Information-Theoretic Optimistic Verifiable Secret Sharing

Martin Hirt[1], Chen-Da Liu-Zhang*[2], and Emanuele Marsicano[3]

[1]Department of Computer Science, ETH Zurich, hirt@inf.ethz.ch
[2]School of Computer Science and Information Technology, Lucerne University of Applied Sciences and Arts, chen-da.liuzhang@hslu.ch
[3]Department of Computer Science, ETH Zurich, emanuelefwm@gmail.com

May 20, 2026

Abstract

Verifiable secret sharing (VSS) is a fundamental primitive for secure computation and its round complexity has been well studied. The works of Gennaro et al. [STOC'01] and Fitzi et al. [TCC'06] settled the landscape in the perfect-security setting, showing that for the optimal corruption threshold $t < n/3$, the exact round complexity is three, and for the sub-optimal corruption threshold $t < n/4$ it is two rounds. Similarly, Patra et al. [CRYPTO'09] and Kumaresan et al. [ASIACRYPT'10] settled the landscape in the statistical setting, showing that for $t < n/2$ (resp. $t < n/3$), the exact round complexity is three (resp. two).

Current protocols with optimal resilience incur three rounds even when the actual number of corruptions f is sub-optimal. Fix corruption threshold parameters $0 \leq k \leq t$. We ask whether it is possible to obtain a VSS protocol that incurs two rounds when $f \leq k$, and three rounds when $k < f \leq t$. We show matching feasibility and impossibility results demonstrating that this is possible if and only if $3t + k < n$ for perfect security, and $2t + k < n$ for statistical security.

1 Introduction

Verifiable secret sharing (VSS) [17] is one of the most important building blocks in distributed computation tasks such as multi-party computation [5, 7–10, 13, 15, 16, 20, 23, 36–38, 40, 41, 56, 60] or randomness generation [12, 24, 45–47]. A VSS protocol is a two-phase protocol, consisting of a sharing phase and a reconstruction phase. During the sharing phase, one designated party (called the dealer) distributes a secret among n parties in such a way that no subset of t corrupted parties can infer information about the secret, and later, in the reconstruction phase, the secret is reconstructed.

The round complexity of VSS protocols has been the subject of a long line of works (e.g. [30, 33, 43, 52]). In the case of perfectly-secure VSS protocols, Gennaro et al. [33] have shown that for the sub-optimal range $t < n/4$, two rounds are necessary and sufficient to achieve VSS. In the

*C. Liu-Zhang was suported by SNSF the grant number 2000-1-243069.

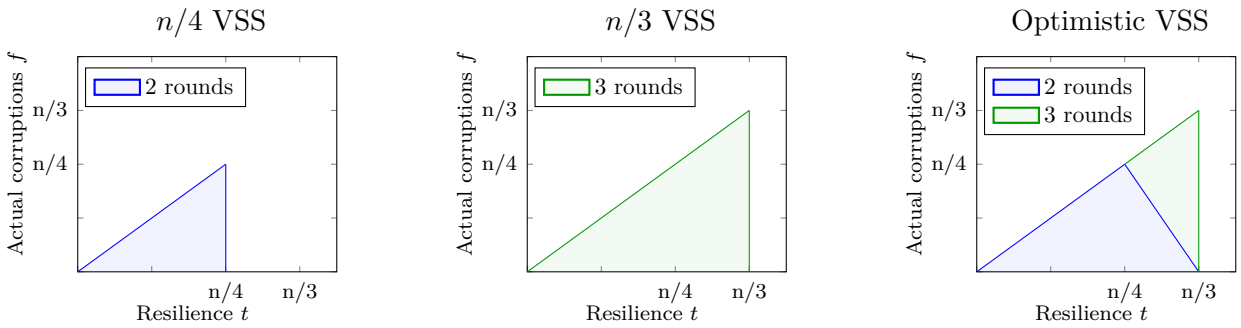

Figure 1: Regular VSS vs Optimistic VSS in the perfect security case.

optimal resilience $t < n/3$ range, Fitzi et al. [30] introduced a protocol with the optimal number of 3 rounds. Note that VSS cannot be achieved with perfect security for $t > n/3$ [25].

When considering the statistical setting where a negligible error probability is allowed, Patra et al. [52] have shown that 2-round VSS is possible if and only if one settles for the sub-optimal resilience range of $t < n/3$. Kumaresan et al. [43] instead showed that 3 rounds are necessary and sufficient for the optimal range $t < n/2$.

The results above indicate that there is an inherent trade-off between the upper bound on the corruption threshold t and the number of rounds incurred by the sharing phase of the VSS protocol. However, in all the protocols mentioned above, the round complexity remains the same even when the actual number of corruptions is much smaller than the upper bound $f \ll t$. In particular, protocols with optimal resilience (both in the perfect and statistical setting) incur three rounds, even if the actual number f of corrupted parties is in the sub-optimal range (where the optimal number of rounds should be two). This motivates the question of whether one can achieve an *optimistic* VSS protocol whose round complexity is two when the actual corruption is in the sub-optimal range, and three rounds otherwise. More precisely, we ask the following question:

Fix $0 \leq k \leq t$. Is there an optimistic VSS protocol that incurs two rounds when the actual number of corruptions is $f \leq k$, and three rounds when $k < f \leq t$?

1.1 Our Contributions

We completely resolve this question by providing matching feasibility and impossibility results. More concretely, we show that optimistic VSS is achievable in the perfect security setting when $n > 3t+k$, and in the statistical setting when $n > 2t+k$. The graphs in Fig 1 show the improvements of our optimistic VSS protocols compared to previous VSS protocols in the perfect security case (the statistical case is similar). The graph displays the number of rounds required to complete the sharing phase when varying the resilience and the actual number of corrupted parties. Of independent interest, we extend the perfect-security protocols to the setting of general adversaries.

Note that our results show improvements over the state of the art even for small n. As an example, for $n = 12$ and perfect security, one can choose among the existing 2-round protocol that tolerates $t = 2$, or the existing 3-round protocol that tolerates $t = 3$. But if one chooses our new protocol with parameters $t = 3$ and $k = 2$, one achieves the best of both worlds.

1.2 Further Related Work

Optimistic protocols attempt to achieve a better efficiency metric (latency, communication, etc.) when an optimistic condition is satisfied, and several conditions have been considered in the literature. A common optimistic condition is considering executions where a small number of parties (sometimes even zero) is corrupted, or a designated party is honest (the broadcaster or dealer). To the best of our knowledge, current synchronous VSS protocols focus on the worst-case number of rounds and do not explicitly attempt to improve the round complexity under optimistic conditions. Nevertheless, there is an extensive literature on optimistic protocols for agreement primitives.

A line of work [2, 14, 49] considers agreement protocols and optimize the *good-case* latency, measured as the number of rounds for all honest parties to output when the designated broadcaster is honest. Another traditional line of work investigates protocols that have a number of rounds proportional to the actual number of corruptions f, rather than the known upper bound on the number of corruptions t. In this case, it is known that deterministic broadcast solutions have $\min\{f+2, t+1\}$ rounds of communication [27, 28]. From the feasibility side, there are protocols without setup [1, 11, 18, 26, 27, 32, 57, 58], or protocols with a setup for cryptographic signatures or pseudosignatures [22, 48, 54].

Many of the current randomized solutions for BA [21,29,31,34,35,42], where parties have access to local randomness, achieve an expected constant number of rounds, and the exact constant can be made to depend on the actual number of corruptions [29, 31, 42, 50, 59].

A different line of work attempted to optimize *the delay* of each round (rather than the number of rounds), by making a protocol make progress as fast as actual network delay in an optimistic case when the number of corruptions is small [44, 51].

Optimistic communication. The work [19] considered protocols with optimistic communication $O(nf)$ for byzantine agreement, improving upon traditional protocols that achieved $O(nt)$ communication.

2 Technical Overview

We outline the main techniques used throughout the paper.

2.1 A Simpler Problem

Our first step contribution is in identifying a new primitive, which we denote *detectable* VSS (DVSS). A DVSS is also parameterized by two corruption thresholds, $0 \leq k \leq t$. The difference with respect to regular VSS scheme, is that the sharing phase is guaranteed to succeed in two rounds only when the actual number of corruptions is $f \leq k$. In the regime where $k < f \leq t$, the sharing phase is allowed to fail, but the primitive guarantees that parties reach agreement on whether the sharing has succeeded or failed.

We then make the observation that one can easily combine an instance of (k,t)-DVSS together with a regular t-VSS in order to construct an optimistic (k,t)-VSS scheme, simply by running both instances in parallel and outputting the share of DVSS if it didn't fail, and the share of VSS otherwise.

The resulting optimistic (k,t)-VSS scheme achieves perfect (resp. statistical) security if both (k,t)-DVSS and t-VSS are perfect (resp. statistical). And if (k,t)-DVSS is perfect and t-VSS achieves statistical correctness and commitment (albeit perfect privacy), the resulting optimistic (k,t)-VSS is also perfectly secure when there are up to k corruptions. See Section 5 for details.

Therefore, we turn out attention into initiating a systematic study of DVSS schemes.

2.2 Perfect DVSS

We first focus on the case of perfect security and build DVSS protocols with 2 rounds. Combining the DVSS with the 3-round VSS of Fitzi et al. [30], gives us an optimistic VSS for $n > 3t+k$ parties where the sharing phase takes 2 rounds if we have at most k corruptions, and 3 rounds if we have up to t corruptions. As an add-on, we also present results for general adversaries in Appendix B.

The Honest-Dealer Case. As a starting point, we explore the weaker notion of honest-dealer DVSS (HD-DVSS) protocols, which has all the properties of DVSS, except that the sharing phase is guaranteed to succeed only in the case where $f \leq k$ **and** the dealer is honest.

We build a 2-round HD-DVSS protocol which is a simple adaptation of the 2-round VSS protocol for $n > 4t$ parties given in [33].

The protocol uses the classical idea of sharing the secret s by picking a random degree t two dimensional polynomial $F(x, y)$ such that $F(0, 0) = s$, and then sending to each party P_i the horizontal and vertical projections $f_i(x) = F(x, i)$ and $g_i(y) = F(i, y)$. If the dealer is honest, all the honest parties will have projections which lie on the same two dimensional polynomial and hence during reconstruction we will be able to reconstruct it (and the dealer's secret), even if the corrupted parties provide wrong values, by using error correction.

If the dealer is corrupted we are not guaranteed that the polynomials sent by the dealer all lie on the same two dimensional polynomial. Hence, we have every pair of parties P_i and P_j perform pairwise consistency checks in order to see if the points they have in common are the same, i.e. if $f_i(j) = g_j(i)$. Moreover, we do this in a way such that every party can tell which pairs of parties agree and which do not, by having each pair of parties broadcast their common points (padded by a random pad exchanged in the first round in order to still maintain privacy of the protocol).

If two parties disagree on the points they have in common, then either one (or both) of them is corrupted, or the dealer is corrupted. We can then simply "disqualify" every pair of parties that has a disagreement. If after disqualifying k pairs we still have more disagreements, we are sure that either the dealer is corrupted or that there are more than k corrupted parties. In either case, we can reject the sharing.

