Robust Secret Sharing Schemes Against Local Adversaries

Allison Bishop Lewko* and Valerio Pastro†

Columbia University

November 4, 2014

Abstract

We study robust secret sharing schemes in which between one third and one half of the players are corrupted. In this scenario, robust secret sharing is possible only with a share size larger than the secrets, and allowing a positive probability of reconstructing the wrong secret. In the standard model, it is known that at least $m+k$ bits per share are needed to robustly share a secret of bit-length m with an error probability of 2^{-k}; however, to the best of our knowledge, the efficient scheme that gets closest to this lower bound has share size $m + \widetilde{O}(n + k)$, where n is the number of players in the scheme.

We show that it is possible to obtain schemes with close to minimal share size in a model of local adversaries, i.e. in which corrupt players cannot communicate between receiving their respective honest shares and submitting corrupted shares to the reconstruction procedure, but may coordinate before the execution of the protocol and can also gather information afterwards. In this limited adversarial model, we prove a lower bound of roughly $m + k$ bits on the minimal share size, which is (somewhat surprisingly) similar to the lower bound in the standard model, where much stronger adversaries are allowed. We then present an efficient secret sharing scheme that essentially meets our lower bound, therefore improving upon the best known constructions in the standard model by removing a linear dependence on the number of players. For our construction, we introduce a novel procedure that compiles an error correcting code into a new randomized one, with the following two properties: a single local portion of a codeword leaks no information on the encoded message itself, and any set of portions of a codeword reconstructs the message with error probability exponentially low in the set size.

*alewko@cs.columbia.edu

†valerio@cs.columbia.edu

Contents

1 Introduction

While many cryptographic primitives require computational hardness assumptions to leverage restrictions on an adversary's computing power, the fundamental primitive of secret sharing protects data information-theoretically. This is accomplished by dispersing a secret among several parties, a sufficient number of whom are trustworthy. In a classical secret sharing scheme (as introduced independently by Shamir [Sha79] and Blakely [Bla79]), a dealer shares a secret among n parties such that any $t+1$ of them can reconstruct the secret, but any coalition of at most t players cannot learn anything about the secret. This is an information-theoretic guarantee, requiring that the joint distribution of any t shares must be independent of the secret.

Applications of secret sharing schemes range widely from secure multiparty computation (MPC), secure storage, secure message transmission, and distributed algorithms. In some of these applications, particularly secure storage and message transmission, an additional feature of "robustness" is desirable. Robust secret sharing is defined to satisfy all the usual properties of secret sharing, while additionally requiring that when the reconstruction procedure receives at most t adversarially corrupted shares out of n, it still outputs the correct secret (with sufficiently high probability).

Prior works on robust secret sharing (e.g. [RB89, CSV93, BS97, CDF01, CFOR12]) have focused on robustness against a "monolithic" adversary, i.e. a (computationally unbounded) centralized adversary who maliciously corrupts t parties and submits arbitrary values for their shares to the reconstruction procedure, potentially using all of the joint information present in the t shares initially received by the corrupted parties. In this model, it is known that for $t < n/3$ robust secret sharing schemes can be perfect, i.e. for any admissible adversary the reconstruction procedure outputs the correct secret with probability one (e.g. Shamir secret sharing, with Reed-Solomon decoding achieves this property). Interestingly, for $n/3 \leq t < n/2$ robust secret sharing is possible, but only by allowing a positive reconstruction failure probability [Cev11]. In this scenario, Cevallos et al. [CFOR12] presented a polynomial time robust secret sharing scheme over m-bit messages with share sizes of $m + \widetilde{O}(k+n)$ and reconstruction failure probability of 2^{-k}. This scheme has the lowest share size among efficient schemes in this model, but does not match the best known lower bound of $m+k$ [CSV93]. Our work is motivated by the following question:

Can the share size be significantly reduced with additional,
but reasonable, restrictions on the adversary?

We identify a very natural and realistic adversary for which we construct a scheme with considerably shorter shares – while still maintaining efficiency. In this new adversarial model, we also prove a lower bound of $m + k - 2 - \log_2(3)$ bits on the share size, which essentially matches our constructions' shares and is almost identical to the best known lower bound in the standard model, in which much stronger adversaries are allowed. By constructing a scheme that approximately attains our lower bound, we have a rather complete understanding of the share sizes that can be obtained for robust secret sharing schemes in this model, a degree of precision that has not yet been achieved against the standard monolithic adversary.

Our Adversarial Model. We consider a "local" adversary, meaning that the t corrupted players cannot communicate with each other during the execution of the protocol – but they may arbitrarily coordinate before and after (the latter to try to gain knowledge on the secret). This means that each of the corrupted parties must decide on his malicious share to submit to the reconstruction procedure based only on some pre-determined strategy and the one honest share it has received by the dealer. This model carries some similarities to the work of Lepinski et al. [LMs05], in the context of collusion-free protocols. In the setting of secret sharing robust against local adversaries, it is still true that for $t < n/3$ schemes can be perfect, and for $n/3 \leq t < n/2$ robustness can be achieved only allowing a failure probability (the same proofs as the ones in the monolithic adversarial model still apply), but in this latter scenario, working with local adversaries allows us to construct schemes with optimal share sizes, still maintaining efficiency.

Motivation. Local adversaries can potentially model several kinds of realistic limitations of adversarial power in many applications. For example, in a secure message transmission, data may travel quickly and realtime cooperation among corrupted nodes may be unlikely. In a large secure multiparty computation, the scale and pace of the computation may also make online coordination among adversarial parties unrealistic. Corrupted parties may also be mutually distrusting, unwilling to coordinate (e.g. if they have opposite goals), or they might not even know about the existence of each other (say in a large scale MPC over the Internet).

Similar adversary models have been well-studied in other subfields of computer science, such as the multi-prover setting for interactive proofs. In the classical result of $IP = PSPACE$ [Sha92], a single, computationally unbounded and potentially duplicitous prover must convince a much less powerful verifier of the truth of a particular statement. As was shown in [BFL91], considering two duplicitous but non-communicating provers greatly expands the class of statements that can be proved, as $MIP = NEXP$. Removing online communication between the provers is precisely what fuels this expanded power, and similar gains may be possible in other interactive scenarios, including secure multiparty computation and robust distributed algorithms.

More Details on Our Results. As mentioned earlier, we prove two complementary results on the share size of secret sharing schemes robust against a local adversary corrupting t of the n players, where $n/3 \leq t < n/2$, and where the reconstruction failure probability is 2^{-k}.

In the first part of the paper, we show a lower bound of $m+k-2-\log_2(3)$ on the minimal share size in this setting. This is somewhat surprising, since it is quantitatively comparable to the lower bound of $m + k$ proven in [CSV93] in the case of a monolithic (and much stronger!) adversary. Our proof uses remarkably little adversarial power to obtain this lower bound: more precisely, we show that this lower bound holds against an oblivious adversary who completely ignores the honest shares given to corrupted parties and replaces them with either default values or fresh shares.

In the second part, we construct a poly-time scheme robust against local adversaries whose share size is $m + O(k)$, which essentially meets our lower bound. Our core idea for shrinking the shares is to authenticate all honest shares with a single MAC key that is "hidden in plain sight" from a local adversary. To do so, while still ensuring that the key can be efficiently recovered by the reconstruction procedure, we develop a novel tool integrating error-correcting codes with "locally hiding" distributions, a rather general tool that may be of independent interest.

Compared to the scheme in the standard model with smallest share size [CFOR12], our scheme reduces the share size by removing the additive factor of n. Thus, we see that restricting to local adversaries allows us to considerably reduce share size down to approximately match a proven lower bound, removing any linear dependence on the number of players, while maintaining polynomial time efficiency. This yields a much tighter understanding of what is achievable against local adversaries than what is known against a monolithic adversary in the context of robust secret sharing.

Techniques for our Construction. Previous constructions of robust secret sharing schemes use MACs to authenticate honest shares. Against a monolithic adversary who can view all of the shares received by corrupt players, it seems necessary to use many different MACs to prevent the adversary from compiling enough information about the keys to forge enough tags for corrupt shares. These many MAC keys and tags significantly increase the size of shares.

In the local adversary setting where each corrupt party can only act based upon a pre-determined strategy and its own received share, we can restrict to a single MAC key to be used on each share for authentication. Essentially, we will design our shares so that each party will be given a share that is distributed independently of the MAC key when considered on its own, but the joint distribution of just a constant number of honest shares reveals the key (hence allowing authentication of honest shares).

The basic idea is as follows: each share consists of a Shamir share of the secret, a tag on the Shamir share, and information on the global MAC key (used for the tag). This information has to be conveyed in a way that a single player obtains no information on the key itself (otherwise it could forge its tag), and the key is still retrievable even if nearly half of the shares are corrupt.

In our construction, the dealer embeds the key in a bit-matrix and distributes one row per player in such a way that each single row looks random, but the joint distribution of enough rows reveals the key. More specifically, each bit of the key is encoded as a column of such matrix, as follows: the bit 0 is encoded as a uniform bit-column, while the bit 1 is encoded as either the all-zero and the all-one column, and this choice is uniform. A single row in such matrix is a uniform string; no information on the key is revealed. On the other hand, looking across a bigger number of honest rows (and seeing them all agree at the positions corresponding to 1) allows us to invert the embedding with probability close to one – the failure probability decreases exponentially with the number of honest rows seen. In order to make the failure probability negligible when the number of inspected rows is constant, we encode the key via an asymptotically good error correcting code before the embedding procedure.

A secondary challenge is that looking at corrupt rows can lead to the wrong key. However, it is possible to detect a corrupt key by the fact that it verifies fewer than $t + 1$ tags with high probability (the honest shares are likely to be incompatible with a non-honest key).

Thus, we can iterate the procedure to invert the embedding of the key through all subsets of shares of a fixed constant size, attempt to reconstruct the MAC key from each set, and stop whenever we find one that authenticates properly. This computation is still polynomial in n and succeeds with sufficiently high probability. This comprises our construction of an efficient secret sharing scheme that is robust against local adversaries, with a significantly reduced share size compared to previous constructions in the standard model.

Techniques for the Lower Bound. To prove our lower bound on minimal share size in this setting, we consider very simple local adversary strategies. We suppose that a local adversary's goal is to cause a reconstruction failure when a challenger generates honest shares from a uniformly random secret. In particular, the adversary identifies a player with a share of minimal length and chooses to corrupt a random set of t of the remaining players and replaces the corrupt players' shares with freshly generated honest shares for a new uniformly chosen secret. Note that these t corrupted shares will be sampled from the same distribution as honest shares, but sampled independently from the true secret. For simplicity of illustration, suppose that this local adversary has replaced the first t shares with its own sample, while the remaining $t+1$ shares are honest. Also suppose that the $t+1$st share has minimal length (any scenario follows these assumptions, up to a relabeling of the players indices). Then, it is likely that the first t corrupted shares and the honest $t+1$st share are also consistent with some honest sharing. This will cause collisions, resulting in a reconstruction error probability that is at least exponential in the overhead of the $t+1$st share (i.e. in the bit length of the share minus the bit length of the secret).

To formally model this, we parameterize the underlying probability space in terms of pairs of secrets and random strings chosen by the share generating algorithm. We group these pairs into various equivalence classes based on collisions of subsets of the resulting shares, and model these equivalences in a layered graph. Our analysis takes advantage of the fact that the adversary can produce the first t corrupted shares in a way that is consistent with the $t+1$st share without knowing what the reconstruction would output. This crucial property comes as a consequence of the privacy guarantee of the scheme: any first t shares are consistent with every secret, otherwise the adversary would get information on the secret after the protocol is over (and communication between corrupt players is allowed).

Additional Related Work. Decentralized adversaries are also considered in [AKL$^+$09, CV12], which provide frameworks for simulation-based security definitions for cryptographic primitives against local adversaries. Similarly, in the setting of leakage-resilient cryptography, various "local" adversarial models have been studied. For example, the "only computation leaks information" axiom of Micali and Reyzin [MR04] restricts an adversary to leakage that happens solely on whatever portion of a secret state is currently involved in a computation. Some other works, such as [DP08] and [DLWW11] consider secret state as divided among multiple devices and leaking independently. [AKMZ12] also present a rather general study of various collusion restrictions on adversarial actors in multiparty protocols.

