We briefly recall a simple construction of AMD Codes given in [20].

Theorem 4 (Theorem 2 in [20]) *Let p be prime, d an integer such that $p \nmid d+2$ and q a power of p. Then*

$$\mathsf{AMD}(m) = \left(m, x, x^{d+2} + \sum_{i=1}^{d} s_i x^i\right)$$

is an $\left(\mathbb{F}_q^d, \mathbb{F}_q^d \times \mathbb{F}_q \times \mathbb{F}_q, \frac{d+1}{q}\right)$-AMD code.

Appendix B: Calculating share size in existing schemes

There are three parameters of interest when calculating the size of shares

- δ The probability of reconstruction failure
- n The number of participants
- m The bit length of the message

We will be using these parameters to define the share size s.

Ignoring robustness, to ensure correctness we need $\mathsf{s} \geq m$. Using Shamir sharing also introduces the requirement $\mathsf{s} \geq \log(n)$.

[51] In this scheme, the secret $s \in \mathbb{F}$ is shared using Shamir sharing resulting in shares $s_1, \ldots, s_n$. Then random, $b_{ij} \neq 0$ and y_{ij} are created and $c_{ij} = s_i + b_{ij} y_{ij} \mod p$ where $p \geq |\mathbb{F}|$. Then player i receives the shares $(s_i, \{y_{ij}\}_j, \{b_{ji}, c_{ji}\}_j)$. These shares are of size $\log|\mathbb{F}| + 3n \log p$. The probability that player i catches player j cheating is $1 - p^{-1}$, thus the probability that all cheaters are caught by all honest players is at least $1 - t(n-t)/p$. Thus the cheating probability is bounded by $t(n-t)/p \approx n^2/p$. So we need to choose $p = \frac{n^2}{\delta}$, which results in share size

$$\mathsf{s} = m + 3n \log \frac{n^2}{\delta}$$

[14] This scheme is very similar to [51], except the MAC used to authenticate shares is weaker, and the reconstruction algorithm is more complex. In particular, the secret, s is Shamir shared into $\{s_i\}$, and the shares are signed $\tau_{ij} = \mathsf{MAC}(\mathsf{k}_{ij}, s_i)$. Player i then receives $(s_i, \{\mathsf{k}_{ji}\}_j, \{\tau_{ij}\}_j)$. If MAC has security δ', then the security of the overall scheme is $e((t+1)\delta')^{(t+1)/2}$. Standard MACs can achieve security $2^{-\kappa}m$ with tags of length λ and keys of length 2λ, and messages of length m. Setting $\kappa = \log(t+1) + \log m + \frac{2}{t+1}(\log \frac{1}{\delta}) + \log e$ yields a scheme with security δ and the resulting share size is

$$\mathsf{s} = \max\left(m + 12 \log \frac{1}{\delta} + 3n(\log(t+1) + \log m + 3), \log n\right)$$

[9, 10] A secret, $s \in \mathbb{F}$ is encoded as $(s, r, r \cdot s) \in \mathbb{F}^3$, and then shared using a $t+1$ out-of-n Shamir sharing scheme. Given a set of $t+1$ shares, the probability that the adversary can cause this to decode to $(s', r'r' \cdot s')$ is $1/|\mathbb{F}|$. Taking a union bound over all subsets of size $t+1$ gives an error probability of $\{0,1\}omnt + 1|\mathbb{F}|^{-1}$. Thus we need $|\mathbb{F}| \geq \{0,1\}omnt + 1\delta^{-1}$. This yields

$$\mathsf{s} \geq 3 \max\left(m, \log\left(\{0,1\}omnt + 1\delta^{-1}, n\right)\right)$$

[20] A secret, $s \in \mathbb{F}$ is encoded as $\mathsf{AMD}(s)$, and then shared using a $t+1$ out-of-n Shamir sharing scheme. Using the AMD codes proposed in that paper, $\mathsf{AMD}(s) =$

$x^{\ell+2} + \sum_{i=1}^{\ell} s_i x^i$ for ($s \in \mathbb{F}^d$ and $x \in \mathbb{F}$) yields a code with detection probability $(\ell+1)/|\mathbb{F}|$. Since reconstruction requires testing all subsets of $t+1$ shares, we have to union bound over $\{0,1\}omnt + 1$ subsets, so the error probability is at most $\{0,1\}omnt + 1(\ell+1)|\mathbb{F}|^{-1}$. Thus we need $|\mathbb{F}| \geq (\ell+1)\{0,1\}omnt + 1\delta^{-1}$, but this extra parameter, ℓ, gives us flexibility. Since the message space is now $\mathbb{F}^{\ell}$, the resulting shares are of size

$$\mathsf{s} \geq \max\left((\ell+2)\log\left(\{0,1\}omnt + 1(\ell+1)\delta^{-1}\right), \frac{\ell+2}{\ell}m, \log(n)\right)$$

When m is very large, we can use the parameter ℓ to balance the first and second terms in the expression.