Otherwise, all the remaining honest parties will have polynomials which are mutually consistent, and there are enough of them to determine a unique two dimensional polynomial $F^*(x, y)$. We then can add back any party which was previously disqualified, as long as we can guarantee that their projections also lie on $F^*(x, y)$. If after doing this we have enough parties left to correct up to t errors during reconstruction, we accept the sharing; and otherwise we reject it.

The protocol is formally described in Section 6.

From HD-DVSS to DVSS. We then show how to lift the HD-DVSS above to DVSS. The idea is as follows: instead of using a two dimensional polynomial, we will use a one dimensional degree t polynomial f with $f(0) = s$, and every party receives one point on the polynomial from the dealer. Moreover, we will not perform any consistency checks during the sharing phase. Instead, if we notice during reconstruction that there are more than t errors, we conclude that the dealer was corrupted and reconstruct a default secret.

In order for this to work, we need to make sure that every party is committed at the end of the sharing phase to what value they will provide during the reconstruction phase, since otherwise a corrupted dealer can send t wrong values to honest parties in the sharing phase, and then depending on whether during the reconstruction phase the corrupted parties provide wrong values or not, we

will reconstruct either the default secret or the dealer's secret. We will achieve this by having each party share a random pad $r \in \mathcal{K}$ using HD-DVSS and broadcasting the share they received padded by r. If the sharing of a party fails, we know it is corrupted and hence we can disqualify it. If there are at most k disqualified parties, we will then still have enough parties left for which we will be able to uniquely reconstruct their value and hence uniquely reconstruct the dealer's secret. This described protocol is formally described in Section 6.2.

The total communication complexity incurred by the protocol is $O(n^3)$. In Appendix A, we give a protocol with $O(n^2)$ communication, albeit with super-polynomial computation.

2.3 Statistical DVSS

We now examine the statistical setting to construct an efficient 2-round detectable VSS. Combining this with the 3 round protocol of Kumaresan et al. [43], we obtain an optimistic VSS for $n > 2t + k$ parties where the sharing phase takes 2 rounds if we have at most k corruptions, and 3 rounds if we have up to t corruptions. Similar to the perfect security case, we first aim for an HD-DVSS. For that, we adapt the protocol of Patra et al. [53].

First Step: HD-DWSS. The first step of the protocol is to construct an honest-dealer *weak* secret sharing. The difference with respect to HD-DVSS is that after a successful run of the sharing phase, the adversary can still unanimously let all honest parties to obtain the value $\perp$ (this is analogous to a cryptographic commitment, which needs the help of the committer to open a value).

The idea is as follows: we again use a random two dimensional polynomial $F(x, y)$ with $F(0, 0) = s$, where s is the secret. We do not do any consistency checks, instead if during reconstruction we notice that the values provided do not interpolate to a degree t polynomial, we simply output $\perp$. This implies that depending on what the corrupted parties provide during reconstruction, we could reconstruct either the dealer's secret or $\perp$, but this is fine if the dealer is corrupted since we are only constructing a WSS.

On the other hand, even for a WSS we need to guarantee that the dealer's secret is reconstructed in case the dealer is honest. In the statistical setting, we do not have enough parties to perform error correction, which means we need to ensure that a corrupted party during reconstruction cannot provide a polynomial different than the one he received from the dealer.

We will achieve this by having the dealer send to each party some points on the polynomials of every other party, and having them randomly reveal part of them during the sharing phase and the remaining ones during the reconstruction phase. The points revealed during the reconstruction phase will then be used to ensure that with high probability a corrupted party cannot reveal a polynomial different than the one he received from the dealer, by having each party P_i accept other parties P_j depending on whether the points that P_i knows from P_j's polynomial do indeed lie on the polynomial revealed by P_j. Any polynomial revealed will then only be used if we are guaranteed that a honest party accepted it. On the other hand, the points revealed during the sharing phase will be used to ensure that if a honest party has received proper values from the dealer, it will with high probability not be disqualified during reconstruction.

Second Step: From HD-DWSS to HD-DVSS. Note that the HD-DWSS protocol shown above can be also be used to share a degree t polynomial $g(y)$ rather than a secret value s, by picking a random two dimensional polynomial $F(x, y)$ such that $F(0, y) = g(y)$. In fact, this is how we will use the protocol in order to construct the HD-DVSS protocol below.

The idea is that we will again use a two dimensional degree t polynomial $F(x, y)$. This time, we use a symmetric polynomial so that the horizontal and vertical projections for each party are equal. Instead of using random pads for the consistency checks, we use random polynomials shared using the HD-DWSS protocol. This makes sure that the value provided by each party during reconstruction is uniquely determined at the end of the sharing phase; instead of having each party share their polynomial during reconstruction we instead run the reconstruction protocol of HD-DWSS in order to reconstruct the random pad polynomials and then determine the actual polynomial from the broadcasted values.

Third Step: From HD-DVSS to DVSS. The idea here will be similar as in the perfect-security setting, where the dealer shares the secret with a one dimensional degree t polynomial f and send to each party one point on the polynomial, and then every party starts an instance of the HD-DVSS protocol to share a random pad and broadcast the padded value he received from the dealer. This ensures that a party is committed in the sharing phase to what share they will provide during the reconstruction phase. However, note that a corrupted party can still commit to a share that is different than what it received from the dealer. This is not a problem in the perfect-security setting, as we have at least $3t + 1$ parties during reconstruction, and thus enough parties to correct up to t errors. On the other hand, in the statistical setting, we are only guaranteed to have $2t + 1$ parties during reconstruction. Hence we cannot perform error correction. Instead, we will make use of *information checking* authentication tools [55]. See more details in Section 7.

2.4 Lower Bounds

Next, we show the optimality in terms of the number of parties of the 2 round detectable VSS protocols we constructed. More specifically, we show that even any 2 round detectable HD-DVSS scheme requires $n > 3t + k$ parties in the perfect-security setting, and $n > 2t + k$ parties in the statistical-security setting.

Our lower bounds follow a partition-based approach, where the set of parties is partitioned into different sets and the behavior of each set is well defined through a set of executions. Then, an attack is devised to argue that if some of the security guarantees hold, another different security guarantee is broken. The technical contribution is in carefully designing the attack, as well as such executions that can make use of the security guarantees of HD-DVSS to reach a contradiction. Differently from prior bounds, our proof crucially leverages the HD-DVSS specific fact that when the dealer is honest and only k parties are corrupted, the sharing phase succeeds. In the following, we provide a high-level description of each lower bound.

Perfect Security Lower Bound. First note that one can assume without loss of generality that every message after the first round is sent via broadcast, with the simple observation that every pair of parties can exchange random secret pads for every future private message in the first round, and use these pads to broadcast the private messages blinded by the pads.

We proceed by contradiction. Assume that we have a 2-round HD-DVSS protocol for $n = 3t+k$ parties. We partition the party set $\mathcal{P}$ into four sets A, B, M and K, the first three containing t parties each, and the final one containing the remaining k parties. For simplicity we consider an external dealer, but the proof follows similarly with an internal dealer located in set A.

We now construct an attack against the protocol that breaks the commitment property. We use the following notation: for any two sets of parties S and R, let m_{SR} denote all the messages sent during the first round by any party in set S to any party in set R. The messages the dealer sends in round one to parties in S are denoted by s_S. Moreover, for any set S let B_S^1 and B_S^2 denote all

the broadcasts done by parties in set S during the first and second round respectively. Similarly, the broadcasts performed by the dealer is denoted by B_D^1 and B_D^2.

We first consider an execution E_1 where an honest dealer is sharing a secret s and nobody is corrupted. Further consider another execution where an honest dealer sharing a secret $\overline{s} \neq s$, with no corruptions. By privacy, there has to be one execution E_2 of the protocol where the joint view of all the parties in block M (containing t parties) is identical to the one in E_1.

The high level idea for a protocol attack is as follows: the dealer and the parties in block A will be corrupted and act in such a way that parties in block B believe we are in execution E_1 but parties in K are corrupted, while parties in block K believe that we are in execution E_2 but parties in B are corrupted.

Contrary to prior lower bounds, we leverage the fact that since it is plausible from the view of some honest parties that only the set K (containing k parties) was corrupted and the dealer was honest, the sharing phase must actually succeed for them (and for all other honest parties, since the protocol guarantees agreement on whether the sharing succeeds). However, in reality it is the set A that was actually corrupted. Then, depending on how parties in A behave during the reconstruction, honest parties must reconstruct either s or $\overline{s}$, breaking the commitment property. See Section 8.1 for details.

Statistical Security Lower Bound. The statistical proof proceeds also by contradiction. Assume that we have a 2-round HD-DVSS protocol for $n = 2t + k$ parties. We again partition the party set $\mathcal{P}$ into three sets A, B, and K, the first two containing t parties each, and the final one containing the remaining k parties. For simplicity we consider an external dealer, the proof follows similarly with an internal dealer located in set A.

The high-level idea of the proof is the following: we will consider an execution where parties in set K receive no information either from the dealer or from parties in set A. Moreover, during the reconstruction phase parties in set A will not participate. In either case, the protocol needs to be able to reconstruct the secret, since this execution can be explained by the set K being corrupted. However, intuitively this also implies that parties in set B hold all the information needed to reconstruct the secret, thus breaking the secrecy property. See details in Section 8.2.

3 Preliminaries

3.1 Notation

We denote by $\mathcal{K}$ an arbitrary finite field with 2^κ elements. For a set S, we denote its power set (the set of all subsets of S) by $P(S)$. For any graph $G = (V, E)$, for any vertex $v \in V$ and any set $C \subseteq V$, we denote by $e_G(v, C)$ the number of edges $\{v, u\}$ with $u \in C$. We denote by $\bar{G}$ the complement graph of G.

3.2 Model

We consider protocols involving a set of n parties $\mathcal{P} = \{P_1, \ldots, P_n\}$. The adversary $\mathcal{A}$ has unbounded computing power and can actively corrupt up to t of the parties. For simplicity, we consider static corruption, although we believe all our protocols can be proven adaptively secure. Parties have access to a complete synchronous network of secure point-to-point channels. In addition, parties have access to Byzantine broadcast channels. We consider synchronous protocols that proceed in rounds, such that every message that is sent via a point-to-point channel or broadcast in round i is received before the start of round $i + 1$.

As almost all of the messages exchanged will be field elements, we will split up the communication complexity in two parts: the number of field elements exchanged, and any additional bits from messages which are not field elements. When broadcasting, we consider the cost of sending the message via the broadcast channel the same as sending it via a point to point channel.

3.3 Reed-Solomon error correction codes

In our protocols we will often have a collection of points where most of them lie on some degree t polynomial $f \in \mathcal{K}[x]$, and we want to recover the polynomial f. This can be achieved using Reed-Solomon error correction codes. We will not give an explicit construction, but instead assume we have a black box function RS-EC$_t$ which takes as input a set of points of size at least $3t+1$ and returns a polynomial f of degree at most t such that all but at most t points lie on f, or the zero polynomial if no such polynomial exists.

3.4 Verifiable secret sharing

A VSS protocol is a multiparty protocol that allows one designated party $D \in \mathcal{P}$ called the dealer to share a value with all the other parties, such that the value remains secret until the parties reconstruct it later. More specifically, a VSS protocol consists of the following two phases, each of which may take several rounds:

Sharing: The dealer holds an input s, called the secret. At the end of the sharing phase, each party P_i outputs a value v_i, called its share of the secret.