2 Preliminaries

In this section we list the classic tools and notation used in our paper.

We usually denote distributions by calligraphic letters (e.g. $\mathcal{D}$), random variables by capital letters (e.g. $D \sim \mathcal{D}$ reads as "D follows the distribution $\mathcal{D}$"), and samples by lowercase letters (e.g. $d \leftarrow D$ reads "d is sampled according to D"). Moreover, for any set X, we denote by $\mathcal{U}_X$ the uniform distribution on X.

Definition 2.1 (Projection)**.** For any integer n, for any set $X = X_1 \times \cdots \times X_n$, and for any $I \subseteq \{1, \ldots, n\}$, we write X_I to denote the set $\prod_{i \in I} X_i$. This notation is carried over to the elements of X.

Definition 2.2 (Hamming Weight)**.** For a vector $v \in \mathbb{F}_2^c$, we define $w(v)$ to be the Hamming weight of v (i.e. the number of non-zero coordinates of v).

We will use the following Chernoff Bound, which appears as Theorem 4.4 in [MU05].

Lemma 2.3. *Let $Y_1, \ldots, Y_m$ be independent random variables with* $\Pr[Y_i = 1] = p$ *and* $\Pr[Y_i = 0] = 1 - p$*. Let $Y = \sum_{i=1}^m Y_i$ and $\mu = p \cdot m$. Then for $0 < \beta \leq 1$,*

$$\Pr[Y \geq (1 + \beta) \cdot \mu] \leq e^{-\mu\beta^2/3}.$$

2.1 Message Authentication Codes

Definition 2.4 (MAC)**.** A (one time) ε-secure *message authentication code* (MAC) for messages in $\mathcal{M}$ is a function $\mathsf{MAC} : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$, for some sets $\mathcal{K}$ (key space) and $\mathcal{T}$ (tag space) such that for all $m \neq m' \in \mathcal{M}$, for all $t, t' \in \mathcal{T}$, and for a uniform random variable $K \sim \mathcal{U}_{\mathcal{K}}$:

$$\Pr[\mathsf{MAC}(K, m') = t' \mid \mathsf{MAC}(K, m) = t] \leq \varepsilon.$$

2.2 Error-Correcting Codes

An error-correcting code for messages that are bit strings of length h is a function $C : \mathbb{F}_2^h \to \mathbb{F}_2^c$, where c is called the block length. The distance d of the code is defined as

$$\min_{x \neq y \in \mathbb{F}_2^h} \{w(x - y)\}.$$

The number E of adversarial errors tolerated is $\lceil \frac{d}{2} - 1 \rceil$, while the fraction e of errors tolerated is $\frac{E}{c}$. The rate of the code r is defined to be h/c. A decoding procedure is a function $D : \mathbb{F}_2^c \to \mathbb{F}_2^h$ such that whenever z satisfies $w(z, C(x)) \leq E$, $D(z) = x$.

An infinite ensemble of codes for increasing block lengths c is said to be *asymptotically good* if the rate r and fraction of errors e are both lower bounded by positive constants. Such codes are known to exist, and with efficient encoding and decoding functions. For example, Justesen [Jus72] gave an explicit family of asymptotically good codes with block lengths $h = 2m(2^m - 1)$ for each positive integer m with efficient encoding and decoding functions.

2.3 Robust Secret Sharing Schemes

Throughout the rest of the paper, we use the following notation:

- n is an integer that denotes the number of players in the scheme.
- $t \leq n$ denotes the maximum number of corruptible players in the scheme.
- $\mathcal{M}$ is the message space. We denote by m the integer such that $2^{m-1} < |\mathcal{M}| \leq 2^m$.
- $\mathcal{R}$ is a set that denotes the randomness space used by the scheme to share messages. We assume that the scheme samples uniform elements in $\mathcal{R}$ to produce sequences of shares.
- $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_n$ is a set that denotes the ambient space of sequences of shares. For $i = 1, \ldots, n$ we denote by 0_i a default element in $\mathcal{S}_i$ (i.e. an element that any Turing machine can retrieve without any input). For example, if $\mathcal{S}_i$ is a group, 0_i could be the zero of $\mathcal{S}_i$ as a group.

Definition 2.5 (Secret Sharing Scheme)**.** A *t-private, n-player secret sharing scheme* over a message space $\mathcal{M}$ is a tuple $(\mathsf{Share}, \mathsf{Rec})$ of algorithms that run as follows:

$\mathsf{Share}(s, r) \to (s_1, \ldots, s_n)$**:** this algorithm takes as input a message $s \in \mathcal{M}$ and randomness $r \in \mathcal{R}$ and outputs a sequence of shares $(s_1, \ldots, s_n) \in \mathcal{S}$.

$\mathsf{Rec}(s_1, \ldots, s_n) \to s'$**:** this algorithm takes as input an element $(s_1, \ldots, s_n) \in \mathcal{S}$ (not necessarily output by Share) and outputs a message $s' \in \mathcal{M}$.

Moreover, the following properties hold:

Privacy: Any t out of n shares of a secret give no information on the secret itself. More formally, for any random variable S over $\mathcal{M}$ and uniform $R \sim \mathcal{U}_{\mathcal{R}}$:

$$S = (S \mid \mathsf{Share}(S,R)_{C_1} = \mathsf{Share}(s,r)_{C_1}, \ldots, \mathsf{Share}(S,R)_{C_t} = \mathsf{Share}(s,r)_{C_t})$$

Correctness: Reconstructing a sequence of shares generated by the sharing procedure leads to the original secret. More formally, for any $s \in \mathcal{M}$, $r \leftarrow \mathcal{U}_{\mathcal{R}}$:

$$\Pr[\mathsf{Rec}(\mathsf{Share}(s,r)) = s] = 1$$

Definition 2.6 (Merging Function)**.** Let $s \in \mathcal{M}, r \in \mathcal{R}$ and let $I \subseteq \{1, \ldots, n\}$. For $i \in I$, let $v_i \in \mathcal{S}_i$. We define the *merging function* of s, r with $I, (v_i)_{i \in I}$ as

$$\mathsf{Merge}(s, r, I, (v_i)_{i \in I}) = S \in \mathcal{S}$$

where for $i \in I$ $S_i = v_i$, and for $i \notin I$ $S_i = \mathsf{Share}(s,r)_i$.

Definition 2.7 (Adversary)**.** For any t-private, n-player secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$, we define the experiment $\mathbf{Exp}_{(\mathsf{Share},\mathsf{Rec})}(\mathcal{D}, \mathsf{Adv})$, where $\mathcal{D}$ is a distribution over $\mathcal{M}$, and Adv is an interactive Turing machine, called *the adversary*.

$\mathbf{Exp}_{(\mathsf{Share},\mathsf{Rec})}(\mathcal{D}, \mathsf{Adv})$ is defined as follows:

E.1. Send the public description $(\mathsf{Share}, \mathsf{Rec})$ of the scheme and the distribution $\mathcal{D}$ to Adv.

E.2. Adv computes and outputs $I = \{i_1, \ldots, i_t\} \subseteq \{1, \ldots, n\}$, i.e. a subset of players whose size is less or equal to t.

E.3. Sample $s \leftarrow \mathcal{D}$, and $r \leftarrow \mathcal{U}_{\mathcal{R}}$, compute $\mathsf{Share}(s,r)$ and send $\mathsf{Share}(s,r)_I$ to Adv.

E.4. Adv outputs $(v_i)_{i \in I}$, where $v_i \leftarrow V_i$ and

$$V_i = V_i(\mathsf{Share}, \mathsf{Rec}, \mathcal{D}, \mathsf{Share}(s,r)_{i_1}, \ldots, \mathsf{Share}(s,r)_{i_t})$$

is a random variable that may depend on the public information of the scheme, and the ensemble of shares indexed by I.

E.5. Return 1 if and only if $\mathsf{Merge}(s, r, I, (v_i)_{i \in I}) \neq s$.

For $v \leq t$, we say that an adversary is *v-local* if for all $i \in I$,

$$V_i = V_i(\mathsf{Share}, \mathsf{Rec}, \mathcal{D}, \mathsf{Share}(s,r)_{i_1}, \ldots, \mathsf{Share}(s,r)_{i_v}),$$

i.e. V_i is a random variable that depends only on the public information of the scheme, and at most v elements of the ensemble of shares indexed by I.

Definition 2.8 (Robust Secret Sharing Scheme)**.** A t-private n-player secret sharing scheme $(\mathsf{Share}, \mathsf{Rec})$ over a message space $\mathcal{M}$ is *(t, δ)-robust* if the following property holds:

Robustness: With probability less or equal to δ the reconstruction procedure fails at outputing the correct shared value, even if t out of the n shares are corrupt by adversary. Formally, for any distribution $\mathcal{D}$, for any adversary Adv:

$$\Pr[\mathbf{Exp}_{(\mathsf{Share},\mathsf{Rec})}(\mathcal{D}, \mathsf{Adv}) = 1] \leq \delta$$

We say that a scheme is *(t, δ)-robust agaist v-local adversaries* if robustness holds for any v-local adversary.

3 Lower Bound

We prove a lower bound for the share size of any secret sharing scheme that is robust against 0-local adversaries, which implies that this lower bound applies to any secret sharing scheme that is robust against v-local adversaries for any $v \geq 0$.

Theorem 3.1. *Let k, m, t be integers. Let $\delta = 2^{-k}$, $n = 2 \cdot t + 1$, $\mathcal{M}$ be a set with $2^{m-1} \leq |\mathcal{M}| \leq 2^m$. Let $(\mathsf{Share}, \mathsf{Rec})$ be an n-player secret sharing scheme over $\mathcal{M}$. If $(\mathsf{Share}, \mathsf{Rec})$ is (t, δ)-robust, then the minimum bit-length of any of its shares is at least $m + k - (2 + \log_2(3))$.*

Proof. We first note that it suffices to prove a lower bound of $m + k - (1 + \log_2(3))$ for $|\mathcal{M}| = 2^m$, since a lower bound for the share size required to share a secret from a space of size 2^{m-1} certainly applies to sharing a secret from larger a space of size $|\mathcal{M}| \geq 2^{m-1}$. Throughout the proof, we will therefore assume that $|\mathcal{M}| = 2^m$.

Our proof will rely solely on very simple local adversary strategies. Namely, we will need to consider only two possible adversary strategies: one that replaces some subset of t shares with default values of all zeros, and another that replaces them with shares generated with fresh randomness for a fresh (uniform) secret. These strategies are both 0-local because the adversary submits shares that are distributed independently of all the shares that the corrupted players receive. The key idea will be that if one share is very short, then it becomes more likely that the adversary submitting t freshly distributed shares will cause a "collision", meaning that the corrupted shares are consistent with the honestly generated short share. This will make it difficult for the reconstruction algorithm to tell which is the honestly shared secret. We also consider the adversary who submits default values for technical reasons within the argument, in order to prove that there are not too many honest sharings for differing secrets that agree in some set of at least $t + 1$ shares. If these were too common, the adversary submitting default values for the complement set would succeed in confusing the reconstruction algorithm with sufficient probability.

To carefully study the probability space of pairs (s, r) where s is a uniformly random secret and r is a random bit string used in the share generating procedure, we define a layered graph whose vertices at each layer correspond to these pairs (s, r), and edges between the layers represent agreeing shares for specified subsets of players. Essentially, our graph models various kinds of equivalence classes of values (s, r) corresponding to partial agreements of the resulting shares. To execute our proof, we will identify paths in our graph corresponding to the events of adversary success, and we will then lower bound the number of such edges and hence the success probability of the adversary.

A Graph. Let $P \in \{1, \ldots, n\}$ be the index of a player, let $I \subset \{1, \ldots, n\} \setminus \{P\}$ be a set of cardinality $|I| = t$, and let $J = \{1, \ldots, n\} \setminus (\{P\} \cup I)$ be the set of size t corresponding to the players that are not in I and are not P. Let $G = G(P, I)$ be a graph defined as:

- $\mathsf{Vertices}(G) = \{1, \ldots, 4\} \times \mathcal{M} \times \mathcal{R}$, i.e. the vertex set consists of four layers of message and random value tuples.
- $((i, s, r), (i + 1, s', r')) \in \mathsf{Edges}(G)$ if:
  - $i = 1$, and $\mathsf{Share}(s, r)_I = \mathsf{Share}(s', r')_I$: i.e. a vertex at layer one is connected to a vertex at layer two if the tuples of shares they define agree on the shares at I.