[39] A secret, $s \in \mathbb{F}$ is shared using a $t+1$ out-of-$t+1$ Shamir sharing scheme. Then this vector in $\mathbb{F}^{t+1}$ is encoded using a $(t+1, n)$ MDS code, and each player receives one symbol of the resulting codeword, thus the shares are of size $\mathbb{F}$. Like the previous schemes, the probability of error is $\{0,1\}omnt + 1|\mathbb{F}|^{-1}$.

$$\mathsf{s} \geq \max\left(m, \log\left(\{0,1\}omnt + 1\delta^{-1}, n\right)\right)$$

(but this scheme works only when $n \geq 2t+2$ instead of $n \geq 2t+1$)

References

1. Beimel, A.: Secure schemes for secret sharing and key distribution. PhD thesis, Technion (1996)
2. Beimel, A.: Secret-sharing schemes: a survey. In: Chee, Y., Guo, Z., Ling, S., Shao, F., Tang, Y., Wang, H., Xing, C. (eds.) Coding and cryptology, volume 6639 of lecture notes in computer science, pp. 11–46. Springer, Berlin (2011)
3. Benaloh, J., Leichter, J.: Generalized secret sharing and monotone functions. In: Goldwasser, S. (ed.) Advances in cryptology — CRYPTO' 88, volume 403 of lecture notes in computer science, pp. 27–35. Springer, New York (1988)
4. Bertilsson, M., Ingemarsson, I.: A construction of practical secret sharing schemes using linear block codes. In: Seberry, J., Zheng, Y. (eds.) Advances in cryptology — AUSCRYPT '92, volume 718 of Lecture Notes in Computer Science, pp. 67–79. Springer, Berlin (1993)
5. Bishop, A., Pastro, V., Rajaraman, R., Wichs, D.: Essentially optimal robust secret sharing with maximal corruptions. In: Eurocrypt, pp. 58–86 (2016)
6. Blakley, G.R.: Safeguarding cryptographic keys. In: International workshop on managing requirements knowledge, volume 0, p. 313. IEEE Computer Society, Los Alamitos (1979)
7. Bracha, G.: An $o(logn)$ expected rounds randomized byzantine generals protocol. J. ACM **34**(4), 910–920 (1987)
8. Brickell, E.F.: Some ideal secret sharing schemes. In: Quisquater, J.-J., Vandewalle, J. (eds.) Advances in cryptology — EUROCRYPT '89, volume 434 of Lecture Notes in Computer Science, chapter 45, pp. 468–475. Springer, Berlin (1989)
9. Cabello, S., Padró, C., Sáez, G.: Secret sharing schemes with detection of cheaters for a general access structure. In: Ciobanu, G., Păun, G. (eds.) Fundamentals of Computation Theory, volume 1684, pp. 185–194. Springer, Berlin (1999)
10. Cabello, S., Padró, C., Germán, S.: Secret sharing schemes with detection of cheaters for a general access structure. Des. Codes Crypt. **25**(2), 175–188 (2002)
11. Carpentieri, M., De Santis, A., Vaccaro, U., Size of Shares and Probability of Cheating in Threshold Schemes. In: EUROCRYPT '93, v.olume.7.65., pp. 118–125. Springer (1993)
12. Lawrence Carter, J., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. **18**(2), 143–154 (1979)
13. Çamtepe, S.A., Yener, B., Yung, M.: Expander Graph based Key Distribution Mechanisms in Wireless Sensor Networks. In: 2006 IEEE International Conference on Communications, volume 5, pp. 2262–2267 (2006)