Reconstruction: Each party has as input its share v_i from the sharing phase. At the end of the reconstruction phase, each party P_i outputs a value s'_i.

Definition 1. *We say that a two phase protocol as described above is a (n, t) VSS protocol with perfect security if for every t-adversary $\mathcal{A}$ the following three properties hold:*

- ***Privacy:*** *If the dealer D is honest, then the adversary should gain no information about the secret value s. More specifically, the adversary's view (the combined view of all of the corrupted parties) after the sharing phase should have the same distribution no matter the value of s.*
- ***Correctness:*** *If the dealer D is honest, then at the end of the reconstruction phase all of the honest parties output the dealer's value, i.e. for all honest P_i we have $s'_i = s$.*
- ***Commitment:*** *If the dealer D is corrupted, then at the end of the sharing phase there is a unique value s^*, such that at the end of the reconstruction phase every honest party P_i outputs $s'_i = s^*$.*

Our main focus will be optimizing the number of rounds of VSS protocols. As reconstruction typically takes only one round, we will mainly care about the number of rounds in the sharing phase.

Statistical security. The definition given above is for a VSS with perfect security, where the Correctness and Commitment properties hold unconditionally. If we relax these conditions so that they can fail with negligible probability ε, while Privacy still holds unconditionally, we obtain a statistically secure VSS:

Definition 2. *A two phase protocol is a (n, t) VSS protocol with statistical security for some security parameter κ if for every t-adversary $\mathcal{A}$ the Privacy property holds as in Definition 1, and for statistical error parameter $\varepsilon = negl(\kappa)$ the following two properties hold:*

- ***(1 - ε)-Correctness:*** *If the dealer D is honest, then with probability $1 - \varepsilon$ at the end of the reconstruction phase all of the honest parties output the dealer's value, i.e. for all honest P_i we have $s'_i = s$.*
- ***(1 - ε)-Commitment:*** *If the dealer D is corrupted, then with probability $1 - \varepsilon$ at the end of the sharing phase there is a unique value s^*, such that at the end of the reconstruction phase every honest party P_i outputs $s'_i = s^*$.*

4 Definitions for VSS Variants

In this section, we give definitions for optimistic VSS (OVSS) and detectable VSS (DVSS).

Definition 3. *We say that a two phase protocol is a (r, R) (n, t, k) OVSS protocol if it is a (n, t) VSS protocol with round complexity r if the adversary corrupts at most k parties, and round complexity R otherwise.*

It is clear that if we are given an OVSS protocol, at the end of the first r rounds of the sharing phase every party is able to tell if the sharing phase has already ended or if we need to go on with the remaining rounds. In fact, we can view the first r rounds of an OVSS protocol as a separate type of VSS protocol in itself, where the sharing phase takes r rounds, and at the end of those rounds if at most k parties were corrupted we have a successful sharing of the secret, and otherwise we have agreement on the fact that the sharing has failed.

In fact, this is the type of VSS protocol we will mainly be focused on constructing. We will call such a protocol a detectable VSS (DVSS) protocol. More specifically, in a DVSS protocol at the end of the sharing phase each party outputs a bit b_i which indicates if the sharing has succeeded or not, such that all honest parties output the same bit. If the sharing has succeeded, then each party also outputs its share v_i. The privacy property still holds no matter the outcome of the sharing phase, while the correctness and commitment properties only hold if the sharing has succeeded:

Definition 4. *We say that a two phase protocol is a (n, t, k) DVSS protocol with perfect security if for every t-adversary $\mathcal{A}$ the following five properties hold:*

- ***Privacy:*** *If the dealer D is honest, then the adversary should gain no information about the secret value s. More specifically, the adversary's view (the combined view of all of the corrupted parties) after the sharing phase should have the same distribution no matter the value of s.*
- ***Agreement:*** *For any two honest parties P_i and P_j, we have $b_i = b_j$.*
- ***Validity:*** *If the adversary is a k-adversary, then the sharing phase succeeds, i.e. we have for every honest party P_i that $b_i = 1$.*

- ***Correctness:*** *If the dealer D is honest and the sharing phase succeeds, then at the end of the reconstruction phase all of the honest parties output the dealer's value, i.e. for all honest P_i we have $s'_i = s$.*
- ***Commitment:*** *If the dealer D is corrupted and the sharing phase succeeds, then at the end of the sharing phase there is a unique value s^*, such that at the end of the reconstruction phase every honest party P_i outputs $s'_i = s^*$.*

If we relax the Correctness and Commitment properties to allow for negligible error property as in Definition 2, we obtain the definition of statistical DVSS.

In many cases it will be easier to first focus on protocols where the validity property only holds in the case where the dealer is honest. We will call such a protocol a honest dealer DVSS (HD-DVSS).

Definition 5. *We say that a two phase protocol is a (n, t, k) HD-DVSS protocol with perfect security if for every t-adversary $\mathcal{A}$ the Privacy, Agreement, Correctness and Commitment properties hold as in Definition 4, and the following property holds:*

- ***HD-Validity:*** *If the dealer is honest and the adversary is a k-adversary, then the sharing phase succeeds, i.e. we have for every honest party P_i that $b_i = 1$.*

As usual, by relaxing the Correctness and Commitment property we obtain the corresponding definition of statistical HD-DVSS.

5 Constructing OVSS from DVSS

Suppose we are given an r-round (n, t, k) DVSS protocol $\Pi_{\mathtt{DVSS}}$ and an R-round (n, t) VSS protocol $\Pi_{\mathtt{VSS}}$, we can now combine them in order to obtain a (r, R) (n, t, k) OVSS protocol. The idea is simple: we start out by running both protocols at the same time. If at the end of the first r rounds the sharing in the DVSS protocol has succeeded, we simply take the resulting share of the DVSS protocol and ignore anything that was done in the VSS protocol. On the other hand, if the sharing in the DVSS protocol failed, we run the remaining rounds of the VSS protocol and take the resulting share. Note that in addition to the share of either the DVSS or the VSS protocol, the flag b_i will also be part of the share of OVSS, as in the reconstruction phase we need to be able to tell if we will run the reconstruction phase of the DVSS or the VSS protocol.

Note that the resulting OVSS inherits the security level of the subprotocols. That is, it is perfectly (resp. statistical) secure if both sub-protocols DVSS and VSS are perfectly (resp. statistical) secure. Moreover, the OVSS is perfectly secure up to k corruptions and statistical up to t corruptions if DVSS is perfectly secure and VSS is statistical.

Protocol $\Pi_{\mathtt{OVSS}}$

Sharing Phase

Party P_i

1: Initiate an instance of protocol $\Pi_{\mathtt{DVSS}}$ and one of $\Pi_{\mathtt{VSS}}$. Participate in the first r rounds of both protocols, acting as either the dealer (if $P_i = \mathcal{D}$), or a regular party otherwise.

2: Let (b_i, v_i) be the output of protocol $\Pi_{\mathtt{DVSS}}$. If $b_i = 1$, abort protocol $\Pi_{\mathtt{VSS}}$ and output the share (b_i, v_i). If $b_i = 0$, run the remaining rounds of $\Pi_{\mathtt{VSS}}$ and let v'_i be its output. Output the share (b_i, v'_i).

Reconstruction Phase

Party P_i

1: Let (b_i, v_i) denote the share from the sharing phase. If $b_i = 1$, run the reconstruction phase of $\Pi_{\texttt{DVSS}}$ with input v_i and let s^* denote its output. If $b_i = 0$, run the reconstruction phase of $\Pi_{\texttt{VSS}}$ with input v_i and let s^* denote its output. Output the reconstructed secret s^*.

Theorem 1. *Let $\Pi_{\texttt{DVSS}}$ be a r-round (n, t, k) DVSS protocol and $\Pi_{\texttt{VSS}}$ a R-round (n, t) VSS protocol. Then $\Pi_{\texttt{OVSS}}$ is a (r, R) (n, t, k) OVSS protocol. Moreover:*

- *If $\Pi_{\texttt{DVSS}}$ and $\Pi_{\texttt{VSS}}$ have perfect (resp. statistical security), then $\Pi_{\texttt{OVSS}}$ has perfect (resp. statistical) security.*
- *If $\Pi_{\texttt{DVSS}}$ has perfect security and $\Pi_{\texttt{VSS}}$ has statistical security, then $\Pi_{\texttt{OVSS}}$ has perfect security against a k-adversary, and statistical security against a t-adversary.*

Proof. We first show that it is a VSS protocol. The Privacy, Correctness and Commitment properties follow directly from the corresponding properties of $\Pi_{\texttt{DVSS}}$ (which hold if the sharing succeeds) and of $\Pi_{\texttt{VSS}}$. Note that when there are k corruptions, every honest party outputs $b_i = 1$ and adopt the share from $\Pi_{\texttt{DVSS}}$, and therefore the security level of $\Pi_{\texttt{DVSS}}$ is inherited in this case.[1]

Since the sharing phase of $\Pi_{\texttt{VSS}}$ takes R rounds, the round complexity of $\Pi_{\texttt{OVSS}}$ is at most R. If the adversary is a k-adversary, by the Validity property of $\Pi_{\texttt{DVSS}}$ we will have for every honest party P_i that $b_i = 1$, and hence the sharing phase will end after r rounds, as desired. □

6 Perfect-security DVSS

VSS protocols have already been widely studied and round-optimal protocols are known for both perfect and statistical-security. Hence, in order to construct round-optimal OVSS protocols, our main focus will be on finding round-optimal DVSS protocols.

6.1 Honest-Dealer VSS

As a starting point, it will be easier to find protocols for the easier case of HD-DVSS. In the following we describe a HD-DVSS protocol, which is a variant of [33]:

Protocol $\Pi_{\texttt{HDDVSS}}$

Sharing Phase

Dealer D

1: Chooses a random bivariate polynomial $F \in \mathcal{K}[x, y]$, of degree t in each variable, such that $F(0, 0) = s$. Send to each party P_i the polynomials $f_i(x) = F(x, i)$ and $g_i(y) = F(i, y)$.

Party P_i

1: Send to each party P_j an independent random pad r_{ij} picked uniformly at random from $\mathcal{K}$.
2: Broadcast the following values:

- $a_{ij} = f_i(j) + r_{ij}$ (r_{ij} is the pad sent from P_i to P_j)
- $b_{ij} = g_i(j) + r_{ji}$ (r_{ji} is the pad P_i received from P_j).

Local computation:

[1] Note that in Definition 2, statistical security refers to the correctness and commitment properties only, and we require perfect privacy either way. This follows prior works (e.g. [52]).

1. Define the consistency graph $\mathcal{G} = (\mathcal{P}, E)$ on n vertices as follows: the edge $\{P_i, P_j\}$ is in E if $a_{ij} = b_{ji}$ and $a_{ji} = b_{ij}$.
2. Find a maximal matching M in $\overline{\mathcal{G}}$. Let N be the endpoints of edges in M, and let $C = \mathcal{P} \setminus N$.
3. Let $A = \{P_i \mid e_{\mathcal{G}}(P_i, C) \geq 2t + 1\}$. Set $S = C \cup A$.
4. If $|S| \geq n - k$, output $b_i = 1$ and $v_i = (S, f_i)$. Otherwise, output $b_i = 0$.

Reconstruction Phase

Party P_i

1: Send the value $s_i = f_i(0)$ to every other party P_j.