– $i = 2$, $s \neq s'$, and $\mathsf{Share}(s,r)_P = \mathsf{Share}(s',r')_P$: i.e. a vertex at layer two is connected to a vertex at layer three if the vertices represent different secrets, and the tuples of shares they define agree on the share at P.

– $i = 3$, and $\mathsf{Share}(s,r)_J = \mathsf{Share}(s',r')_J$: i.e. a vertex at layer three is connected to a vertex at layer four if the tuples of shares they define agree on the shares at J.

Path Sets, Labeling, Balance. We want to construct a labeling system for paths from layer one to layer four, that will be useful to analyze certain reconstruction properties of the secret sharing scheme associated with the graph. Firstly, however, we need to construct a function that maps paths containing edges from layer two to layer three to sequences of shares. For $1 \leq i < j \leq 4$, let $\mathcal{E}_{i,j}$ be the set of paths succesively connecting vertices at layer i to vertices at layer j; formally,

$$\mathcal{E}_{i,j} := \{((i,s_i,r_i),(i+1,s_{i+1},r_{i+1}),\ldots,(j,s_j,r_j)) \mid \\ \text{for } i \leq k < j : ((k,s_k,r_k),(k+1,s_{k+1},r_{k+1})) \in \mathsf{Edges}(G(P,I))\}.$$

We also define another set, $\mathcal{E}$, containing all paths with an edge between layer two and three; formally,

$$\mathcal{E} = \bigcup_{i\in\{1,2\},j\in\{3,4\}} \mathcal{E}_{i,j}.$$

Now, we construct a *string function* S that assigns sequences of shares to paths in $\mathcal{E}$. Formally, for $\ell \in \mathcal{E}$, $\ell = (\ldots,(2,s_2,r_2),(3,s_3,r_3),\ldots)$, define $S(\ell)$ as the sequence of shares with the following properties:

- $S(\ell)_I := \mathsf{Share}(s_2,r_2)_I$,
- $S(\ell)_P := \mathsf{Share}(s_2,r_2)_P = \mathsf{Share}(s_3,r_3)_P$,
- $S(\ell)_J := \mathsf{Share}(s_3,r_3)_J$.

Notice that the function S depends only on the edges between layer two and three, so any two paths in $\mathcal{E}$ sharing the same edge from layer two and three have the same image.

Now, for $i \in \{1,2\}, j \in \{3,4\}$, we define a *labeling relation* L as follows:

$$\begin{aligned} L : \mathcal{E} &\longrightarrow \{I,J\} \\ \ell &\longmapsto \begin{cases} I, & \text{if } \mathsf{Rec}(S(\ell)) \neq s_3, \\ J, & \text{if } \mathsf{Rec}(S(\ell)) \neq s_2 \end{cases} \end{aligned}$$

Analogously to S, L depends only on the edges between layer two and three of a path. Also notice that L is not necessarily a function, as we do not exclude the existence of paths $\ell = (\ldots,(2,s_2,r_2),(3,s_3,r_3),\ldots)$ with $s_2 \neq \mathsf{Rec}(S(\ell)) \neq s_3$. Such paths would be labeled as both I and J.

Finally, we say that the graph G is I*-oriented* if there are at least as many edges in $\mathcal{E}_{2,3}$ labeled by I than J, i.e. if $|\{\ell \in \mathcal{E}_{2,3} \mid L(\ell) = I\}| \geq |\{\ell \in \mathcal{E}_{2,3} \mid L(\ell) = J\}|$.

Setup. Let λ be the minimal bit-length of any share of $(\mathsf{Share},\mathsf{Rec})$. Without loss of generality, assume that P is a player associated with a share of $(\mathsf{Share},\mathsf{Rec})$ of length λ.

Construction of an Adversary. Let Adv_A be the adversary who does the following during $\mathbf{Exp}_{(\mathsf{Share},\mathsf{Rec})}(\mathcal{D},\mathsf{Adv}_A)$:

1. Given the public information (Share, Rec), $\mathcal{D}$ in step E.1, sample $x \leftarrow \mathcal{U}_{\mathcal{M}}$, $r_x \leftarrow \mathcal{U}_{\mathcal{R}}$.

2. Compute $(v_1, \ldots, v_n) \leftarrow \mathsf{Share}(x, r_x)$.

3. Sample a uniform set $I \subset \{1, \ldots, n\} \setminus \{P\}$ with $|I| = t$.

4. Construct $G(P, I)$.

5. If $G(P, I)$ is I-oriented, output I at step E.2, and $(v_i)_{i \in I}$ at step E.4.
   Else, output J at step E.2, and $(v_j)_{j \in J}$ at step E.4.

Notice that Adv_A is a valid 0-local adversary, since all the computation Adv_A performs is independent of the values it is inputed at step E.3.

Representing Adversarial Success in the Graph. Assume that, if I is the set chosen by the adversary, the graph $G(P, I)$, induced by the given secret sharing scheme, is I-oriented. Let $z \in \mathcal{M}$, $r_z \leftarrow \mathcal{U}_{\mathcal{R}}$, let C be a sequence of shares defined as: for $i \in I$, $C_i = V_i = \mathsf{Share}(x, r_x)_i$; for $j \in J \cup \{P\}$, $C_j = \mathsf{Share}(z, r_z)_j$. Notice that C can be seen as a sharing of z corrupted at I by the above adversary, therefore, by the robustness property:

$$\Pr[\mathsf{Rec}(C) \neq z] \leq \delta = 2^{-k}, \tag{3.1}$$

where the probability is taken over uniform choices of $x, z \in \mathcal{M}$, $r_x, r_z \in \mathcal{R}$. Notice that if there exists (y, r_y) such that $\ell := ((1, x, r_x), (2, y, r_y), (3, z, r_z)) \in \mathcal{E}_{1,3}$ and $V(\ell) = I$ then $\mathsf{Rec}(C) \neq z$: in fact, if $\ell \in \mathcal{E}_{1,3}$, then $V(\ell) = I$ implies $\mathsf{Rec}(S(\ell)) \neq z$, by definition of V; and since $S(\ell) = C$ (by the following: $S(\ell)_I = \mathsf{Share}(y, r_y)_I = \mathsf{Share}(x, r_x)_I = C_I$, $S(\ell)_J = \mathsf{Share}(z, r_z)_J = C_J$ and $S(\ell)_P = \mathsf{Share}(y, r_y)_P = \mathsf{Share}(z, r_z)_P = C_P$) then $V(\ell) = I$ implies $\mathsf{Rec}(C) \neq z$. This means that

$$\Pr[\exists (y, r_y), \ell := ((1, x, r_x), (2, y, r_y), (3, z, r_z)) \in \mathcal{E}_{1,3}, V(\ell) = I] \leq \Pr[\mathsf{Rec}(C) \neq z], \tag{3.2}$$

which implies

$$\Pr[\exists (y, r_y), \ell := ((1, x, r_x), (2, y, r_y), (3, z, r_z)) \in \mathcal{E}_{1,3}, V(\ell) = I] \leq 2^{-k}, \tag{3.3}$$

by combining equation 3.1 and 3.2.

A More Refined Graph. In order to better analyze the left-hand side of equation 3.3, we introduce a subgraph $G'(P, I)$ of $G(P, I)$, defined by the following algorithm:

1. Initialize $G' \leftarrow G(P, I)$

2. For $a = (a_{i_1}, \ldots, a_{i_{t+1}}) \in \mathcal{S}_{I \cup \{P\}}$:
   (a) Define $H_a := \{(2, s, r) \in \mathsf{Vertices}(G) \mid \mathsf{Share}(s, r)_{I \cup \{P\}} = a\}$
   (b) Initialize $H'_a := H_a$
   (c) While there exist $(2, s, r), (2, s', r') \in H'_a$ such that $s \neq s'$:
      i. Update the graph G' by removing $(2, s, r)$ and $(2, s', r')$:
         - $\mathsf{Edges}(G') \leftarrow \mathsf{Edges}(G')_{|\mathsf{Vertices}(G') \setminus \{(2,s,r),(2,s',r')\}}$
         - $\mathsf{Vertices}(G') \leftarrow \mathsf{Vertices}(G') \setminus \{(2, s, r), (2, s', r')\}$
      ii. Update $H'_a \leftarrow \{(2, s, r) \in \mathsf{Vertices}(G') \mid \mathsf{Share}(s, r)_{I \cup \{P\}} = a\}$

3. Output $G'(P,I) \leftarrow G'$.

In the following, we bound the number $V_R = |\mathsf{Vertices}(G(P,I)) \setminus \mathsf{Vertices}(G'(P,I))|$ of vertices removed from $G(P,I)$ by the above algorithm to obtain $G'(P,I)$. To do so, we relate V_R to $\Pr[\mathbf{Exp}_{(\mathsf{Share},\mathsf{Rec})}(\mathcal{U}_\mathcal{M}, \mathsf{Adv}_B) = 1]$ where Adv_B is a specific adversary, defined as follows:

1. Let $b = (0_{j_1}, \ldots, 0_{j_t}) \in \mathcal{S}_J$
2. Output J at step E.2, b at step E.4.

Notice that Adv_B is a valid 0-local adversary, as b depends only on the public specifications $(\mathsf{Share}, \mathsf{Rec})$ of the scheme (and therefore it is independent of any value inputed to B at step E.3). Let

$$G_B := \{(s,r) \in \mathcal{M} \times \mathcal{R} \mid \mathsf{Rec}(\mathsf{Merge}(s,r,J,b)) \neq s\}$$

Notice that if any element (s,r) of G_B is sampled at step E.3 of $\mathbf{Exp}_{(\mathsf{Share},\mathsf{Rec})}(\mathcal{U}_\mathcal{M}, \mathsf{Adv}_B)$, then $\mathbf{Exp}_{(\mathsf{Share},\mathsf{Rec})}(\mathcal{U}_\mathcal{M}, \mathsf{Adv}_B)$ outputs 1, by definition of G_B. Notice also that the probability of sampling (s,r) in G_B at step E.3 is $|G_B|/|\mathcal{M} \times \mathcal{R}|$, as the experiment considers uniform messages (and randomness). Therefore, by the robustness of the scheme,

$$|G_B|/|\mathcal{M} \times \mathcal{R}| \leq 2^{-k} \tag{3.4}$$

Now, we want to relate G_B and V_R. Notice that any two vertices $(2,s,r)$, $(2,s',r')$, simultaneously removed in step 2(c)i, belong to the same set H_a for some a, which implies that

$$\mathsf{Share}(s,r)_{I \cup \{P\}} = a = \mathsf{Share}(s',r')_{I \cup \{P\}}, \tag{3.5}$$

by definition of H_a. Combining equation 3.5 with the fact that $\{1,\ldots,n\} \setminus J = I \cup \{P\}$, it follows that $\mathsf{Merge}(s,r,J,b) = S = \mathsf{Merge}(s',r',J,b)$. Now, let $s'' \leftarrow \mathsf{Rec}(S)$. Since $s \neq s'$ then at least one between s and s' differs from s'', which means that at least one between (s,r) and (s',r') lies in G_B. Therefore,

$$V_R \leq 2 \cdot |G_B| \tag{3.6}$$

In other words, at least half of the vertices $(2,s,r)$ removed in the construction of G' are such that to $(s,r) \in G_B$. Combining equation 3.6 with equation 3.4, we get

$$V_R \leq 2 \cdot 2^{-k} \cdot |\mathcal{M} \times \mathcal{R}| \tag{3.7}$$

General Facts about the Connectivity between Layers. Now that we have a bound on the number of vertices removed from $G(P,I)$ to obtain $G'(P,I)$ we can proceed and study how some specific sets of vertices are connected between the layers of $G'(P,I)$. We are mostly interested in vertices on layer one and two. For any vertex $(2,s,r) \in \mathsf{Vertices}(G'(P,I))$, and for any secret $s' \in \mathcal{M}$, define

$$C_{s'}(2,s,r) := \{(1,s',r') \mid ((1,s',r'),(2,s,r)) \in \mathsf{Edges}(G'(P,I))\}$$

i.e. the set of vertices at layer one that represent secret s' and are connected to $(2,s,r)$. Notice that the set $\{C_{s'}(2,s,r)\}_{s' \in \mathcal{M}}$ is a partition of the set of vertices at layer one connected to $(2,s,r)$. We want to show that for any s', s'', $|C_{s'}(2,s,r)| = |C_{s''}(2,s,r)|$. For the sake of contradiction, assume this is not the case, so without loss of generality there exist $s' \neq s''$ such that $|C_{s'}(2,s,r)| > |C_{s''}(2,s,r)|$. By definition of $G'(P,I)$, this means that

$$|\{r' \in \mathcal{R} \mid \mathsf{Share}(s',r')_I = \mathsf{Share}(s,r)_I\}| > |\{r'' \in \mathcal{R} \mid \mathsf{Share}(s'',r'')_I = \mathsf{Share}(s,r)_I\}|$$

which implies that

$$\Pr[s' \mid \mathsf{Share}(s,r)_I] > \Pr[s'' \mid \mathsf{Share}(s,r)_I]$$

and therefore violates the privacy of the scheme, as $\mathsf{Share}(s,r)_I$ would reveal that the secret is more likely to be s' than s'', but by privacy given any t shares the secret should look uniform. Therefore,

$$\text{for any } s', s'' \in \mathcal{M},\ (2,s,r) \in G'(P,I)\text{: } |C_{s'}(2,s,r)| = |C_{s''}(2,s,r)| \tag{3.8}$$