14. Cevallos, A., Fehr, S., Ostrovsky, R., Rabani, Y.: Unconditionally-Secure Robust Secret Sharing with Compact Shares. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT, vol. 7237, pp. 195–208. Springer, Berlin (2012)
15. Chandran, N., Garay, J.A., Ostrovsky, R.: Almost-Everywhere secure computation with edge corruptions. J. Cryptol. 1–24 (2013)
16. Chen, H., Cramer, R.: Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computations over Small Fields. In CRYPTO '06, pp. 521–536 (2006)
17. Cheraghchi, M.: Nearly Optimal Robust Secret Sharing. In: ISIT, pp. 2509–2513 (2016)
18. Cramer, R., Damgård, I., Döttling, N., Fehr, S., Spini, G.: Linear secret sharing schemes from error correcting codes and universal hash functions. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015, volume 9057 of Lecture Notes in Computer Science, pp. 313–336. Springer, Berlin (2015)
19. Cramer, R., Damgård, I., Fehr, S.: On the Cost of Reconstructing a Secret, or VSS with Optimal Reconstruction Phase. In: Kilian, J. (ed.) CRYPTO, volume 2139, pp. 503–523. Springer, Berlin (2001)
20. Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors. In: Smart, N.P. (ed.) EUROCRYPT, volume 4965 of Lecture Notes in Computer Science, pp. 471–488. Springer, Berlin (2008)
21. Cramer, R., Fehr, S.: Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups. In: Yung, M. (ed.) Advances in Cryptology — CRYPTO 2002, volume 2442 of Lecture Notes in Computer Science, chapter 18, pp. 272–287. Springer, Berlin (2002)
22. Damgård, I., Ishai, Y.: Scalable Secure Multiparty Computation. In: Dwork, C. (ed.) Advances in Cryptology - CRYPTO 2006, volume 4117 of Lecture Notes in Computer Science, pp. 501–520. Springer, Berlin (2006)
23. Damgård, I., Ishai, Y., Krøigaard, M.: Perfectly Secure Multiparty Computation and the Computational Overhead of Cryptography. In: Gilbert, H. (ed.) Advances in Cryptology – EUROCRYPT 2010, volume 6110 of Lecture Notes in Computer Science, pp. 445–465. Springer, Berlin (2010)
24. Damgård, I., Ishai, Y., Krøigaard, M., Nielsen, J.B., Smith, A.: Scalable Multiparty Computation with Nearly Optimal Work and Resilience. In: Wagner, D. (ed.) Advances in Cryptology – CRYPTO 2008, volume 5157 of Lecture Notes in Computer Science, pp. 241–261. Springer, Berlin (2008)
25. Damgård, I., Nielsen, J.B.: Scalable and Unconditionally Secure Multiparty Computation. In: Menezes, A. (ed.) Advances in Cryptology - CRYPTO 2007, volume 4622 of Lecture Notes in Computer Science, pp. 572–590. Springer, Berlin (2007)
26. den Boer, B.: A simple and key-economical unconditional authentication scheme. J. Comput. Secur. **2**(1), 65–71 (1993)
27. Dolev, D., Dwork, C., Waarts, O., Moti, Y.: Perfectly secure message transmission. J. ACM **40**(1), 17–47 (1993)
28. Fitzi, M., Franklin, M., Garay, J., Harsha Vardhan, S.: Towards optimal and efficient perfectly secure message transmission. In: Vadhan, S. (ed.) Theory of Cryptography, volume 4392 of Lecture Notes in Computer Science, pp. 311–322. Springer, Berlin (2007)
29. Franklin, M., Moti, Y.: Communication Complexity of Secure Computation (Extended Abstract). In: Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing, STOC '92, pp. 699–710, ACM, New York (1992)
30. Gál, A.: Combinatorial Methods in Boolean Function Complexity. PhD thesis, University of Chicago (1995)
31. Garay, J., Givens, C., Rafail, O.: Secure Message Transmission With Small Public Discussion. IEEE Trans. Inf. Theory **60**(4), 2373–2390 (April 2014)
32. Garay, J.A., Ostrovsky, R.: Almost-Everywhere Secure Computation. In: Smart, N. (ed.) Advances in Cryptology – EUROCRYPT 2008, volume 4965 of Lecture Notes in Computer Science, pp. 307–323. Springer, Berlin (2008)
33. Gennaro, R.: Theory and Practice of Verifiable Secret Sharing. PhD thesis, MIT (1996)
34. Ghosh, S.K.: On Optimality of Key Pre-distribution Schemes for Distributed Sensor Networks. In: Security and Privacy in Ad-Hoc and Sensor Networks: Third European Workshop, ESAS 2006, Hamburg, Germany, September 20-21, 2006, Revised Selected Papers, pp. 121–135. Springer, Berlin (2006)
35. Gilbert, E.N., MacWilliams, F.J., Sloane, N.JA.: Codes which detect deception. Bell Labs Technical J. **53**(3), 405–424 (1974)
36. Hirt, M., Maurer, U.: Player simulation and general adversary structures in perfect multiparty computation. J. Cryptol. **13**(1), 31–60 (2000)
37. Hoory, S., Linial, N., Wigderson, A.: Expander graphs and their applications. Bull. Am. Math. Soc. **43**(4), 439–561 (2006)