Local computation:

1. Denote the value received from party P_j by s_{ji}. Let $SH_i = \{(j, s_{ji})\}_{j \in S}$.
2. Let $s^* = \text{RS-EC}_t(SH_i)(0)$ be the reconstructed secret.

Theorem 2. *Protocol* $\Pi_{\texttt{HDDVSS}}$ *is a 2-round* (n, t, k) *HD-DVSS protocol with perfect security for* $n > 3t + k$*. Its communication complexity is* $O(n^2)$ *field elements.*

Proof. We say that two parties $\mathcal{P}_i$ and $\mathcal{P}_j$ are consistent with each other if the corresponding values of their polynomials match, i.e. if $f_i(j) = g_j(i)$ and $g_i(j) = f_j(i)$. We also say that a party $\mathcal{P}_i$ is consistent with bivariate polynomial $F(x, y)$ if $f_i(x)$ and $g_i(x)$ lie on $F(x, y)$, i.e. $f_i(x) = F(x, i)$ and $g_i(x) = F(i, x)$. Note that two honest P_i and P_j are consistent with each other iff $\{P_i, P_j\} \in E$.

We show the five required properties one by one:

- **Privacy:** The adversary $\mathcal{A}$ can learn at most t polynomials f_i, g_i. For any two honest parties P_i and P_j, the adversary has no information about the value r_{ij} and thus the broadcasts a_{ij} and b_{ij} provide no additional information. Since the polynomial F has degree t, the adversary learns no information about F, and thus no information about the secret s.

- **Agreement:** Follows directly from the fact that the decision only depends on broadcasted values, and by consistency of broadcast.

- **HD-Validity:** Note that since the dealer is honest, for any two honest parties P_i and P_j we have $\{P_i, P_j\} \in E$. Hence every edge in M has to contain at least one corrupted party and at most one honest party. It follows that $|M| \leq k$, and thus C contains at least $n - t - k \geq 2t + 1$ honest parties. Any honest party $P_i \in \mathcal{P} \setminus C$ will have an edge towards every honest party in C, and thus will be in the set A. It follows that the set S contains every honest party, and thus has size at least $n - k$, and the sharing will be accepted.

- **Commitment:** Since $|S| \geq n - t$, we must have $|C| \geq 2t + 1$. It follows that C contains at least $t + 1$ honest parties, and since C is a clique in $\mathcal{G}$ (M is a maximal matching), these honest parties are all consistent with each other and define a unique degree t two dimensional polynomial $F^*(x, y)$. Any honest party in A is by definition consistent with at least $t + 1$ honest parties in C, and thus is also consistent with polynomial $F^*(x, y)$. Hence every honest party in S is consistent with $F^*(x, y)$, and since $|S| \geq n - k \geq 3t + 1$, during reconstruction Reed-Solomon error correction will correctly recover the polynomial $F^*(0, y)$.

- **Correctness:** If the dealer $\mathcal{D}$ is honest, then the polynomial $F^*(x, y)$ that the honest parties uniquely determine is the dealer's polynomial $F(x, y)$, and hence the reconstructed secret $s^* = F(0, 0)$ is the dealer's value.

The dealer sends a total of $2n$ polynomials to other parties. A degree t polynomial can be uniquely determined using $t+1$ field elements (its coefficients), and hence the dealer sends a total of $O(nt) = O(n^2)$ field elements. Each party sends n field elements (the random pads) during the first round and broadcasts $2n$ values during the second round. Hence, they also communicate a total of $O(n^2)$ field elements. Thus, the total communication complexity is $O(n^2)$ field elements. □

6.2 From HDVSS to DVSS

We now proceed to describe $\Pi_{\texttt{DVSS}}$, a DVSS protocol that makes use of protocol $\Pi_{\texttt{HDDVSS}}$.

Protocol $\Pi_{\texttt{DVSS}}$

Sharing Phase

Dealer *D*

1: Chooses a random polynomial $f \in \mathcal{K}[x]$ of degree t, such that $f(0) = s$. Send to each party P_i the value $\alpha_i = f(i)$.

Party P_i

1: Initiate n instances Π_j of protocol $\Pi_{\texttt{HDDVSS}}$. In the i-th instance, participate as a dealer sharing a uniformly random value r_i from $\mathcal{K}$.
2: Broadcast the value $a_i = \alpha_i + r_i$.

Local computation:

1. Define the set S to be all parties whose sharing in the instance where they acted as the dealer has been accepted.
2. If $|S| \geq n - k$, output $b_i = 1$ and the share v_i consisting of S and the share v_{ij} from every instance Π_j for $P_j \in S$. Otherwise, output $b_i = 0$.

Reconstruction Phase

Party P_i

1: For every party $P_j \in S$, execute the reconstruction phase of protocol Π_j in order to reconstruct the values r_j.

Local computation:

1. Define the value $s_j = a_j - r_j$. Let $SH = \{(j, s_j)\}_{j \in S}$.
2. Let $s^* = \text{RS-EC}_t(SH)(0)$ be the reconstructed secret.

Theorem 3. *Protocol* $\Pi_{\texttt{DVSS}}$ *is a* 2*-round* (n, t, k) *DVSS protocol with perfect security for* $n > 3t+k$*. Its communication complexity is* $O(n^3)$ *field elements.*

Proof. We show the five required properties one by one:

- **Privacy:** The adversary $\mathcal{A}$ can learn at most t points α_i. For any honest party P_i, by privacy of $\Pi_{\texttt{HDDVSS}}$ the adversary has no information about the value r_i and thus the broadcast a_i provides no additional information. Since the polynomial f has degree t, the adversary learns no information about f, and thus no information about the secret s.
- **Agreement:** Follows directly from the agreement property of the $\Pi_{\texttt{HDDVSS}}$ protocol.
- **Validity:** By the HD-Validity of $\Pi_{\texttt{HDDVSS}}$, the sharing in instance Π_i succeeds for every honest party P_i. It follows that S contains every honest party and thus has size at least $n - k$.

- **Commitment:** By commitment of $\Pi_{\texttt{HDDVSS}}$, at the end of the sharing phase there are unique values r_i^* that will be reconstructed for all parties in S. Hence, there is also a unique set SH^* and a unique secret $s^* = \text{RS-EC}(SH^*)(0)$ that every honest party will reconstruct.
- **Correctness:** If the dealer $\mathcal{D}$ is honest, by correctness of $\Pi_{\texttt{HDDVSS}}$ and validity of broadcast we will have for every honest party P_i that $s_i = \alpha_i$. Hence, all but t points in SH will lie on f and we will have $s^* = \text{RS-EC}_t(SH)(0) = f(0) = s$.

The dealer sends one field element to each party, for a total of $O(n)$ field elements. Every party runs one instance of protocol $\Pi_{\texttt{HDDVSS}}$ which communicates $O(n^2)$ field elements, for a total of $O(n^3)$ field elements. Moreover, each party broadcasts one field elements for a total of $O(n)$ field elements. Thus, the total communication complexity is $O(n^3)$.

□

7 Statistical-security DVSS

In this section we will build a round optimal (n, t, k) DVSS protocol with statistical security for $n > 2t + k$.

7.1 Weak Secret Sharing

It will be easier to first build as a starting point a protocol with a slightly weaker commitment property, called weak secret sharing (WSS) [55]. While in regular secret sharing we are required by the Commitment property to reconstruct one specific secret, in a weak secret sharing we are additionally also allowed to "fail" and instead reconstruct $\perp$. If the reconstruction succeeds, the reconstructed secret still has to be uniquely determined.

More specifically, we have the following:

Definition 6. *We say that a two phase protocol is a (n, t) WSS protocol if for every t-adversary $\mathcal{A}$ the Privacy and Correctness properties hold as in Definition 2, and the $(1 - \varepsilon)$-Commitment property is replaced by:*

- *$(1 - \varepsilon)$-**Weak Commitment:** If the dealer D is corrupted, then with probability $1 - \varepsilon$ at the end of the sharing phase there is a unique value s^*, such that at the end of the reconstruction phase either every honest party P_i outputs $s'_i = s^*$, or every honest party outputs $\perp$.*

Similarly, we also get the corresponding definitions of DWSS and HD-DWSS.

7.2 Honest-Dealer VSS

We again start out by first building a HD-DVSS protocol. The protocol is the same as the one in Patra et al [52], but adapted to the definition of HD-DVSS and for a smaller number of parties. We reproduce the protocol and proofs here using our own notation and definitions.

Honest-Dealer WSS. The first step to obtain HD-DVSS is to construct a (n, t, k) HD-DWSS protocol for $n > 2t + k$.

We describe the protocol below:

Protocol $\Pi_{\texttt{HDDWSS}}$

Sharing Phase

Dealer D

1: Pick the following random values:

- A random bivariate polynomial $F \in \mathcal{K}[x, y]$, of degree t in the variable y and degree $n\kappa + 1$ in the variable x, such that $F(0, 0) = s$.
- n random polynomials $r_i \in \mathcal{K}[x]$ of degree $n\kappa + 1$.
- $n\kappa$ random non-zero distinct field elements $\alpha_{i,\ell}$ for $1 \leq i \leq n$ and $1 \leq \ell \leq \kappa$.

Send to each party P_i the following:

- The polynomials $f_i(x) = F(x, i)$ and $r_i(x)$.
- The random evaluation points $\alpha_{i,\ell}$ for $1 \leq \ell \leq \kappa$.
- The values $a_{j,i,\ell} = f_j(\alpha_{i,\ell})$ and $b_{j,i,\ell} = r_j(\alpha_{i,\ell})$ for $1 \leq \ell \leq \kappa$ and $1 \leq j \leq n$.

Denote all the values sent by the dealer to party i by s_i.

Party P_i

1: Do nothing.

2: Pick a random subset of indices $I_i \subseteq \{1, \ldots, \kappa\}$ of size $|I_i| = \frac{\kappa}{2}$. Broadcast the following values:

- A random non-zero value c_i and polynomial $g_i(x) = f_i(x) + c_i r_i(x)$.
- The set I_i.
- The evaluation points $\alpha_{i,\ell}$ and the values $a_{j,i,\ell}$ and $b_{j,i,\ell}$ for $1 \leq j \leq n$ and $\ell \in I_i$.

Local computation:

1. We say party P_i is accepted by party P_j if $a_{i,j,\ell} + c_i b_{i,j,\ell} = g_i(\alpha_{j,\ell})$ for all indices $\ell \in I_j$ broadcasted by party P_j.
2. Set $S = \{P_i \mid P_i \text{ is accepted by at least } 2t + 1 \text{ parties}\}$.
3. If $|S| \leq 2t$ reject the sharing. Otherwise, accept the sharing and output the share $v_i = (S, I_i, s_i)$.

Reconstruction Phase

Party P_i

1: Broadcast the polynomial f_i.

2: Broadcast all the evaluation points $\alpha_{i,\ell}$ and the values $a_{j,i,\ell}$ for $1 \leq j \leq n$ and $\ell \in \{1, \ldots, \kappa\} \setminus I_i$.