This implies that any $(2,s,r) \in G'(P,I)$ is connected to $2^n \cdot |C_s(2,s,r)|$ vertices at layer one ($2^n \cdot |C_s(2,s,r)| = |\cup_{s' \in \mathcal{S}} C_{s'}(2,s,r)|$, by the fact that $\{C_{s'}(2,s,r)\}_{s' \in \mathcal{M}}$ is a partition).

Particular Facts about the Connectivity between Layers. Now, with a notation similar to the one in the construction of $G'(P,I)$, for $a \in \mathcal{S}_{I \cup \{P\}}$, let

$$H'_a := \{(2,s,r) \in \mathsf{Vertices}(G'(P,I)) \mid \mathsf{Share}(s,r)_{I \cup \{P\}} = a\}$$

Moreover, let

$$C'_a := \{(1,s,r) \in \mathsf{Vertices}(G'(P,I)) \mid \exists (2,s',r') \in H'_a : ((1,s,r),(2,s',r')) \in \mathsf{Edges}(G'(P,I))\}$$

i.e. the set of vertices at layer one that are connected to H'_a. Notice that all vertices in H'_a represent the same secret: namely, if $(2,s,r),(2,s',r') \in H'_a$, then $s = s'$, by construction of $G'(P,I)$. Also, for any $(2,s,r) \in H'_a$, if $(2,s,r') \in H'_a$, then $((1,s,r'),(2,s,r)) \in \mathsf{Edges}(G'(P,I))$, again by construction of H'_a, and in particular from the fact that $\mathsf{Share}(s,r)_I = \mathsf{Share}(s,r')_I$. This implies that for any $(2,s,r) \in H'_a$, $|C_s(2,s,r)| \geq |H'_a|$. Using property 3.8, we get that any $(2,s,r) \in H'_a$ is connected to a set X of vertices at layer one of cardinality at least $2^m \cdot |H'_a|$. Since $|C'_a| \geq |X|$ (as $C'_a \supseteq X$), we get Therefore,

$$|C'_a| \geq 2^m \cdot |H'_a| \tag{3.9}$$

Putting things together. We can now proceed and bound the left-hand side of equation 3.3 in terms of the size of $\mathcal{S}_P$. The following calculation starts with a probability space where (x, r_x) and (z, r_z) are independently and uniformly sampled form $\mathcal{M} \times \mathcal{R}$. We begin with some simple consequences of our definitions:

$$\begin{aligned}
2^{-k} &\geq \Pr[\exists (y,r_y), \ell := ((1,x,r_x),(2,y,r_y),(3,z,r_z)) \in \mathcal{E}_{1,3}, V(\ell) = I] \\
&= \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[\exists (y,r_y), y \neq z, \mathsf{Share}(x,r_x)_I = a_I, \mathsf{Share}(y,r_y)_I = a_I, \mathsf{Share}(y,r_y)_P = a_P, \\
&\qquad\qquad \mathsf{Share}(z,r_z)_P = a_P, V(\ell) = I] \qquad \text{(definition of } \mathcal{E}_{1,3}) \\
&= \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[\mathsf{Share}(x,r_x)_I = a_I, \exists (2,y,r_y) \in \mathsf{Vertices}(G'(P,I)), y \neq z, \mathsf{Share}(y,r_y)_{I \cup \{P\}} = a, \\
&\qquad\qquad \mathsf{Share}(z,r_z)_P = a_P, V(\ell) = I] \qquad (\mathsf{Vertices}(G'(P,I)) \subseteq \mathsf{Vertices}(G(P,I)))
\end{aligned}$$

Next we recall that the label of the ℓ can be determined without reference to (x, r_x). We will write $\ell_{2,3}$ as the edge connecting $(2, y, r_y)$ and $(3, z, r_z)$, and we note that $V(\ell) = V(\ell_{2,3})$. We note that the condition on x can now be written independently:

$$= \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[(1, x, r_x) \in C'_a] \cdot \Pr[\exists (2, y, r_y) \in \mathsf{Vertices}(G'(P, I)), y \neq z, \mathsf{Share}(y, r_y)_{I \cup \{P\}} = a,$$
$$\mathsf{Share}(z, r_z)_P = a_P, V(\ell_{2,3}) = I] \qquad \text{(definition of } C'_a\text{)}$$
$$= \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \frac{|C'_a|}{|\mathcal{M} \times \mathcal{R}|} \cdot \Pr[\exists (2, y, r_y) \in \mathsf{Vertices}(G'(P, I)), y \neq z, \mathsf{Share}(y, r_y)_{I \cup \{P\}} = a,$$
$$\mathsf{Share}(z, r_z)_P = a_P, V(\ell_{2,3}) = I] \qquad \text{(unif. of } (x, r_x) \in \mathcal{M} \times \mathcal{R}\text{)}$$
$$= \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \frac{2^m \cdot |H'_a|}{|\mathcal{M} \times \mathcal{R}|} \cdot \Pr[\exists (2, y, r_y) \in \mathsf{Vertices}(G'(P, I)), y \neq z, \mathsf{Share}(y, r_y)_{I \cup \{P\}} = a,$$
$$\mathsf{Share}(z, r_z)_P = a_P, V(\ell_{2,3}) = I] \qquad \text{(equation 3.9)}$$

Now in order to express this in a more convenient form and then replace the existence condition on y with something easier to manipulate, we introduce a fresh random variable (Y, r_Y) sampled independently and uniformly from $\mathcal{M} \times \mathcal{R}$:

$$= 2^m \cdot \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[(2, Y, r_Y) \in H'_a] \cdot \Pr[\exists (2, y, r_y) \in \mathsf{Vertices}(G'(P, I)), y \neq z, \mathsf{Share}(y, r_y)_{I \cup \{P\}} = a,$$
$$\mathsf{Share}(z, r_z)_P = a_P, V(\ell_{2,3}) = I] \qquad \text{(unif. of } (Y, r_Y) \in \mathcal{M} \times \mathcal{R}\text{)}$$
$$\geq 2^m \cdot \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[(2, Y, r_Y) \in H'_a, Y \neq z, (2, Y, r_Y) \notin V_R, \mathsf{Share}(z, r_z)_P = a_P, V(\ell_{2,3}) = I]$$

In this last expression, $\ell_{2,3}$ now denotes the edge between $(2, Y, r_Y)$ and $(3, z, r_z)$. Our labeling condition now applied to an edge between two uniformly sampled vertices at layer 2 and layer 3, hence we can directly apply our knowledge that the graph is I-oriented to conclude:

$$\geq \frac{2^m}{2} \cdot \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[(2, Y, r_Y) \in H'_a, Y \neq z, (2, Y, r_Y) \notin V_R, \mathsf{Share}(z, r_z)_P = a_P]$$

We next observe that the events $Y \neq z$ and $\mathsf{Share}(z, r_z)_P = a_P$ are independent, by privacy. This allows us to proceed as:

$$\geq (1 - 2^{-m}) \cdot \frac{2^m}{2} \cdot \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[(2, Y, r_Y) \in H'_a, (2, Y, r_Y) \notin V_R] \cdot$$
$$\cdot \Pr[\mathsf{Share}(z, r_z)_P = a_P] \qquad \text{(independence)}$$
$$= (1 - 2^{-m}) \cdot \frac{2^m}{2} \cdot \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[\mathsf{Share}(Y, r_Y)_I = a_I, \mathsf{Share}(Y, r_Y)_P = a_P, (2, Y, r_Y) \notin V_R] \cdot$$
$$\cdot \Pr[\mathsf{Share}(z, r_z)_P = a_P] \qquad \text{(definition of } H'_a\text{)}$$

We will next apply a union bound to remove the condition $(2, Y, r_Y) \notin V_R$, and then use our prior bound on the size of V_R:

$$\geq -\frac{|V_R|}{|\mathcal{M} \times \mathcal{R}|} + (1 - 2^{-m}) \cdot \frac{2^m}{2} \cdot \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[\mathsf{Share}(Y, r_Y)_I = a_I, \mathsf{Share}(Y, r_Y)_P = a_P] \cdot \\ \cdot \Pr[\mathsf{Share}(z, r_z)_P = a_P] \qquad \text{(union bound)}$$

$$\geq -2^{-k+1} + \frac{2^m}{2} \cdot \sum_{a \in \mathcal{S}_{I \cup \{P\}}} \Pr[\mathsf{Share}(Y, r_Y)_I = a_I, \mathsf{Share}(Y, r_Y)_P = a_P] \cdot \\ \cdot \Pr[\mathsf{Share}(z, r_z)_P = a_P] \qquad \text{(equation 3.7)}$$

Next we reorganize our sum by looking at each a_P value and summing over all the values of a_I:

$$= -2^{-k+1} + \frac{2^m}{2} \cdot \sum_{a \in \mathcal{S}_P} \Pr[\mathsf{Share}(Y, r_Y)_P = a_P] \cdot \Pr[\mathsf{Share}(z, r_z)_P = a_P]$$

The remainder of the calculation is an application of the Cauchy-Schwarz inequality after exploiting the fact that (Y, r_Y) and (z, r_z) are identically distributed and now subject to the same condition:

$$= -2^{-k+1} + \frac{2^m}{2} \cdot \sum_{a \in \mathcal{S}_P} \Pr[\mathsf{Share}(Y, r_Y)_P = a_P]^2 \qquad \text{(identical random variables)}$$

$$= -2^{-k+1} + \frac{2^m}{2} \cdot \frac{1}{|\mathcal{S}_P|} \cdot \sum_{a \in \mathcal{S}_P} \Pr[\mathsf{Share}(Y, r_Y)_P = a_P]^2 \sum_{a \in \mathcal{S}_P} 1^2$$

$$\geq -2^{-k+1} + \frac{2^m}{2} \cdot \frac{1}{|\mathcal{S}_P|} \cdot \left(\sum_{a \in \mathcal{S}_{\{P\}}} \Pr[\mathsf{Share}(Y, r_Y)_P = a_P] \cdot 1 \right)^2 \qquad \text{(Cauchy-Schwarz inequality)}$$

$$= -2^{-k+1} + \frac{2^m}{2} \cdot \frac{1}{2^\lambda} \qquad \text{(definition of } \lambda\text{)}$$

$$= 2^{m-\lambda-1} - 2^{-k+1}$$

Therefore, we must have

$$2^{m-\lambda-1} - 2^{-k+1} \leq 2^{-k},$$

which implies that

$$\lambda \geq m + k - (1 + \log_2(3)). \qquad \square$$

4 New Tools for Scheme Constructions

In this section, we develop some general tools that will be used in our efficient scheme construction. First, we will define a simple "locally hiding function" that generates two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$. While any single bit of the output is distributed identically in $\mathcal{D}_0$ and $\mathcal{D}_1$, the joint distribution of a relatively small number of bits is sufficient to distinguish $\mathcal{D}_0$ from $\mathcal{D}_1$ with high probability.