38. Ishai, Y., Ostrovsky, R., Seyalioglu, H.: Identifying Cheaters without an Honest Majority. In: Ronald Cramer, editor, Theory of Cryptography, volume 7194 of Lecture Notes in Computer Science, pp. 21–38. Springer, Berlin (2012)
39. Jhanwar, M.P., Safavi-Naini, R.: Unconditionally-Secure Robust Secret Sharing with Minimum Share Size. In: Sadeghi, A.-R. (ed.) Financial Cryptography, vol. 7859, pp. 96–110. Springer, Berlin (2013)
40. Johansson, T., Kabatianskii, G., Smeets, B.: On the relation between a-codes and codes correcting independent errors. In: Workshop on the Theory and Application of of Cryptographic Techniques, pp. 1–11. Springer (1993)
41. Kendall, M., Martin, K.M.: On the Role of Expander Graphs in Key Predistribution Schemes for Wireless Sensor Networks. In: Research in Cryptology: 4th Western European Workshop, WEWoRC 2011, Weimar, Germany, July 20-22, 2011, Revised Selected Papers, pp. 62–82, Springer, Berlin (2012)
42. Kurosawa, K.: General error decodable secret sharing scheme and its application. IEEE Trans. Inf. Theory **57**(9), 6304–6309 (2011)
43. Kurosawa, K., Kazuhiro, S.: Almost Secure (1-Round, n-Channel) Message Transmission Scheme. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E92-A**(1), 105–112 (2009)
44. Lewko, A.B., Pastro, V.: Robust Secret Sharing Schemes Against Local Adversaries Cryptology ePrint Archive: Report 2014/909 (2014)
45. Martí-Farré, J., Padró, C.: On Secret Sharing Schemes, Matroids and Polymatroids. In: Vadhan, S.P. (ed.) Theory of Cryptography, volume 4392 of Lecture Notes in Computer Science, pp. 273–290. Springer, Berlin (2007)
46. Martin, K.M., Paterson, M.B., Stinson, D.R.: Error decodable secret sharing and one-round perfectly secure message transmission for general adversary structures. Cryptogr. Commun. **3**(2), 65–86 (2011)
47. Ram Murty, M.: Ramanujan graphs. J.-Ramanujan Math. Soc. **18**(1), 33–52 (2003)
48. Ogata, W., Kurosawa, K.: Optimum Secret Sharing Scheme Secure against Cheating. In: Maurer, U. (ed.) EUROCRYPT, vol. 1070, pp. 200–211. Springer, Berlin (1996)
49. Ogata, W., Kurosawa, K., Stinson, D.R., Saido, H.: New combinatorial designs and their applications to authentication codes and secret sharing schemes. Discret. Math. **279**(1-3), 383–405 (2004)
50. Puder, D.: Expansion of random graphs: New proofs, new results. Invent. Math. 1–64 (2015)
51. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with hest majority. In: Proceedings of the twenty-first annual ACM symposium on Theory of computing, STOC '89, pp. 73–85. ACM, New York (1989)
52. Safavi-Naini, R., Wang, P.: A model for adversarial wiretap channels and its applications. J. Inf. Process. **23**(5), 554–561 (2015)
53. Shamir, A.: How to share a secret. Commun. ACM **22**, 612–613 (1979)
54. Simmons, G.J.: A survey of information authentication. Proc. IEEE **76**(5), 603–620 (1988)
55. Taylor, R.: An integrity check value algorithm for stream ciphers. In: Annual International Cryptology Conference, pp. 40–48. Springer (1993)
56. Tompa, M., Heather, W.: How to share a secret with cheaters. J. Cryptol. **1**(3), 133–138 (1989)
57. Van Dijk, M.: Secret key sharing and secret key generation. PhD thesis, Eindhoven University of Technology (1997)