Local computation:

1. We say that party $P_i \in S$ is re-accepted by party $P_j \in \mathcal{P}$ if for any $\ell \in \{1, \ldots, \kappa\} \setminus I_j$, it holds that $a_{i,j,\ell} = f_i(\alpha_{j,\ell})$.
2. Let $S' = \{P_i \in S \mid P_i \text{ is re-accepted by at least } t + 1 \text{ parties}\}$. If the values $\{(i, f_i(0))\}_{P_i \in S'}$ interpolate to a degree t polynomial $g(y)$, output $s^* = g(0)$. Otherwise output $\perp$.

Theorem 4. *Protocol $\Pi_{\texttt{HDDWSS}}$ is a 2-round (n, t, k) HD-DWSS protocol with statistical security for $n > 2t + k$.*

Proof. We show the five required properties one by one:

- **Privacy:** The adversary $\mathcal{A}$ can learn at most t polynomials f_i, r_i. For the remaining polynomials f_j, r_j of honest parties, the adversary can learn at most κt points on them during

the first round (the points received from the dealer), and an additional $\frac{\kappa}{2}(n-t)$ points from the broadcasts in the second round, for a total of $\frac{n\kappa}{2} + \frac{\kappa t}{2} = \kappa(\frac{n}{2} + \frac{t}{2}) < n\kappa + 1$. Since the polynomials f_i and r_i have degree t, the adversary learns no information about them. Hence he know at most t points on $F(0, y)$, which implies he has no information about the secret $F(0, 0)$.

- **Agreement:** Follows directly from the fact that the decision only depends on broadcasted values, and by consistency of broadcast.

- **HD-Validity:** If the dealer is honest, clearly every honest party will be accepted by every other honest party. Since we have at most k corruptions, every honest party will be accepted by $n - k > 2t$ parties. Hence the set S will have size at least $n - k > 2t$ and the sharing will be accepted.

- $(1 - \varepsilon)$**-Weak Commitment:** We first show that with high probability every honest party that is in S will also be in S'. Suppose by contradiction that this is not the case. Hence, there is a honest party P_i that was accepted by at least $2t + 1$ parties but was re-accepted by at most t parties. Hence, there is at least one honest party P_j that accepted party P_i but did not re-accept it. This means that for all indices $\ell \in I_j$, we have $a_{i,j,\ell} + c_i b_{i,j,\ell} = g_i(\alpha_j, \ell)$ and for all indices $\ell \in \{1, \ldots, \kappa\} \setminus I_j$ we have $a_{i,j,\ell} \neq f_i(\alpha_{j,\ell})$. Since P_i chooses the value c_i independently from the values of the dealer and P_j picks the set I_j uniformly at random, the probability of this happening is $\frac{1}{\binom{\kappa}{\kappa/2}} = 2^{-\Omega(\kappa)}$.

  Next, consider the values $f_i(0)$ for all honest P_i in S (of which there are at least $t+1$). If the values interpolate to a degree t polynomial $g(y)$, let $s^* = g(0)$. Otherwise let $s^* = \perp$. Since by the previous part with high probability every honest part in S is also in S', it follows that we will reconstruct only s^* or $\perp$, as desired.

- $(1-\varepsilon)$**-Correctness:** We will show that with high probability if a corrupted party broadcasts a polynomial $f_i' \neq f_i$ different than the one he received from the dealer, then he will not be included in the set S'. The result then follows.

  If a party P_i is in the set S', it means that it was re-accepted by at least $t + 1$ parties and thus by at least 1 honest party. Note that the two degree $n\kappa + 1$ polynomials f_i and f_i' can have at most $n\kappa + 1$ points in common. Hence, since a corrupted party P_i does not know the evaluation points of a honest party P_j, the probability that it will be re-accepted by P_j is at most $\frac{nk}{|\mathcal{K}|} = \frac{nk}{2^\kappa}$. Hence, the probability that P_j is re-accepted is at most $\frac{nk(n-t)}{2^\kappa}$ and the probability that any of the corrupted parties gets re-accepted is $\frac{nk(n-t)t}{2^\kappa} = 2^{-\Omega(\kappa)}$.

□

From HD-DWSS to HD-DVSS. Note that the HD-DWSS protocol shown above can be also be used to share a degree t polynomial $g(y)$ rather than a secret value s, by picking a random two dimensional polynomial $F(x, y)$ such that $F(0, y) = g(y)$. In fact, this is how we will use the protocol in order to construct the HD-DVSS protocol below.

We again use a two dimensional degree t polynomial $F(x, y)$. This time, we use a symmetric polynomial so that the horizontal and vertical projections for each party are equal. Instead of using random pads for the consistency checks, we use random polynomials shared using the HD-DWSS protocol. This makes sure that the value provided by each party during reconstruction is uniquely determined at the end of the sharing phase; instead of having each party share their

polynomial during reconstruction we instead run the reconstruction protocol of HD-DWSS in order to reconstruct the random pad polynomials and then determine the actual polynomial from the broadcasted values:

Protocol $\Pi_{\texttt{HDDVSS}}$

Sharing Phase

Dealer D

1: Chooses a random symmetric bivariate polynomial $F \in \mathcal{K}[x, y]$ of degree t, such that $F(0, 0) = s$. Send to each party P_i the polynomial $f_i(y) = F(i, y)$.

Party P_i

1: Initiate n instances Π_j of protocol $\Pi_{\texttt{HDDWSS}}$. In the i-th instance, participate as a dealer sharing a uniformly random degree t polynomial g_i from $\mathcal{K}$.

2: Execute round 2 of every instance Π_i. Denote as S_i the set S outputted by instance Π_i. Broadcast the polynomial $h_i(y) = f_i(y) + g_i(y)$ and the values $a_{ji} = f_i(j) + g_j(i) = (f_j + g_j)(i)$ for $1 \leq j \leq n$.

Local computation:

1. We say that party P_i is accepted by party P_j if $h_i(j) = a_{ij}$.
2. Let $A_i = \{P_j . P_i \text{ is accepted by party } P_j\}$.
3. Set $S = \{P_i . |A_i| > 2t \text{ and the sharing in } \Pi_i \text{ has succeeded}\}$.
4. Remove P_i from S if $|S \cap A_i \cap S_i| \leq 2t$.
5. If $|S| \leq 2t$, reject the sharing. Otherwise, accept the sharing and output the share v_i consisting of S, the values h_i, and parties P_i's share in all instances Π_j that succeeded.

Reconstruction Phase

Party P_i

1: For every party $P_j \in S$, execute the reconstruction phase of protocol Π_j in order to reconstruct the values $g_j(y)$.

Local computation:

1. Let $S' = \{P_i \in S. \text{ the value reconstructed in } \Pi_i \text{ is not } \perp\}$.
2. If the values $\{(i, h_i(0) - g_i(0))\}_{P_i \in S'}$ interpolate to a degree t polynomial $f(x)$, output $s^* = f(0)$. Otherwise output 0.

Theorem 5. *Protocol $\Pi_{\texttt{HDDVSS}}$ is a 2-round (n, t, k) HD-DVSS protocol with statistical security for $n > 2t + k$.*

Proof. We show the five required properties one by one:

- **Privacy:** The adversary $\mathcal{A}$ can learn at most t polynomials f_i. By the Privacy property of the HD-DWSS protocol, the broadcasted values during the second round provide no additional information about the polynomials f_j of honest parties P_j. Hence the adversary learns at most t points on $F(0, y)$, which implies he has no information about the secret $F(0, 0)$.
- **Agreement:** Follows directly from consistency of broadcast and the Agreement property of $\Pi_{\texttt{HDDWSS}}$.
- **HD-Validity:** If the dealer is honest, clearly every honest party will be accepted by every other honest party. Moreover, for every honest party P_i, the set S_i will contain all honest

parties. By the HD-Validity property of $\Pi_{\texttt{HDDWSS}}$, all the sharings of honest parties will succeed. Thus, we have that the set S will contain every honest party and have size at least $n-k > 2t$, which implies that the sharing is accepted.

- $(1-\varepsilon)$**-Commitment:** Since the set S has size at least $2t+1$, it contains at least $t+1$ honest parties. Moreover, the reconstruction phase of Π_i will succeed for every honest party, hence every honest party in S will also be in S'. Consider the polynomials f_i corresponding to honest parties in S. If those polynomials do not interpolate to a degree t polynomial, then during reconstruction we will output 0.

  Thus suppose that all the polynomials f_i of honest parties in S lie on the same two dimensional polynomial $F^*(x,y)$. Fix any corrupted party P_i in S and let g_i denote the polynomial that it weakly secret shared and is committed to at the end of the sharing phase. We will show that with high probability the polynomial $h_i - g_i$ also lies on $F^*(x,y)$, which implies that during reconstruction we will uniquely reconstruct $F^*(0,0)$, provided that every corrupted party P_i in S' reconstructs the polynomial g_i that it is committed to, which by the $(1-\varepsilon)$-Weak Commitment property of $\Pi_{\texttt{HDDWSS}}$ happens with high probability.

  Since P_i is in S, we have that $|A_i \cap S_i| > 2t$. Hence, there are $t+1$ honest parties in $A_i \cap S_i$ which uniquely define the random pad polynomial g_i that P_i is committed to. Moreover, each such honest party P_j knows the value $f_j^*(i) = f_i^*(j) = F^*(i,j)$ and the value $g_i(j)$, and verified that $(f_i^* + g_i)(j) = h_i(j)$. Hence, we do indeed have that $h_i - g_i = f_i^*$.

- $(1-\varepsilon)$**-Correctness:** If the dealer is honest, the polynomials f_i of all honest parties in S lie on the dealer's polynomial $F(x,y)$. Hence, by the same reasoning as in the previous party, the dealer's secret $s = F(0,0)$ will be reconstructed.

□

7.3 From HD-DVSS to DVSS

We will now use the HD-DVSS protocol in order to construct a regular DVSS protocol. The idea is similar as in the perfect-security setting, where the dealer will share the secret with a one dimensional degree t polynomial f and send to each party one point on the polynomial, and then every party will initiate an instance of the HD-DVSS protocol in order to share a random pad and broadcast the padded value he received from the dealer. This ensures that a party is already committed in the sharing phase to what share they will provide during the reconstruction phase.

Nevertheless, a corrupted party is still able to commit on a share that is different than what it received from the dealer. This is not a problem in the perfect-security setting, as we have at least $3t+1$ parties during reconstruction, and thus enough parties to correct up to t errors. On the other hand, in the statistical setting, we are only guaranteed to have $2t+1$ parties during reconstruction. Hence we cannot perform error correction and need some way to ensure that corrupted parties are not able to provide a value that is different than the one they received from the dealer.