4.1 Locally Hiding Function

Definition 4.1 (Locally Hiding Function)**.** Let $\mathcal{D}_0 = \mathcal{U}_{\mathbb{F}_2^n}$ be the uniform distribution over $\mathbb{F}_2^n$, and let $\mathcal{D}_1 = \mathcal{U}_X$ be the uniform distribution over $X = \{0^n, 1^n\} \subseteq \mathbb{F}_2^n$. The *n-locally hiding function* is a randomized function $\eta : \mathbb{F}_2 \to \mathbb{F}_2^n$ defined as:

$$\begin{aligned} \eta : \mathbb{F}_2 &\longrightarrow \mathbb{F}_2^n \\ v &\longmapsto \mathcal{D}_v. \end{aligned}$$

Lemma 4.2 (Properties)**.** *The n-locally hiding function has the following properties:*

Local Hiding: *For any distribution $\mathcal{D}$ over $\mathbb{F}_2$, for any $v \in \mathbb{F}_2$, for any $i \in \{1, \dots, n\}$, and for any $w_i \in \mathbb{F}_2$, if $B \sim \mathcal{D}$,*

$$\Pr[B = v] = \Pr[B = v \mid \eta(B)_i = w_i].$$

Local Almost Invertibility: *For any $I \subseteq \{1, \dots, n\}$, $|I| = \alpha$, the function $\iota_I : \mathbb{F}_2^\alpha \to \mathbb{F}_2$*

$$\begin{aligned} \iota_I : \mathbb{F}_2^\alpha &\longrightarrow \mathbb{F}_2 \\ u &\longmapsto \begin{cases} 1 \text{ if } u \in \{0^\alpha, 1^\alpha\} \\ 0 \text{ otherwise} \end{cases} \end{aligned}$$

fails to invert η with probability less or equal to $2^{-\alpha+1}$. More formally, for any $v \in \mathbb{F}_2$,

$$\Pr[\iota_I(\eta(v)_I) \neq v] \leq 2^{-\alpha+1}.$$

Proof. To prove local hiding, notice that for any $i \in \{1, \dots, n\}$

$$\eta(0)_i = \left(\mathcal{U}_{\mathbb{F}_2^n}\right)_i = \mathcal{U}_{\mathbb{F}_2} = \left(\mathcal{U}_{\{0^n,1^n\}}\right)_i = \eta(1)_i,$$

which means that for any distribution $\mathcal{D}$ and $B \sim \mathcal{D}$, $\eta(B)_i$ is a uniform bit, independent of B. Therefore, for any $v, w_i \in \mathbb{F}_2$, we have $\Pr[B = v] = \Pr[B = v \mid \eta(B)_i = w_i]$.

To prove local almost invertibility, simple manipulation leads to the result:

$$\begin{aligned} \Pr[\iota_I(\eta(v)_I) \neq v] &= \Pr[\iota_I(\eta(v)_I) \neq v, v = 0] + \Pr[\iota_I(\eta(v)_I) \neq v, v = 1] \\ &\leq \Pr[\iota_I(\eta(0)_I) = 1] + \Pr[\iota_I(\eta(1)_I) = 0] \\ &\leq \Pr[\iota_I((\mathcal{U}_{\mathbb{F}_2^n})_I) = 1] + \Pr[\iota_I((\mathcal{U}_{\{0^n,1^n\}})_I) = 0] \\ &\leq \Pr[S \in \{0^\alpha, 1^\alpha\} \mid S \sim (\mathcal{U}_{\mathbb{F}_2^n})_I] + \Pr[S \notin \{0^\alpha, 1^\alpha\} \mid S \sim (\mathcal{U}_{\{0^n,1^n\}})_I\}] \\ &\leq \Pr[S \in \{0^\alpha, 1^\alpha\} \mid S \sim \mathcal{U}_{\mathbb{F}_2^\alpha}] + \Pr[S \notin \{0^\alpha, 1^\alpha\} \mid S \sim \mathcal{U}_{\{0^\alpha,1^\alpha\}}] \\ &\leq 2 \cdot 2^{-\alpha} = 2^{-\alpha+1}. \end{aligned}$$

□

4.2 Extended Locally Hiding Function

Definition 4.3 (Extended Locally Hiding Function)**.** Let η be the n-locally hiding function. For any vector space $\mathbb{F}_2^c$, the *extended n-locally hiding function* is the coordinate-wise extension of η, as follows:

$$\begin{aligned} \eta^c : \mathbb{F}_2^c &\longrightarrow \mathbb{F}_2^{n \times c} \\ v = (v_1, \dots, v_c) &\longmapsto (\eta(v_1), \dots, \eta(v_c)). \end{aligned}$$

Notice that the local hiding and invertibility properties are carried over as follows:

Lemma 4.4 (Properties)**.** *The extended n-locally hiding function has the following properties:*

Local Hiding: *For any distribution $\mathcal{D}$ over $\mathbb{F}_2^c$, for any $v \in \mathbb{F}_2^c$, for any $i \in \{1, \dots, n\}$, and for any $w_i \in \mathbb{F}_2^c$, if $B \sim \mathcal{D}$,*

$$\Pr[B = v] = \Pr[B = v \mid \eta^c(B)_i = w_i].$$

Local Almost Invertibility: *For any $I \subseteq \{1, \dots, n\}$, $|I| = \alpha$, the function $\iota_I^c : \mathbb{F}_2^{\alpha \times c} \to \mathbb{F}_2^c$*

$$\begin{aligned} \iota_I^c : \mathbb{F}_2^{\alpha \times c} &\longrightarrow \mathbb{F}_2^c \\ u = (u_1, \dots, u_c)^T &\longmapsto (\iota_I(u_1), \dots, \iota_I(u_c)) \end{aligned}$$

maps $u = \eta^c(v)$ "close to" v. More formally, for any $v \in \mathbb{F}_2^c$, $0 < \beta \leq 1$:

$$\Pr[w(v - \iota_I^c(\eta^c(v)_I)) \geq (1+\beta) \cdot c2^{-\alpha+1}] \leq e^{-\frac{c\beta^2}{3 \cdot 2^{\alpha-1}}}.$$

Proof. Similarly to the argument above, for all $v \in \mathbb{F}_2^c$, for all $i \in \{1, \dots, n\}$:

$$\eta^c(v)_i = (\eta(v_1), \dots, \eta(v_c))_i = (\eta(v_1)_i, \dots, \eta(v_c)_i) = (\mathcal{U}_{\mathbb{F}_2}, \dots, \mathcal{U}_{\mathbb{F}_2}) = \mathcal{U}_{\mathbb{F}_2}^c$$

which means that for any distribution $\mathcal{D}$ and $B \sim \mathcal{D}$, $\eta(B)_i$ is a uniform string of length c, independent of B. Therefore, for any $v, w_i \in \mathbb{F}_2^c$, we have $\Pr[B = v] = \Pr[B = v \mid \eta(B)_i = w_i]$.

To prove local almost invertibility, firstly for $i = 1, \dots, c$ define the following (Bernoulli) random variable:

$$x_i := \begin{cases} 1 \text{ if } v_i - \iota_I(\eta(v_i)_I) \neq 0 \\ 0 \text{ otherwise} \end{cases}$$

By the local almost invertibility property of the (standard) locally hiding function, we have

$$\Pr[x_i = 1] \leq 2^{-\alpha+1}$$

and applying the Chernoff bound in Lemma 2.3 on the x_i, for any $0 < \beta \leq 1 - 2^{-\alpha+1}$ we get

$$\Pr\left[\sum_{i=1}^{c} x_i \geq (1+\beta) \cdot c2^{-\alpha+1}\right] \leq e^{-\frac{c\beta^2}{3 \cdot 2^{\alpha-1}}}. \tag{4.1}$$

To conclude, notice that

$$(v - \iota_I^c(\eta^c(v)_I))_i = v_i - \iota_I^c(\eta^c(v)_I)_i = v_i - \iota_I(\eta(v_i)_I)$$

therefore $w(v - \iota_I^c(\eta^c(v)_I)) = \sum_{i=1}^c x_i$, by definition of x_i and Hamming weight. Combining this with equation 4.1, we get

$$\Pr\left[w(v - \iota_I^c(\eta^c(v)_I)) \geq (1+\beta) \cdot c2^{-\alpha+1}\right] \leq e^{-\frac{c\beta^2}{3 \cdot 2^{\alpha-1}}}. \qquad \square$$

4.3 Locally Hiding Transform

To use our locally hiding function inside an efficient robust secret sharing scheme, we would like it to be more resilient to inversion errors when we invert using a relatively small set of bits. This leads us to define the combined primitive of a locally hiding transform, a concatenation of an error-correcting code and our locally hiding function.

Definition 4.5 (Locally Hiding Transform)**.** Let $C : \mathbb{F}_2^h \to \mathbb{F}_2^c$ be a block (error-correcting) code over alphabet $\mathbb{F}_2$, with message length h, block length c and relative distance γ. Its *locally hiding transform* is a randomized function $\widehat{C} : \mathbb{F}_2^h \to \mathbb{F}_2^{n\times c}$, defined as $\widehat{C} = \eta^c \circ C$:

$$\widehat{C} : \mathbb{F}_2^h \to \mathbb{F}_2^{n\times c}$$

$$\begin{array}{ccccc} \mathbb{F}_2^h & \xrightarrow{\;C\;} & \mathbb{F}_2^c & \xrightarrow{\;\eta^c\;} & \mathbb{F}_2^{n\times c} \\ z = (z_1, \dots, z_h) & \longmapsto & C(z) = (v_1, \dots, v_c) & \longmapsto & (\eta(v_1), \dots, \eta(v_c)). \end{array}$$

Moreover, for any $I \subseteq \{1, \dots, n\}$ with $|I| = \alpha$, define $\widehat{D}_I = D \circ \iota_I$ (where D is the decoding function for C):

$$\widehat{D}_I : \mathbb{F}_2^{\alpha\times c} \to \mathbb{F}_2^h$$

$$\begin{array}{ccccc} \mathbb{F}_2^{\alpha\times c} & \xrightarrow{\;\iota_I^c\;} & \mathbb{F}_2^c & \xrightarrow{\;D\;} & \mathbb{F}_2^h \\ u = (u_1, \dots, u_c)^T & \longmapsto & (\iota_I(u_1), \dots, \iota_I(u_c)) = v & \longmapsto & D(v). \end{array}$$

Notice that the local hiding property of η^c is trivially translated to $\widehat{C}$. For local invertibility, if $\gamma > 2 \cdot (1 + \beta)2^{-\alpha+1}$, then $\widehat{D}$ is locally inverts $\widehat{C}$ with error probability less or equal to $e^{-\frac{c\beta^2}{3\cdot 2^{\alpha-1}}}$.

5 A Suitable MAC for our Scheme

5.1 The MAC and some of its Algebraic Properties

Definition 5.1. In the following, we assume that $h = 2 \cdot g$, $m = d \cdot g$, and use the following MAC, for $\mathcal{M} \subseteq \mathbb{F}_{2^m} \cong (\mathbb{F}_{2^g})^d$ (note that any set $\mathcal{M}$ can be thought of as a subset of $\mathbb{F}_{2^m}$, for large enough m), $\mathcal{K} = (\mathbb{F}_{2^g})^2$, and $\mathcal{T} = \mathbb{F}_{2^g}$:

$$\begin{array}{rcl} \mathsf{MAC} : (\mathbb{F}_{2^g})^2 \times (\mathbb{F}_{2^g})^d & \longrightarrow & \mathbb{F}_{2^g} \\ (a, b), (m_1, \dots, m_d) & \longmapsto & \displaystyle\sum_{l=1}^{d} a^l \cdot m_l + b. \end{array}$$

It is well known that the MAC described in definition 5.1 is ε-secure for $\varepsilon = d \cdot 2^{-g}$, [dB93, JKS93, Tay93].

Lemma 5.2. *The MAC described in definition 5.1 has the following properties:*

- *For any $m \in \mathcal{M}$ and $t \in \mathcal{T}$, there are at most 2^g different keys $z \in \mathcal{K}$ such that $\mathsf{MAC}(z, m) = t$.*
- *For $m_0, m_1 \in \mathcal{M}$, $m_0 \neq m_1$, and $t_0, t_1 \in \mathcal{T}$, there are at most d different keys $z \in \mathcal{K}$ such that $\mathsf{MAC}(z, m_0) = t_0$, $\mathsf{MAC}(z, m_1) = t_1$.*

Proof. For the first property, fix an arbitrary $m \in \mathcal{M}$ and $t \in \mathcal{T}$. Let define the set $K_{m,t} := \{z \in \mathcal{K} \mid \mathsf{MAC}(z, m) = t\}$ of keys that produce t as a tag of m. We want to study $|K_{m,t}|$. Using definition 5.1, we have

$$K_{m,t} = \left\{ (a, b) \in \mathbb{F}_{2^g}^2 \mid \sum_{l=1}^{d} a^l \cdot m_l + b = t \right\}$$

This means that if $(a, b) \in K_{m,t}$, then $b = t - \sum_{l=1}^{d} a^l \cdot m_l$. Therefore,

$$K_{m,t} = \left\{ \left(a, t - \sum_{l=1}^{d} a^l \cdot m_l \right) \in \mathbb{F}_{2^g}^2 \right\}$$

Since the function $a \mapsto (a, t - \sum_{l=1}^{d} a^l \cdot m_l)$ is a bijection from $\mathbb{F}_{2^g}$ to $K_{m,t}$ (with inverse $(a, b) \mapsto a$), we have $|K_{m,t}| = |\mathbb{F}_{2^g}| = 2^g$.

For the second property, let $m_0, m_1 \in \mathcal{M}$, $m_0 \neq m_1$, and $t_0, t_1 \in \mathcal{T}$. We want to study the cardinality of the following set X

$$X := \{z \in \mathcal{K} \mid \mathsf{MAC}(z, m_0) = t_0, \mathsf{MAC}(z, m_1) = t_1\}$$

Again, using definition 5.1,

$$X = \left\{ (a, b) \in \mathbb{F}_{2^g}^2 \mid \sum_{l=1}^{d} a^l \cdot m_{0,l} + b = t_0, \sum_{l=1}^{d} a^l \cdot m_{1,l} + b = t_1 \right\}$$

We can rewrite the above set as follows:

$$X = \left\{ \left(a, t_0 - \sum_{l=1}^{d} a^l \cdot m_{0,l} \right) \in \mathbb{F}_{2^g}^2 \mid \sum_{l=1}^{d} a^l \cdot (m_{0,l} - m_{1,l}) - t_0 + t_1 = 0 \right\} \tag{5.1}$$

Since $m_0 \neq m_1$, the polynomial $x \mapsto \sum_{l=1}^{d} x^l \cdot (m_{0,l} - m_{1,l}) - t_0 + t_1$ is a non-zero polynomial over $\mathbb{F}_{2^g}$ of degree at most d, which therefore has at most d roots. Since a is one of those roots, a can take only d values. From this, and the fact that for any $(a, b) \in X$ a completely defines b (by equation 5.1), we get that there are at most d pairs $(a, b) \in X$. □

5.2 Behavior towards Local Adversaries

We now prove another important property of the above MAC that will be useful for our construction of a robust secret sharing scheme. Intuitively, we want to study the probability that an honest message/tag pair is authenticated by any key that validates two distinct message/tag pairs, each of them chosen by a local adversary after seeing an honest message/tag pair. We also require that at least one between the two adversarially chosen pairs is not honest, otherwise the success probability of the adversaries would be trivially 1. To formalize this notion, we define the following game played between a challenger (who provides the honest message/tag pairs to the adversaries) and two, unbounded but non-communicating adversaries (whose target is to provide new message message/tag pairs).

Game A:

1. The challenger samples uniform messages $m_0, m_1 \neq m_2 \in \mathcal{M}$.
2. The challenger samples a uniform key $z \in \mathcal{K}$.
3. For $i = 0, 1, 2$, the challenger computes $t_i = \mathsf{MAC}(z, m_i)$.
4. For $i = 1, 2$, the challenger sends m_i, t_i to adversary i.
5. For $i = 1, 2$, adversary i generates $\widetilde{m}_i, \widetilde{t}_i$ and sends them to the challenger.
6. The challenger checks and whether $\widetilde{m}_2 \neq \widetilde{m}_1 \neq m_1$ and whether there exists $\widetilde{z}$ such that

$$t_0 = \mathsf{MAC}(\widetilde{z}, m_0), \qquad \widetilde{t}_1 = \mathsf{MAC}(\widetilde{z}, \widetilde{m}_1), \qquad \widetilde{t}_2 = \mathsf{MAC}(\widetilde{z}, \widetilde{m}_2).$$

   If so, the challenger sets $W = 1$; otherwise, it sets $W = 0$.

Lemma 5.3. *In the notation of* **Game A***,*

$$\Pr[W = 1] \leq 2 \cdot d \cdot \varepsilon.$$

Proof. In order to analyze $\Pr[W = 1]$, we define another game which is equivalent to **Game A** – equivalent in the sense that the distribution of the random variables that are involved remains the same. First, since in **Game A** the value m_0, t_0 are never revealed to any adversary, they might as well be generated after the challenger receives $\widetilde{m}_1, \widetilde{t}_1$ from adversary 1 and $\widetilde{m}_2, \widetilde{t}_2$ from adversary 2. Therefore, **Game A** is equivalent to the following game

Game A1:

1. The challenger samples uniform messages $m_1 \neq m_2 \in \mathcal{M}$.
2. The challenger samples a uniform key $z \in \mathcal{K}$.
3. For $i = 1, 2$, the challenger computes $t_i = \mathsf{MAC}(z, m_i)$.
4. For $i = 1, 2$, the challenger sends m_i, t_i to adversary i.
5. For $i = 1, 2$, adversary i generates $\widetilde{m}_i, \widetilde{t}_i$ and sends them to the challenger.
6. The challenger samples a uniform $m_0 \in \mathcal{M}$ and computes $t_0 = \mathsf{MAC}(z, m_0)$.
7. The challenger checks whether $\widetilde{m}_2 \neq \widetilde{m}_1 \neq m_1$ and whether ther exists $\widetilde{z}$ such that

$$t_0 = \mathsf{MAC}(\widetilde{z}, m_0), \qquad \widetilde{t}_1 = \mathsf{MAC}(\widetilde{z}, \widetilde{m}_1), \qquad \widetilde{t}_2 = \mathsf{MAC}(\widetilde{z}, \widetilde{m}_2).$$

   If so, the challenger sets $W = 1$; otherwise, it sets $W = 0$.

We are ready to analyze $\Pr[W = 1]$ in **Game A1**. First, define $\widetilde{Z} \subseteq \mathcal{K}$ as the set of keys compatible with $\widetilde{m}_1, \widetilde{t}_1$ and $\widetilde{m}_2, \widetilde{t}_2$, i.e.

$$\widetilde{Z} = \{\widetilde{z} \in \mathcal{K} \mid \widetilde{t}_1 = \mathsf{MAC}(\widetilde{z}, \widetilde{m}_1), \widetilde{t}_2 = \mathsf{MAC}(\widetilde{z}, \widetilde{m}_2)\}.$$

We can rewrite $\Pr[W = 1]$ as follows:

$$\begin{aligned} \Pr[W = 1] &= \Pr{}_{(z,m_0)}[\widetilde{m}_2 \neq \widetilde{m}_1 \neq m_1, \exists \widetilde{z} \in \widetilde{Z} : t_0 = \mathsf{MAC}(\widetilde{z}, m_0)] \\ &\leq \sum_{\widetilde{z} \in \widetilde{Z}} \Pr{}_{(z,m_0)}[\widetilde{m}_2 \neq \widetilde{m}_1 \neq m_1, t_0 = \mathsf{MAC}(\widetilde{z}, m_0)]. \end{aligned} \tag{5.2}$$

Making the requirement $t_0 = \mathsf{MAC}(\widetilde{z}, m_0)$ explicit, we obtain:

$$t_0 = \sum_{l=1}^{d} \widetilde{a}^l \cdot m_{0,l} + \widetilde{b}. \tag{5.3}$$

Now, remember that m_0 is uniform, and t_0 is computed as follows, for $z = (a, b)$ sampled according to step 2:

$$t_0 = \sum_{l=1}^{d} a^l \cdot m_{0,l} + b. \tag{5.4}$$

Subtracting equation 5.4 from equation 5.3, we get that any key $(\widetilde{a}, \widetilde{b})$ should satisfy

$$\sum_{l=1}^{d} \left(\widetilde{a}^l - a^l\right) \cdot m_{0,l} + \widetilde{b} - b = \left\langle \left(1, m_{0,1}, \ldots, m_{0,d}\right), \left(\widetilde{b} - b, \widetilde{a}^1 - a^1, \ldots, \widetilde{a}^d - a^d\right) \right\rangle = 0. \tag{5.5}$$

In equation 5.5, if $\widetilde{a} = a$, then $\widetilde{b} = b$. This means that $\widetilde{m}_1, \widetilde{t}_1$ is a valid message/tag pair for key (a, b), as it is valid for $(\widetilde{a}, \widetilde{b})$, since $(\widetilde{a}, \widetilde{b}) = (a, b)$. Since the MAC is ε-secure, and the adversaries are local (in particular adversary 1 only sees m_1, t_1 and provides $\widetilde{m}_1, \widetilde{t}_1$ with $m_1 \neq \widetilde{m}_1$), then $\widetilde{m}_1, \widetilde{t}_1$ is a forgery for (a, b) – since (a, b) is a uniform key, valid for both m_1, t_1 and $\widetilde{m}_1, \widetilde{t}_1$, with $\widetilde{m}_1 \neq m_1$. Therefore, for any $(\widetilde{a}, \widetilde{b}) = \widetilde{z} \in \widetilde{Z}$:

$$\Pr{}_{(z,m_0)}[\widetilde{m}_2 \neq \widetilde{m}_1 \neq m_1, t_0 = \mathsf{MAC}(\widetilde{z}, m_0), \widetilde{a} = a] \leq \varepsilon. \tag{5.6}$$

Now, If $\widetilde{a} \neq a$, then the vector $v = (\widetilde{b} - b, \widetilde{a}^1 - a^1, \ldots, \widetilde{a}^d - a^d) \in \mathbb{F}_2^{d+1}$ is non-zero, and equation 5.5 holds if and only if v is orthogonal to a uniformly chosen direction $u = (1, m_{0,1}, \ldots, m_{0,d})$, which happens with probability 2^{-g} for any non-zero v. Therefore,

$$\Pr{}_{(z,m_0)}[\widetilde{m}_2 \neq \widetilde{m}_1 \neq m_1, t_0 = \mathsf{MAC}(\widetilde{z}, m_0), \widetilde{a} \neq a] \leq 2^{-g} \leq \varepsilon. \tag{5.7}$$

Combining equations 5.6 and 5.7 with inequality 5.2 we get:

$$\Pr[W = 1] \leq \sum_{\widetilde{z} \in \widetilde{Z}} \Pr{}_{(z,m_0)}[\widetilde{m}_2 \neq \widetilde{m}_1 \neq m_1, t_0 = \mathsf{MAC}(\widetilde{z}, m_0)] \leq \sum_{\widetilde{z} \in \widetilde{Z}} 2 \cdot \varepsilon \leq 2 \cdot d \cdot \varepsilon,$$

since $|\widetilde{Z}| = d$, from lemma 5.2. □

6 An Efficient Scheme

The main idea behind our efficient scheme is similar to many other robust secret sharing schemes in the standard model: in order to achieve robustness we use Shamir's secret sharing scheme and expand each share with some authentication data so that any adversary who submits a corrupt share cannot provide authentication data that matches it. Differently from previous work, however, we have more freedom in what authentication data we can add, since each corrupt share depends only on a single share sent by the dealer, instead of depending on all the shares assigned to the adversary. We use this property and embed the same MAC key into each share and add a tag to the share in such a way that the key is not recoverable by individual corrupt players, while it is recoverable by the reconstructor, who will then check the authenticity of each share.