Information Checking. To achieve this, we will use a tool similar to the error checking vectors described in [55]. The idea is as follows: whenever the dealer sends a share s_i to party P_i, he additionally sends some authentication tag ϕ_i to party P_i and some verification tags σ_j to every other party P_j, such that when P_i reveals the value and the authentication tag he received from the dealer, the other parties will accept a value $s_i' \neq s_i$ only with negligible probability:

Definition 7. *Given a security parameter κ, a set of values $\mathcal{K}$ and sets of authentication tags $\mathcal{A}$ and verification tags $\mathcal{V}$, a pair of functions $Gen : \mathcal{K} \to \mathcal{A} \times \mathcal{V}$ and $Vfy : \mathcal{K} \times \mathcal{A} \times \mathcal{V} \to \{0,1\}$ are called information checking functions if the following properties hold:*

- ***Secrecy:*** *For any value v and $(\phi, \sigma) = Gen(v)$, the verification tag σ reveals no info about the value v, i.e. the distribution of σ is independent of the value v.*
- ***Correctness:*** *For any value v and $(\phi, \sigma) = Gen(v)$, we have $Vfy(v, \phi, \sigma) = 1$.*
- ***Unforgeability:*** *For any value v and $(\phi, \sigma) = Gen(v)$, without knowing σ the probability of finding a value $v' \neq v$ and an authentication tag ϕ' such that $Vfy(v', \phi', \sigma) = 1$ is at most ε, for $\varepsilon = 2^{-\Omega(\kappa)}$.*

We first provide a simple construction of information checking functions. Let $A = \mathcal{K}$ and $V = \mathcal{K} \times \mathcal{K}$. We define the two functions as follows:

- Gen(v): Choose two uniformly random values $b \neq 0$ and y from $\mathcal{K}$. Let $\phi = y$ and $\sigma = (b, v + by)$.
- Vfy(v, ϕ, σ): Let $\sigma = (b, c)$. If $v + b\phi = c$, output 1. Otherwise output 0.

Theorem 6. *The functions Gen and Vfy as described above are a pair of information checking functions.*

Proof. We show the three required properties one by one:

- **Secrecy:** Let $\sigma = (b, c)$. The value b is picked uniformly at random and is thus independent of v. The value $c = v + by$ is padded by the uniformly random value y, and thus is also independent from v.
- **Correctness:** Follows directly from the definition of the functions.
- **Unforgeability:** Let $(\phi, \sigma) = \text{Gen}(v)$ and $\sigma = (b, c)$. For any choice of v' and $\phi' \neq 0$ (the case where it is $\phi' = 0$ is straightforward), without knowing b the value of $v' + b\phi'$ is distributed uniformly at random, and hence will be equal to c with probability $\frac{1}{|\mathcal{K}|} = 2^{-\kappa}$.

□

Protocol Description. Next we provide the actual construction of the statistical DVSS protocol.

Protocol Π_{DVSS}

Sharing Phase

Dealer *D*

1: Chooses a random polynomial $f \in \mathcal{K}[x]$ of degree t, such that $f(0) = s$. Let $\alpha_i = f(i)$. For every pair of parties P_i and P_j, let $(\phi_{ij}, \sigma_{ij}) = \text{Gen}(\alpha_i)$. Send to each party P_i the value α_i and the values ϕ_{ij} and σ_{ji} for $1 \leq j \leq n$.

Party *P_i*

1: Initiate $(2n + 1)n$ instances Π_j of protocol Π_{HDDVSS}. Participate as the dealer in $2n + 1$ of them, sharing $2n + 1$ uniformly random values from $\mathcal{K}$. Denote the first $2n$ of these values by r_{bij} for $b \in \{0, 1\}$ and $1 \leq j \leq n$, and the final one by r_i.
2: Execute round 2 of every instance Π_j. Broadcast the following values:

- $a_i = \alpha_i + r_i$
- $b_{ij} = \phi_{ij} + r_{0ij}$ for $1 \leq j \leq n$
- $c_{ij} = \sigma_{ji} + r_{1ij}$ for $1 \leq j \leq n$.

Local computation:

1. Define the set S to be all parties whose all the sharings in which they acted as the dealer have been accepted.
2. If $|S| \geq n - k$, the sharing is accepted. Otherwise it is rejected.

Reconstruction Phase

Party P_i

1: For every party $P_j \in S$, execute the reconstruction phase of the protocols Π_j where P_j acted as the dealer in order to reconstruct the random pads.

Local computation:

1. For every party $P_i \in S$, reconstruct the values α_i, ϕ_{ij} and σ_{ij} using the broadcasted values a_i, b_{ij} and c_{ij} and the reconstructed random pads.
2. For any pair of parties P_i and P_j in S, we say that party P_i is accepted by party P_j if $\text{Vfy}(\alpha_i, \phi_{ij}, \sigma_{ij}) = 1$. Let $S' = \{P_i \in S.\ P_i \text{ is accepted by at least } t+1 \text{ parties}\}$. If the values $\{(i, \alpha_i)\}_{P_i \in S'}$ interpolate to a degree t polynomial $f(x)$, output $s^* = f(0)$. Otherwise output 0.

Theorem 7. *Protocol* $\Pi_{\texttt{DVSS}}$ *is a* 2*-round* (n, t, k) *DVSS protocol with statistical security for* $n > 2t + k$.

Proof. We show the five required properties one by one:

- **Privacy:** The adversary $\mathcal{A}$ can learn at most t points α_i. For any honest party P_i, by privacy of $\Pi_{\texttt{HDDVSS}}$ the adversary has no information about the value r_i and thus the broadcast a_i provides no additional information. Moreover, by Secrecy of the information checking functions the values σ_{ij} provide no information about α_i either. Since the polynomial f has degree t, the adversary learns no information about f (nor s).
- **Agreement:** Follows directly from the agreement property of the $\Pi_{\texttt{HDDVSS}}$ protocol.
- **Validity:** By the HD-Validity of $\Pi_{\texttt{HDDVSS}}$, the sharing in which P_i acts as a dealer succeed for every honest party P_i. It follows that S contains every honest party and thus has size at least $n - k$.
- **(1 - ε)-Commitment:** By (1 - ε)-Commitment of $\Pi_{\texttt{HDDVSS}}$, except for negligible probability at the end of the sharing phase there are unique values for the random pads that will be reconstructed for all parties in S. Hence, there are also unique values of α_i, ϕ_{ij} and σ_{ji} for all parties $P_i \in S$. Since the reconstructed secret is a deterministic function of these values, there is a uniquely determined reconstructed secret.
- **(1 - ε)-Correctness:** If the dealer $\mathcal{D}$ is honest, by correctness of $\Pi_{\texttt{HDDVSS}}$ and validity of broadcast the point (i, α_i) will lie on the dealer's polynomial for every honest party. We will show that with high probability if a corrupted party P_i is in S', then the point (i, α_i) also lies on the dealer's polynomial. This then implies that all the points $\{(i, \alpha_i)\}_{P_i \in S'}$ lie on the dealer's polynomial, and hence the correct secret is reconstructed.

Note that if a corrupt party P_i is in S', it was accepted by at least $t+1$ parties and thus by at least 1 honest party P_j. If the point corresponding to P_i does not lie on the dealer's polynomial, this implies that the adversary was able to find a value $\alpha'_i \neq \alpha_i$ and an authentication tag ϕ' such that $\text{Vfy}(\alpha'_i, \phi', \sigma_{ij}) = 1$, which contradicts unforgeability of information checking.

□

8 Lower Bounds

We show that our 2-round DVSS protocols have optimal resilience. Note that this implies optimality of their corresponding $(2,3)$ OVSS protocols as well, as any (r,R) OVSS protocol implies an r-round detectable VSS. We will in fact show a slightly stronger statement, namely for HD-DVSS protocols.

8.1 Perfect Security

We first make the simple well-known observation that we can assume private messages are exchanged only in the first round, since any private messages after the first round can be broadcast by masking it with one-time pads pairwise exchanged in the first round. This makes the subsequent analysis simpler:

Lemma 1. *For any r-round HD-DVSS protocol, there is an equivalent protocol where every message sent after the first round is a broadcast message.*

Proof. Consider any HD-DVSS protocol where private point-to-point messages are exchanged after the first round. We assume that each message is one field element. We build another protocol as follows: in the first round, in addition to the private messages dictated by the protocol, each party P_i sends to every other party P_j uniformly random pads $r_{ijk} \in \mathcal{F}$ for $2 \leq k \leq l$, where l is the number of private messages that P_i will send to P_j after the first round. In every round $r \geq 2$, any time a party P_i needs to send a private message m_{ijk} to party P_j, they instead broadcast the value $B_{ijk} = m_{ijk} + r_{ijk}$. Party P_j can then recover the message $m_{ijk} = B_{ijk} - r_{ijk}$, while for every other party the broadcast is uniformly random. It is easy to see that the resulting protocol still maintains privacy and is otherwise equivalent to the starting protocol. □

Theorem 8. *For any 2-round perfect-security (n,t,k) HD-DVSS protocol, we must have $n > 3t+k$.*

Proof. By the previous lemma we can assume that every message sent in the second round is a broadcast message. We proceed by contradiction. Assume that we have a 2-round (n,t,k) HD-DVSS protocol for $n = 3t+k$ parties. We partition the party set $\mathcal{P}$ into four sets A, B, M and K, the first three containing t parties each, and the final one containing the remaining k parties. For simplicity we consider an external dealer, the proof follows similarly with an internal dealer located in set A.

We now construct an attack against the protocol that breaks the commitment property. We use the following notation: for any two sets of parties S and R, let m_{SR} denote all the messages sent during the first round by any party in set S to any party in set R. The messages the dealer sends in round one to parties in S is denoted by s_S. Moreover, for any set S let B^1_S and B^2_S denote all the broadcasts done by parties in set S during the first and second round respectively. Similarly, the broadcasts performed by the dealer is denoted by B^1_D and B^2_D. We first consider an execution E_1

where an honest dealer is sharing a secret s and nobody is corrupted. The following table shows the views of the four blocks of parties at the end of the sharing phase:

	V_A	V_B	V_M	V_K
Sent	m_{AB} m_{AM} m_{AK}	m_{BA} m_{BM} m_{BK}	m_{MA} m_{MB} m_{MK}	m_{KA} m_{KB} m_{KM}
Received	s_A m_{BA} m_{MA} m_{KA}	s_B m_{AB} m_{MB} m_{KB}	s_M m_{AM} m_{BM} m_{KM}	s_K m_{AK} m_{BK} m_{MK}
Broadcasts	B_A^1 B_B^1 B_M^1 B_K^1 B_D^1 B_A^2 B_B^2 B_M^2 B_K^2 B_D^2			

Next we consider again an honest dealer sharing a secret $\overline{s} \neq s$, with no corruptions. By the Privacy property, there has to be one execution E_2 of the protocol where the joint view of all the parties in block M is identical to the one in E_1. If this was not the case, at the end of the sharing phase in E_1 an adversary corrupting block M would receive some information about the secret (namely that it is not $\overline{s}$). The following table shows the views of the four blocks of parties at the end of the sharing phase:

	V_A	V_B	V_M	V_K
Sent	$\overline{m_{AB}}$ m_{AM} $\overline{m_{AK}}$	$\overline{m_{BA}}$ m_{BM} $\overline{m_{BK}}$	m_{MA} m_{MB} m_{MK}	$\overline{m_{KA}}$ $\overline{m_{KB}}$ m_{KM}
Received	$\overline{s_A}$ $\overline{m_{BA}}$ m_{MA} $\overline{m_{KA}}$	$\overline{s_B}$ $\overline{m_{AB}}$ m_{MB} $\overline{m_{KB}}$	s_M m_{AM} m_{BM} m_{KM}	$\overline{s_K}$ $\overline{m_{AK}}$ $\overline{m_{BK}}$ m_{MK}
Broadcasts	B_A^1 B_B^1 B_M^1 B_K^1 B_D^1 B_A^2 B_B^2 B_M^2 B_K^2 B_D^2			