More precisely, we will use our locally hiding transform developed in Section 4 to distribute a MAC key among the parties so that it cannot be learned by a local adversary but can be reliably extracted from a number of honest shares. Recovery of the key and authentication in the reconstruction procedure will be performed by iterating over constant subsets of shares, extracting a candidate key value, and then attempting to authenticate at least $t+1$ shares. Since the local adversary cannot learn the real MAC key (during the execution of the protocol), we will prove that is it unlikely that a corrupted share will authenticate properly under the correct key. Similarly, we will prove it is unlikely for an incorrect candidate key to authenticate at least $t+1$ shares. The error-correcting code in our locally hiding transform will ensure that when we attempt to extract a key from a subset of honest shares, we produce the correct key with very high probability. Putting this all together, we can argue that the correct key will be recovered and the correct secret will be reconstructed.

6.1 Construction

In the following, we use the MAC defined in Section 5 and the locally hiding transform defined in Section 4. We let g denote the tag length of our MAC (the bit-length of its keys is then $h = 2 \cdot g$), and we define an additional parameter $d := m/g$, where m is the bit-length of messages. The security parameter for the MAC is $\varepsilon = d \cdot 2^{-g}$.

We give an explicit construction of our secret sharing scheme in Figure 1 and 2.

The procedure Share:

Local computation: In the notation of Shamir secret sharing scheme, for $i = 1, \ldots, n$ let x_i be the evaluation point associated with P_i. On input $s \in \mathcal{M}$, the dealer does the following:

1. Choose a uniform polynomial $f \in \mathcal{M}[X]$ of degree t such that $f(0) = s$. Compute Shamir shares $s_1 = f(x_1), \ldots, s_n = f(x_n)$.
2. Choose a uniform MAC key $z = (a, b) \in \mathcal{K}$.
3. Define $M \in (\mathbb{F}_2)^{n \times c}$ as $M = \widehat{C}(z)$.
4. For $i = 1, \ldots, t$, define $t_i = \mathsf{MAC}(z, s_i)$.

Share Distribution: For $i = 1, \ldots, n$, the dealer sends (M_i, s_i, t_i) to P_i.

Figure 1: The sharing procedure Share.

Theorem 6.1. *For $n = 2 \cdot t + 1$, the scheme $(\mathsf{Share}, \mathsf{Rec})$ given in Figure 1 and 2 is (t, δ)-robust against 1-local adversaries, where*

$$\delta = 2 \cdot (t+1) \cdot t/|\mathcal{M}| + \binom{n}{\alpha} \cdot (4 \cdot d \cdot \varepsilon + 5/|\mathcal{M}|) + e^{-\frac{c\beta^2}{3 \cdot 2^{\alpha-1}}}$$

Proof. Let $C \subseteq \{1, \ldots, n\}$ be the set of indices corresponding to the corrupt players, and let $H = \{1, \ldots, n\} \setminus C$ be the set of indices corresponding to the honest players. Also, let $A \subseteq C$ be the subset containing each player who in the reconstruction phase submits a share that was not the one obtained in the sharing procedure.

The procedure Rec:

Communication: Every player P_i sends (M_i, s_i, t_i) to the reconstructor.

Default Check: For $y \in \{s_1, \ldots, s_n\}$ define I_y as $I_y = \{i \in \{1, \ldots, n\} \mid s_i = y\}$. Then:

D1. If there exists y such that $|I_y| > t$, abort.

D2. If there exists y such that $|I_y| = t$, define $G = \{1, \ldots, n\} \setminus I_y$, use Shamir reconstruction on $(s_i)_{i \in G}$ to obtain s and finish.

D3. Else, proceed with the local computation.

Local computation: The reconstructor does the following, for each set $R \subseteq \{1, \ldots, n\}$ with $|R| = \alpha$:

L1. Evaluate $\widehat{D}_R(M_R)$ to obtain $z = (a, b)$.

L2. Define $G_R = \{i \in \{1, \ldots, n\} \mid t_i = \mathsf{MAC}(z, s_i)\}$.

L3. If $|G_R| \geq t + 1$, use Shamir reconstruction on $(s_i)_{i \in G_R}$ to obtain s and finish.

Figure 2: The reconstruction procedure Rec.

Privacy. For privacy, notice that each share (M_i, s_i, t_i) is composed by s_i, which is a share computed via Shamir secret sharing with a polynomial of degree t, M_i, which is uniform and independent of s, and $t_i = \mathsf{MAC}(z, s_i)$, which can be computed from s_i and z, where the latter is independent of s. Therefore, the privacy of (M_i, s_i, t_i) is reduced to the privacy of s_i, which is guaranteed by the privacy of Shamir secret sharing scheme when using a polynomial of degree t.

Robustness. For robustness, we study the probability that the reconstruction procedure outputs the correct secret value.

Default Check. First, we analyze the "Default Check" stage. These are the possible scenarios:

1. Step D1 is reached. The reconstructor goes to step D1 and aborts only if strictly more than t Shamir shares are equal, which is impossible for an honest set of shares (since a nonzero polynomial of degree at most t has at most t roots). Therefore, there exist $i \in \overline{A}$ and $a \in A$ such that $s_i = s_a$, which happens with probability $1/|\mathcal{M}|$, since s_i looks uniform to P_a. Assuming the worst case scenario in which all the honest players have different Shamir shares, the reconstruction aborts in step D1 with probability at most

$$\Pr[\mathsf{Rec} \text{ fails in step D1}] \leq (t+1) \cdot t / |\mathcal{M}| \tag{6.1}$$

by a union bound on the number of corrupt players and a union bound on the number (of the values of the shares) of the honest shares.

2. Step D2 is reached. In step D2, $|I_y| = t$, and we have the following cases:

   (a) $G \cap A \neq \varnothing$. There is at least a dishonest player in G, which means that $|A \cap I_y| < t$, therefore at least one honest player is in I_y. There are two subcases:

      i. $I_y \subseteq \overline{A}$. This happens with probability less or equal to $(1/|\mathcal{M}|)^t \cdot \binom{n}{t}$, since it is equivalent to the event of sampling a uniform polynomial of degree t that evaluates to a single value at t different points. This is less or equal to $\frac{1}{|\mathcal{M}|^t} \cdot \left(\frac{ne}{t}\right)^t \leq \left(\frac{e^2}{|M|}\right)^t$

ii. $I_y \not\subseteq \overline{A}$. In this case, there exist $a \in A$ and $i \in H$ such that $s_a = s_i$, which happens with probability at most $(t+1) \cdot t/|\mathcal{M}|$, as shown for step D1.

(b) $G \cap A = \varnothing$. In this case the reconstruction trivially succeeds, as none of the players in G submits dishonest shares.

This means that the reconstruction fails in step D2 with probability

$$\Pr[\mathsf{Rec}\text{ fails in step D2}] \leq \max\left\{\left(\frac{e^2}{|M|}\right)^t, (t+1)\cdot t/|\mathcal{M}|\right\} = (t+1)\cdot t/|\mathcal{M}| \qquad (6.2)$$

3. Step D3 is reached. This leads to the "Local Computation" stage, and is analyzed in the following.

By putting together inequalities 6.1 and 6.2, we show that:

$$\Pr[\mathsf{Rec}\text{ fails in "Default Check"}] \leq 2\cdot(t+1)\cdot t/|\mathcal{M}| \qquad (6.3)$$

Notice that if $|A| = t$ and there exists $y \in \mathcal{M}$ such that for $s_a = y$ for all $a \in A$, then "Default Check" either aborts in step D1, or reconstructs a (possibly incorrect) secret in step D2, but it never reaches step D3. This means that if step D3 is reached and $|A| = t$, then there exist $a_1 \neq a_2 \in A$ such that $s_{a_1} \neq s_{a_2}$. In other words, we ruled out the possibility that all the dishonest shares are the same when the set of dishonest players is of maximal size t.

Local Computation. Now we analyze "Local Computation", assuming that step D3 is reached in "Default Check". At this stage, for the reconstruction to succeed, it is sufficient that all the potential sets R that pass the test $|G_R| \geq t+1$ in step L3 lead to the right secret and there exists at least one set that passes the test. More formally, let s be the secret shared in the sharing procedure, and let F denote the function that takes $t+1$ indices and outputs the Shamir reconstruction applied to shares associated with the input indices; we then have

$$(\forall R \subseteq \{1,\ldots,n\}, |R| = \alpha : |G_R| \geq t+1 \Rightarrow F(G_R) = s) \wedge$$
$$\wedge\, (\exists R \subseteq \{1,\ldots,n\}, |R| = \alpha : |G_R| \geq t+1) \Rightarrow \mathsf{Rec}\text{ succeeds.}$$

Define the events $X =$ "$\forall R \subseteq \{1,\ldots,n\}, |R| = \alpha : |G_R| \geq t+1 \Rightarrow F(G_R) = s$, and $Y =$ "$\exists R \subseteq \{1,\ldots,n\}, |R| = \alpha : |G_R| \geq t+1$". We know that

$$\Pr[\mathsf{Rec}\text{ fails in "Local Computation"}] \leq \Pr[\overline{X} \vee \overline{Y}] \leq \Pr[\overline{X}] + \Pr[\overline{Y}], \qquad (6.4)$$

We now analyze $\Pr[\overline{Y}]$ and $\Pr[\overline{X}]$ separately.

Analysis of $\overline{Y}$. For $\Pr[\overline{Y}]$, notice that, if $R \subseteq \overline{A}$, then by the local invertibility property of $\widehat{C}$ at step L1 z is retrieved correctly except with probability $e^{-\frac{c\beta^2}{3\cdot 2^{\alpha-1}}}$. If so, $|G_R| \geq t+1$, because $H \subseteq \overline{A} \subseteq G_R$, since the correct shares respect the condition on the validity of the MACs expressed in the definition of G_R in step L2. Since there many such sets (as $R \subseteq H$ is sufficient for $R \subseteq \overline{A}$ and there are $\binom{t+1}{\alpha}$ sets $R \subseteq H$ with $|R| = \alpha$), we have

$$\Pr[\overline{Y}] \leq e^{-\frac{c\beta^2}{3\cdot 2^{\alpha-1}}}. \qquad (6.5)$$

(This is of course a very loose bound, but will be sufficient for our purposes.)

Analysis of $\overline{X}$. We now analyze $\Pr[\overline{X}]$. Recall that $X =$ "$\forall R \subseteq \{1,\dots,n\}, |R| = \alpha : |G_R| \geq t+1 \Rightarrow F(G_R) = s$", therefore $\overline{X} =$ "$\exists R \subseteq \{1,\dots,n\}, |R| = \alpha, |G_R| \geq t+1, F(G_R) \neq s$". Therefore,

$$\begin{aligned}\Pr[\overline{X}] &= \Pr[\exists R \subseteq \{1,\dots,n\}, |R| = \alpha, |G_R| \geq t+1, F(G_R) \neq s] \\ &\leq \sum_{\substack{R\subseteq\{1,\dots,n\},\\ |R|=\alpha}} \Pr[|G_R| \geq t+1, F(G_R) \neq s] \qquad (6.6)\end{aligned}$$

Assume that for a given R with $|R| = \alpha$, $|G_R| \geq t+1$. By a simple counting argument, this implies that $G_R \cap \overline{A} \neq \varnothing$; in other words, there is (at least) one player in G_R who submitted a correct share. Define the following events

- $V =$ "$|G_R| \geq t+1, |G_R \cap \overline{A}| = 1$",
- $W =$ "$|G_R| \geq t+1, |G_R \cap \overline{A}| \geq 2, |G_R \cap A| \geq 1$".