Next, we describe the attack on the protocol. The idea is as follows: the dealer and the parties in block A will be corrupted and act in such a way that parties in block B believe we are in execution E_1 but parties in K are corrupted, while parties in block K believe that we are in execution E_2 but parties in B are corrupted. Then, depending on how parties in A behave during the reconstruction, we will reconstruct either s or $\overline{s}$, breaking the Commitment property. More specifically, we consider an execution E_3 as follows: the dealer sends messages s_B to parties in B, s_M to parties in M, $\overline{s_K}$ to parties in K, and both s_A and $\overline{s_A}$ to parties in A. Parties in B have the same randomness as in execution E_1, while parties in K have the same randomness as in execution E_2. Parties in A act as follows: they send messages $m_{AB}, m_{AM}, \overline{m_{AK}}$ to blocks A, M and K respectively, and broadcast B_A^1 and B_A^2. The following table shows the views of parties in B, K and M at the end of the sharing phase:

	V_B	V_M	V_K
Sent	m_{BA} m_{BM} m_{BK}	m_{MA} m_{MB} m_{MK}	$\overline{m_{KA}}$ $\overline{m_{KB}}$ m_{KM}
Received	s_B m_{AB} m_{MB} $\overline{m_{KB}}$	s_M m_{AM} m_{BM} m_{KM}	$\overline{s_K}$ $\overline{m_{AK}}$ m_{BK} m_{MK}
Broadcasts	B_A^1 B_B^1 B_M^1 B_K^1 B_D^1 B_A^2 B'^2_B B_M^2 B'^2_K B_D^2		

Note that the second round broadcast B'^2_B done by parties in block B is different than the one done in execution E_1, as they have received a different message from parties in block K. The same holds for parties in block K. In order to break the commitment property, we first need to argue that the sharing does indeed get accepted:

Lemma 2. *The sharing in execution E_3 is accepted.*

Proof. We consider an execution E_4 that is identical to execution E_1 but where parties in block K are corrupted. They behave as follows: they send messages m_{KA}, $\overline{m_{KB}}$ and m_{KM} to parties in A, B and M respectively, and broadcast B_K^1 and B'^2_K. The following table shows the views of parties in block A, B and M at the end of the sharing phase:

	V_A	V_B	V_M
Sent	m_{AB} m_{AM} m_{AK}	m_{BA} m_{BM} m_{BK}	m_{MA} m_{MB} m_{MK}
Received	s_A m_{BA} m_{MA} m_{KA}	s_B m_{AB} m_{MB} $\overline{m_{KB}}$	s_M m_{AM} m_{BM} m_{KM}
Broadcasts	B_A^1 B_B^1 B_M^1 B_K^1 B_D^1 B_A^2 B'^2_B B_M^2 B'^2_K B_D^2		

Since the dealer is honest and only k parties are corrupted, by the HD-Validity property the sharing succeeds. Note that the view of parties in M is identical in executions E_4 and E_3. Hence parties in M will accept the sharing in execution E_3 as well. By the Agreement property, we have that every honest party accepts the sharing. □

We now show that two secrets can be reconstructed, depending on the set A.

Lemma 3. *There is a behaviour for parties in block A during the reconstruction phase of E_3 such that the secret s will be reconstructed.*

Proof. We consider the reconstruction phase of execution E_4, where during reconstruction parties in block K act as if their view is identical to the one at the end of execution E_3. By the Correctness property, the secret s will be reconstructed. Note that the view of parties in block B, M and K is identical at the start of the reconstruction phase of E_4 and E_3. Hence, if during the reconstruction

phase of E_3 parties in block A behave the same way as they would have in E_4, the secret s will be reconstructed. □

Lemma 4. *There is a behaviour for parties in block A during the reconstruction phase of E_3 such that the secret $\overline{s}$ will be reconstructed.*

Proof. We consider an execution E_5 that is identical to execution E_2 but where parties in block B are corrupted. They behave as follows: they send messages $\overline{m_{BA}}$, m_{BM} and m_{BK} to parties in A, M and K respectively, and broadcast B_B^1 and B'^2_B. The following table shows the views of parties in block A, M and K at the end of the sharing phase:

	V_A	V_M	V_K
Sent	$\overline{m_{AB}}$ m_{AM} $\overline{m_{AK}}$	m_{MA} m_{MB} m_{MK}	$\overline{m_{KA}}$ $\overline{m_{KB}}$ m_{KM}
Received	$\overline{s_A}$ $\overline{m_{BA}}$ m_{MA} $\overline{m_{KA}}$	s_M m_{AM} m_{BM} m_{KM}	$\overline{s_K}$ $\overline{m_{AK}}$ m_{BK} m_{MK}
Broadcasts	B_A^1 B_B^1 B_M^1 B_K^1 B_D^1 B_A^2 B'^2_B B_M^2 B'^2_K B_D^2		

Since the view of parties in M is identical in executions E_5 and E_3 and the sharing in E_3 is accepted, the sharing in E_5 is accepted as well. During reconstruction, parties in block B act as if their view is identical to the one at the end of execution E_3. By the Correctness property, the secret $\overline{s}$ will be reconstructed. Note that the view of parties in block B, M and K is identical at the start of the reconstruction phase of E_5 and E_3. Hence, if during the reconstruction phase of E_3 parties in block A behave the same way as they would have in E_5, the secret $\overline{s}$ will be reconstructed. □

By Lemma 3 and Lemma 4 we have shown the adversary can, depending on its behaviour, let two different secrets be reconstructed, thus breaking the Commitment property. □

8.2 Statistical security

We now proceed to the statistical security case.

Theorem 9. *For any 2-round statistical-security (n, t, k) HD-DVSS protocol, we must have $n > 2t + k$.*

Proof. We proceed by contradiction. Assume that we have a 2-round (n, t, k) HD-DVSS protocol for $n = 2t + k$ parties. We again partition the party set $\mathcal{P}$ into three sets A, B, and K, the first two containing t parties each, and the final one containing the remaining k parties. For simplicity we consider an external dealer, the proof follows similarly with an internal dealer located in set A. At a high level, we consider an execution where parties in set K receive no information either from the dealer or from parties in set A. Moreover, during the reconstruction phase parties in set A will not participate. Nevertheless, we will still need to be able to reconstruct the secret, which implies that parties in set B hold all the information needed to reconstruct the secret, thus breaking Secrecy.

More precisely, consider an execution E where the dealer D is corrupted and sharing the secret s and parties in set A are corrupted. They act as they would if they were honest, except for the

following difference: whenever the dealer would send a message to players in set K, he sends $\perp$ instead. Whenever a party in set A sends a message to a party in set K, he sends $\perp$ instead. Any party in set A or the dealer receive the first-round messages from K.

Also consider an alternative execution F where the dealer (sharing secret s) and parties in A and B are honest, but where parties in K are corrupted. Here, K behaves as follows: whenever they receive a message from a party in A or the dealer D, it replaces it with $\perp$. But K sends the first-round messages honestly. Whenever they should send a message to parties in A, they send $\perp$ instead.

Lemma 5. *The sharing in execution E is accepted.*

Proof. Note that execution E is identical to execution F.

Hence, by the HD-Validity property the execution gets accepted. □

□

Lemma 6. *If everyone behaves honestly during the reconstruction phase of E, we will with high probability reconstruct the secret s.*

Proof. Follows directly from the $(1-\varepsilon)$-Correctness property and the fact that execution E is identical F, where the dealer is honest and sharing s. □

Lemma 7. *If parties in A do not participate during the reconstruction phase of E, we will with high probability reconstruct the secret s.*

Proof. Follows directly from the $(1-\varepsilon)$-Commitment property and Lemma 6. □

Finally, consider an execution G where a honest dealer is sharing a secret s. We argue that at the end of the sharing phase, parties in B can simulate the reconstruction phase of execution E in order to reconstruct the secret s.

First note that the view of parties in B during execution E can be reproduced from the view of parties in B during execution G. To see this, it is enough to reproduce the view in F instead.

As in the perfect-security setting, assume without loss of generality that the only private-message round happens in the first round. First note that the first-round messages that B receives in F and G are the same. Moreover, in the second round broadcast message, the dealer or any party in A's message is also the same (since they accepted the messages from K in execution G), and thus this message has no additional information on the secret.

Now, observe that the parties in B during execution G can simulate the reconstruction process in E. To see this, first note that in execution they trivially know the messages that parties in K would send (since K only receives messages from B). They know the messages that parties in A would send (which are simply $\perp$ since they do not participate in the reconstruction phase), and they know the messages that parties in B would send, since they know all messages that parties in B have received from other parties, as argued in the previous paragraph.

Hence, they can simulate the reconstruction phase in order to reconstruct s, thus breaking the Privacy property of the protocol.

9 Future Work

We have settled the landscape of VSS protocols that incur two rounds under a sub-optimal corruption regime $f \leq k$, and three rounds when $k < f \leq t$ in both the perfect and statistical setting. Note that the computational setting is not interesting, since 2-round protocols exist with optimal resilience [6]. But several future lines of work remain. First, the DVSS protocol given in the statistical-security setting has two rounds of reconstruction rather than one round like in the perfect-security setting. In a recent work Applebaum et al. [3] have constructed the first round-optimal VSS protocol with three rounds of sharing and one round of reconstruction. Hence, a natural question is if those techniques could be adapted to our DVSS protocols in order to construct DVSS (and OVSS) protocols having only one round of reconstruction. Second, in the perfect security setting, we have provided one protocol which is efficient but has communication complexity $O(n^2)$, and one which is inefficient and has communication complexity $O(n^3)$. Ideally, we could construct a protocol which is both efficient and which has the lower communication complexity of $O(n^2)$. Finally, we have only adapted the protocols for the perfect-security setting to general adversaries. It would be interesting to find similar adaptations for the statistical-security setting.

References

[1] Ittai Abraham and Danny Dolev. Byzantine agreement with optimal early stopping, optimal resilience and polynomial complexity. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 605–614, New York, NY, USA, 2015. Association for Computing Machinery. doi:10.1145/2746539.2746581.

[2] Ittai Abraham, Kartik Nayak, Ling Ren, and Zhuolun Xiang. Good-case latency of byzantine broadcast: a complete categorization. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *40th ACM PODC*, pages 331–341. ACM, July 2021. doi:10.1145/3465084.3467899.

[3] Benny Applebaum, Eliran Kachlon, and Arpita Patra. The round complexity of statistical MPC with optimal resiliency. In Barna Saha and Rocco A. Servedio, editors, *55th ACM STOC*, pages 1527–1536. ACM Press, June 2023. doi:10.1145/3564246.3585228.

[4] Kaoru Kurosawa Ashish Choudhury and Arpita Patra. The round complexity of perfectly secure general vss. In *ICITS*, volume 6673 of *Lecture Notes in Computer Science*, pages 143–162. Springer, 2011.

[5] Renas Bacho, Daniel Collins, Chen-Da Liu-Zhang, and Julian Loss. Network-agnostic security comes (almost) for free in DKG and MPC. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 71–106. Springer, Cham, August 2023. doi:10.1007/978-3-031-38557-5_3.

[6] Michael Backes, Aniket Kate, and Arpita Patra. Computational verifiable secret sharing revisited. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 590–609. Springer, Berlin, Heidelberg, December 2011. doi:10.1007/978-3-642-25385-0_32.