We can then rewrite inequality 6.6 as follows:

$$\begin{aligned}\Pr[\overline{X}] &\leq \sum_{\substack{R\subseteq\{1,\dots,n\},\\ |R|=\alpha}} \Pr[|G_R| \geq t+1, F(G_R) \neq s] \\ &\leq \sum_{\substack{R\subseteq\{1,\dots,n\},\\ |R|=\alpha}} \Pr[V, F(G_R) \neq s] + \Pr[|G_R| \geq t+1, |G_R \cap \overline{A}| \geq 2, F(G_R) \neq s] \\ &\leq \sum_{\substack{R\subseteq\{1,\dots,n\},\\ |R|=\alpha}} \Pr[V] + \Pr[|G_R| \geq t+1, |G_R \cap \overline{A}| \geq 2, F(G_R) \neq s] \\ &\leq \sum_{\substack{R\subseteq\{1,\dots,n\},\\ |R|=\alpha}} \Pr[V] + \Pr[W, F(G_R) \neq s] + \Pr[|G_R| \geq t+1, |G_R \cap \overline{A}| \geq 2, G_R \cap A = \varnothing, F(G_R) \neq s] \\ &\leq \sum_{\substack{R\subseteq\{1,\dots,n\},\\ |R|=\alpha}} \Pr[V] + \Pr[W] + \Pr[|G_R| \geq t+1, G_R \subseteq \overline{A}, F(G_R) \neq s] \\ &= \sum_{\substack{R\subseteq\{1,\dots,n\},\\ |R|=\alpha}} \Pr[V] + \Pr[W] \qquad (6.7)\end{aligned}$$

where the last step is due to the fact that if G_R contains only correct shares, then $F(G_R) = s$. We now analyze V and W separately.

Analysis of V. For V, notice that if $|G_R| \geq t+1$ and $|G_R \cap \overline{A}| = 1$ the reconstruction fails, since there is more than one corrupt player involved in the Shamir reconstruction. By a simple counting argument, we have $|G_R| = t+1$, $|A| = t$, and $G_R = A \cup \{i\}$, where $i \in \overline{A}$. Since "Default Check" reached step D3, and $|A| = t$, there exist $a_1 \neq a_2 \in A \subseteq G_R$ such that $s_{a_1} \neq s_{a_2}$. Moreover, $\Pr[s_{a_1} = s_i \vee s_{a_2} = s_i] \leq 2/|\mathcal{M}|$, by the fact that for $l = 1,2$, s_i looks uniform to P_{a_l}. Assuming $s_{a_1} \neq s_i \neq s_{a_2}$, lemma 5.3 applies (by replacing m_0, t_0 with s_i, t_i and for $l = 1,2$: m_l, t_l with s_{a_l}, t_{a_l}), and it guarantees that the probability that s_i, s_{a_1}, s_{a_2} and the corresponding tags are valid for the same key (which is a necessary condition for i, a_1, a_2 to be in G_R) is less or equal to $2 \cdot d \cdot \varepsilon$. This implies that

$$\Pr[V] \leq \Pr[V, s_{a_1} \neq s_i \neq s_{a_2}] + \Pr[V, (s_{a_1} = s_i \vee s_i = s_{a_2})] \leq 2 \cdot d \cdot \varepsilon + 2/|\mathcal{M}| \tag{6.8}$$

Analysis of W. For W, remember that in this case $|G_R \cap \overline{A}| \geq 2$, which means that there exist $i_1 \neq i_2 \in \overline{A} \cap G_R$. Moreover, $\Pr[s_{i_1} = s_{i_2}] = 1/|\mathcal{M}|$, since Shamir shares are t-wise independent. Assume $s_{i_1} \neq s_{i_2}$. Since $|G_R \cap A| \geq 1$, then there exists $a \in G_R \cap A$. By a similar argument to the one above, $\Pr[s_{i_1} = s_a \vee s_a = s_{i_2}] \leq 2/|\mathcal{M}|$. Assuming $s_{i_1} \neq s_a \neq s_{i_2}$, we again apply lemma 5.3 (by replacing m_1, t_1 with s_a, t_a and for $l = 0, 2$: m_l, t_l with s_{i_l}, t_{i_l}), to show that the probability that s_a, s_{i_1}, s_{i_2} and the corresponding tags are valid for the same key is less or equal to $2 \cdot d \cdot \varepsilon$. By the above analysis,

$$\begin{aligned}\Pr[W] &\leq \Pr[W, s_{i_1} = s_{i_2}] + \Pr[W, s_{i_1} \neq s_{i_2}] \\ &\leq 1/|\mathcal{M}| + \Pr[W, s_{i_1} \neq s_{i_2}, (s_{i_1} = s_a \vee s_a = s_{i_2})] + \Pr[W, s_{i_1} \neq s_{i_2}, s_{i_1} \neq s_a \neq s_{i_2})] \\ &\leq 1/|\mathcal{M}| + 2/|\mathcal{M}| + 2 \cdot d \cdot \varepsilon = 2 \cdot d \cdot \varepsilon + 3/|\mathcal{M}| \end{aligned} \tag{6.9}$$

Putting V and W together. By substituting inequalities 6.8 and 6.9 into inequality 6.7, we get

$$\begin{aligned}\Pr[\overline{X}] &\leq \sum_{\substack{R \subseteq \{1,\ldots,n\},\\ |R| = \alpha}} (\Pr[V] + \Pr[W]) \\ &\leq \sum_{\substack{R \subseteq \{1,\ldots,n\},\\ |R| = \alpha}} (2 \cdot d \cdot \varepsilon + 2/|\mathcal{M}| + 2 \cdot d \cdot \varepsilon + 3/|\mathcal{M}|) \\ &\leq \sum_{\substack{R \subseteq \{1,\ldots,n\},\\ |R| = \alpha}} (4 \cdot d \cdot \varepsilon + 5/|\mathcal{M}|) \\ &\leq \binom{n}{\alpha} \cdot (4 \cdot d \cdot \varepsilon + 5/|\mathcal{M}|) \end{aligned} \tag{6.10}$$

Putting $\overline{X}$ and $\overline{Y}$ together. Now, by substituting inequalities 6.5 and 6.10 into inequality 6.4, we get

$$\begin{aligned}\Pr[\textsf{Rec} \text{ fails in “Local Computation”}] &\leq \Pr[\overline{X}] + \Pr[\overline{Y}] \\ &\leq \binom{n}{\alpha} \cdot (4 \cdot d \cdot \varepsilon + 5/|\mathcal{M}|) + e^{-\frac{c\beta^2}{3 \cdot 2^{\alpha-1}}} \end{aligned} \tag{6.11}$$

Putting Default Check and Local Computation together. By inequality 6.3 and 6.11, we have that

$$\begin{aligned}\Pr[\textsf{Rec} \text{ fails}] &\leq \Pr[\textsf{Rec} \text{ fails in “Default Check”}] + \Pr[\textsf{Rec} \text{ fails in “Local Computation”}] \\ &\leq 2 \cdot (t+1) \cdot t/|\mathcal{M}| + \binom{n}{\alpha} \cdot (4 \cdot d \cdot \varepsilon + 5/|\mathcal{M}|) + e^{-\frac{c\beta^2}{3 \cdot 2^{\alpha-1}}} \end{aligned} \tag{6.12}$$

□

Corollary 6.2. *Given an error-correcting code C with block length $c = \Theta(g)$ and constant relative distance γ and $m = \Omega(g)$, there exists positive constants σ_1, σ_2 such that our construction in Figure 1 and 2 is δ-robust for $\delta \leq 2^{-k}$ and share size is*

$$m + c + g = m + c + k \cdot \sigma_1^{-1} + \sigma_2 \cdot \sigma_1^{-1} \cdot (\log(n) + \log(d)) = m + O(k).$$

Proof. Given γ, we can choose positive constants α and β such that $\gamma > 2 \cdot (1+\beta) \cdot 2^{-\alpha+1}$. Then, observe that

$$2 \cdot (t+1) \cdot t/|\mathcal{M}| = O(n^2 \cdot 2^{-m}),$$
$$\binom{n}{\alpha} \cdot (4 \cdot d \cdot \varepsilon + 5/|\mathcal{M}|) = O(n^\alpha \cdot (d^2 \cdot 2^{-g} + 2^{-m})),$$

and $d = m/g$. Note that $e^{-\frac{c\beta^2}{3 \cdot 2^{\alpha-1}}} \leq 2^{-\sigma' \cdot g}$ for some positive constant σ', as $c = \Omega(g)$ and α, β are constants. Thus, combining all of these terms with equation 6.12 yields a quantity $\delta \leq 2^{-\sigma_1 g + \sigma_2(\log(n) + \log(d))}$ for some σ_1, σ_2, when we have $m = \Omega(g)$. Setting

$$g = k \cdot \sigma_1^{-1} + \sigma_2 \cdot \sigma_1^{-1} \cdot (\log(n) + \log(d))\,,$$

we get $\delta \leq 2^{-k}$. Finally, the share size is $m+c+g$ (i.e. message length + key length + tag length), and replacing g with the formula above, we get that the share size is

$$m + c + k \cdot \sigma_1^{-1} + \sigma_2 \cdot \sigma_1^{-1} \cdot (\log(n) + \log(d))\,.$$

Since $n = \mathsf{poly}(k)$ (any scheme with super-polynomially many player would not be efficient), we get $g = O(k)$ and $c = O(k)$; therefore, the share size in our scheme is $m + O(k)$, as claimed. □

Note that the restriction that $m = \Omega(g)$ can be removed, if one simply shares the shorter secrets in $\mathcal{M}$ with Shamir shares over a field of bit length g. In this case, the share size becomes $g + c + g = m + c + O(g)$ instead of precisely $m + c + g$.

References

[AKL+09] Joël Alwen, Jonathan Katz, Yehuda Lindell, Giuseppe Persiano, abhi shelat, and Ivan Visconti. Collusion-free multiparty computation in the mediated model. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 524–540. Springer, 2009.

[AKMZ12] Joël Alwen, Jonathan Katz, Ueli Maurer, and Vassilis Zikas. Collusion-preserving computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 124–143. Springer, 2012.

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[Bla79] George Robert Blakley. Safeguarding cryptographic keys. In *Managing Requirements Knowledge, International Workshop on*, pages 313–317. IEEE Computer Society, 1979.

[BS97] Carlo Blundo and Alfredo De Santis. Lower bounds for robust secret sharing schemes. *Inf. Process. Lett.*, 63(6):317–321, 1997.

[CDF01] Ronald Cramer, Ivan Damgård, and Serge Fehr. On the cost of reconstructing a secret, or VSS with optimal reconstruction phase. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 503–523. Springer, 2001.

[Cev11] Alfonso Cevallos. Reducing the share size in robust secret sharing. `http://www.algant.eu/documents/theses/cevallos.pdf`, 2011.

[CFOR12] Alfonso Cevallos, Serge Fehr, Rafail Ostrovsky, and Yuval Rabani. Unconditionally-secure robust secret sharing with compact shares. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 195–208. Springer, 2012.

[CSV93] Marco Carpentieri, Alfredo De Santis, and Ugo Vaccaro. Size of shares and probability of cheating in threshold schemes. In Helleseth [Hel94], pages 118–125.

[CV12] Ran Canetti and Margarita Vald. Universally composable security with local adversaries. In Ivan Visconti and Roberto De Prisco, editors, *Security and Cryptography for Networks - 8th International Conference, SCN 2012, Amalfi, Italy, September 5-7, 2012. Proceedings*, volume 7485 of *Lecture Notes in Computer Science*, pages 281–301. Springer, 2012.

[dB93] Bert den Boer. A simple and key-economical unconditional authentication scheme. *Journal of Computer Security*, 2:65–72, 1993.

[DLWW11] Yevgeniy Dodis, Allison B. Lewko, Brent Waters, and Daniel Wichs. Storing secrets on continually leaky devices. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 688–697. IEEE, 2011.

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 293–302. IEEE Computer Society, 2008.

[Hel94] Tor Helleseth, editor. *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*. Springer, 1994.

[JKS93] Thomas Johansson, Gregory Kabatianskii, and Ben J. M. Smeets. On the relation between a-codes and codes correcting independent errors. In Helleseth [Hel94], pages 1–11.

[Jus72] J. Justesen. Class of constructive asymptotically good algebraic codes. *Information Theory, IEEE Transactions on*, 18(5):652–656, Sep 1972.

[LMs05] Matt Lepinski, Silvio Micali, and abhi shelat. Collusion-free protocols. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 543–552. ACM, 2005.

[MR04] Silvio Micali and Leonid Reyzin. Physically observable cryptography (extended abstract). In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer, 2004.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, New York, NY, USA, 2005.

[RB89] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85. ACM, 1989.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[Tay93] Richard Taylor. An integrity check value algorithm for stream ciphers. In Douglas R. Stinson, editor, *CRYPTO*, volume 773 of *Lecture Notes in Computer Science*, pages 40–48. Springer, 1993.