[7] Akhil Bandarupalli, Xiaoyu Ji, Aniket Kate, Chen-Da Liu-Zhang, Daniel Pöllmann, and Yifan Song. Velox: Scalable fair asynchronous MPC from lightweight cryptography. In Chun-Ying

Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *ACM CCS 2025*, pages 1799–1813. ACM Press, October 2025. doi:10.1145/3719027.3765114.

[8] Akhil Bandarupalli, Xiaoyu Ji, Aniket Kate, Chen-Da Liu-Zhang, and Yifan Song. Computationally efficient asynchronous MPC with linear communication and low additive overhead. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part IV*, volume 16003 of *LNCS*, pages 261–294. Springer, Cham, August 2025. doi:10.1007/978-3-032-01884-7_9.

[9] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th ACM STOC*, pages 1–10. ACM Press, May 1988. doi:10.1145/62212.62213.

[10] Michael Ben-Or, Boaz Kelmer, and Tal Rabin. Asynchronous secure computations with optimal resilience (extended abstract). In Jim Anderson and Sam Toueg, editors, *13th ACM PODC*, pages 183–192. ACM, August 1994. doi:10.1145/197917.198088.

[11] Piotr Berman, Juan A Garay, and Kenneth J Perry. Optimal early stopping in distributed consensus. In *Distributed Algorithms: 6th International Workshop, WDAG'92 Haifa, Israel, November 2–4, 1992 Proceedings 6*, pages 221–237. Springer, 1992.

[12] Amey Bhangale, Chen-Da Liu-Zhang, Julian Loss, Kartik Nayak, and Sravya Yandamuri. Leader election with poly-logarithmic communication per party. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part II*, volume 16001 of *LNCS*, pages 37–68. Springer, Cham, August 2025. doi:10.1007/978-3-032-01878-6_2.

[13] Erica Blum, Chen-Da Liu Zhang, and Julian Loss. Always have a backup plan: Fully secure synchronous MPC with asynchronous fallback. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 707–731. Springer, Cham, August 2020. doi:10.1007/978-3-030-56880-1_25.

[14] Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OsDI*, volume 99, pages 173–186, 1999.

[15] David Chaum, Claude Crépeau, and Ivan Damgård. Multiparty unconditionally secure protocols (extended abstract). In *20th ACM STOC*, pages 11–19. ACM Press, May 1988. doi:10.1145/62212.62214.

[16] Annick Chopard, Martin Hirt, and Chen-Da Liu-Zhang. On communication-efficient asynchronous MPC with adaptive security. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 35–65. Springer, Cham, November 2021. doi:10.1007/978-3-030-90453-1_2.

[17] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *26th FOCS*, pages 383–395. IEEE Computer Society Press, October 1985. doi:10.1109/SFCS.1985.64.

[18] Brian A Coan. Efficient agreement using fault diagnosis. *Distributed computing*, 7:87–98, 1993.

[19] Shir Cohen, Idit Keidar, and Alexander Spiegelman. Make Every Word Count: Adaptive Byzantine Agreement with Fewer Words. In Eshcar Hillel, Roberto Palmieri, and Etienne Rivière, editors, *26th International Conference on Principles of Distributed Systems (OPODIS 2022)*, volume 253 of *Leibniz International Proceedings in Informatics (LIPIcs)*,

pages 18:1–18:21, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.OPODIS.2022.18, doi:10.4230/LIPIcs.OPODIS.2022.18.

[20] Bernardo David, Giovanni Deligios, Aarushi Goel, Yuval Ishai, Anders Konring, Eyal Kushilevitz, Chen-Da Liu-Zhang, and Varun Narayanan. Perfect MPC over layered graphs. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 360–392. Springer, Cham, August 2023. doi:10.1007/978-3-031-38557-5_12.

[21] Giovanni Deligios, Martin Hirt, and Chen-Da Liu-Zhang. Round-efficient byzantine agreement and multi-party computation with asynchronous fallback. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part I*, volume 13042 of *LNCS*, pages 623–653. Springer, Cham, November 2021. doi:10.1007/978-3-030-90459-3_21.

[22] Giovanni Deligios, Ivana Klasovita, and Chen-Da Liu-Zhang. Asymptotically optimal early termination for dishonest majority broadcast. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part V*, volume 15605 of *LNCS*, pages 116–144. Springer, Cham, May 2025. doi:10.1007/978-3-031-91092-0_5.

[23] Giovanni Deligios, Anders Konring, Chen-Da Liu-Zhang, and Varun Narayanan. Statistical layered MPC. In Elette Boyle and Mohammad Mahmoody, editors, *TCC 2024, Part IV*, volume 15367 of *LNCS*, pages 362–394. Springer, Cham, December 2024. doi:10.1007/978-3-031-78023-3_12.

[24] Gabriel Dettling, Chen-Da Liu-Zhang, Elisaweta Masserova, Matthieu Rambaud, and Antoine Urban. Broadcast for dynamic committees without trusted setup. *IACR Cryptol. ePrint Arch.*, page 2078, 2025. URL: https://eprint.iacr.org/2025/2078.

[25] Danny Dolev, Cynthia Dwork, Orli Waarts, and Moti Yung. Perfectly secure message transmission. *J. ACM*, 40(1):17–47, jan 1993. doi:10.1145/138027.138036.

[26] Danny Dolev, Ruediger Reischuk, and H Raymond Strong. 'eventual'is earlier than'immediate'. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 196–203. IEEE, 1982.

[27] Danny Dolev, Ruediger Reischuk, and H Raymond Strong. Early stopping in byzantine agreement. *Journal of the ACM (JACM)*, 37(4):720–741, 1990.

[28] Danny Dolev and H. Raymond Strong. Authenticated algorithms for byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

[29] Paul Feldman and Silvio Micali. Optimal algorithms for byzantine agreement. In *20th ACM STOC*, pages 148–161. ACM Press, May 1988. doi:10.1145/62212.62225.

[30] Matthias Fitzi, Juan Garay, Shyamnath Gollakota, C. Pandu Rangan, and Kannan Srinathan. Round-optimal and efficient verifiable secret sharing. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 329–342, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[31] Matthias Fitzi, Chen-Da Liu-Zhang, and Julian Loss. A new way to achieve round-efficient byzantine agreement. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *40th ACM PODC*, pages 355–362. ACM, July 2021. doi:10.1145/3465084.3467907.

[32] Juan A. Garay and Yoram Moses. Fully polynomial byzantine agreement for $n > 3t$ processors in $t + 1$ rounds. *SIAM Journal on Computing*, 27(1):247–290, 1998.

[33] Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. The round complexity of verifiable secret sharing and secure multicast. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 580–589, New York, NY, USA, 2001. Association for Computing Machinery. doi:10.1145/380752.380853.

[34] Diana Ghinea, Vipul Goyal, and Chen-Da Liu-Zhang. Round-optimal byzantine agreement. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part I*, volume 13275 of *LNCS*, pages 96–119. Springer, Cham, May / June 2022. doi:10.1007/978-3-031-06944-4_4.

[35] Diana Ghinea, Ivana Klasovitá, and Chen-Da Liu-Zhang. Round-optimal byzantine agreement without trusted setup. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 427–456. Springer, 2026.

[36] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987. doi:10.1145/28395.28420.

[37] Vipul Goyal, Chen-Da Liu-Zhang, and Rafail Ostrovsky. Asymmetric multi-party computation. In *4th Conference on Information-Theoretic Cryptography (ITC 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[38] Vipul Goyal, Chen-Da Liu-Zhang, and Yifan Song. Towards achieving asynchronous MPC with linear communication and optimal resilience. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VIII*, volume 14927 of *LNCS*, pages 170–206. Springer, Cham, August 2024. doi:10.1007/978-3-031-68397-8_6.

[39] Martin Hirt and Ueli M. Maurer. Player simulation and general adversary structures in perfect multiparty computation. *Journal of Cryptology*, 13(1):31–60, January 2000. doi:10.1007/s001459910003.

[40] Xiaoyu Ji, Chen-Da Liu-Zhang, Daniel Pöllmann, and Yifan Song. Information-theoretic network-agnostic mpc with polynomial communication. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 307–336. Springer, 2026.

[41] Xiaoyu Ji, Chen-Da Liu-Zhang, and Yifan Song. Perfectly secure network-agnostic mpc comes for free. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 337–365. Springer, 2026.

[42] Jonathan Katz and Chiu-Yuen Koo. On expected constant-round protocols for byzantine agreement. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 445–462. Springer, Berlin, Heidelberg, August 2006. doi:10.1007/11818175_27.

[43] Ranjit Kumaresan, Arpita Patra, and C. Pandu Rangan. The round complexity of verifiable secret sharing: The statistical case. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, pages 431–447, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[44] Chen-Da Liu-Zhang, Julian Loss, Ueli Maurer, Tal Moran, and Daniel Tschudi. MPC with synchronous security and asynchronous responsiveness. In Shiho Moriai and Huaxiong Wang,

editors, *ASIACRYPT 2020, Part III*, volume 12493 of *LNCS*, pages 92–119. Springer, Cham, December 2020. doi:10.1007/978-3-030-64840-4_4.

[45] Chen-Da Liu-Zhang, Elisaweta Masserova, João Ribeiro, Pratik Soni, and Sri Aravinda Krishnan Thyagarajan. Improved YOSO randomness generation with worst-case corruptions. In Jeremy Clark and Elaine Shi, editors, *Financial Cryptography and Data Security - 28th International Conference, FC 2024, Willemstad, Curaçao, March 4-8, 2024, Revised Selected Papers, Part II*, volume 14745 of *Lecture Notes in Computer Science*, pages 73–89. Springer, 2024. doi:10.1007/978-3-031-78679-2_4.

[46] Chen-Da Liu-Zhang, Elisaweta Masserova, João Ribeiro, Pratik Soni, and Sri Aravinda Krishnan Thyagarajan. Efficient distributed randomness generation from minimal assumptions where PArties speak sequentially once. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part V*, volume 15605 of *LNCS*, pages 176–206. Springer, Cham, May 2025. doi:10.1007/978-3-031-91092-0_7.

[47] Chen-Da Liu-Zhang, Elisaweta Masserova, João Ribeiro, and Sri AravindaKrishnan Thyagarajan. Game theory does not always help: The case of statistical multi-party coin tossing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 488–514. Springer, 2026.

[48] Julian Loss and Jesper Buus Nielsen. Early stopping for any number of corruptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part III*, volume 14653 of *LNCS*, pages 457–488. Springer, Cham, May 2024. doi:10.1007/978-3-031-58734-4_16.

[49] J-P Martin and Lorenzo Alvisi. Fast byzantine consensus. *IEEE Transactions on Dependable and Secure Computing*, 3(3):202–215, 2006.

[50] Silvio Micali. Very simple and efficient byzantine agreement. In Christos H. Papadimitriou, editor, *ITCS 2017*, volume 67, pages 6:1–6:1. LIPIcs, January 2017. doi:10.4230/LIPIcs.ITCS.2017.6.

[51] Rafael Pass and Elaine Shi. Thunderella: Blockchains with optimistic instant confirmation. In Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 3–33. Springer, Cham, April / May 2018. doi:10.1007/978-3-319-78375-8_1.

[52] Arpita Patra, Ashish Choudhary, Tal Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. Cryptology ePrint Archive, Paper 2008/172, 2008. https://eprint.iacr.org/2008/172. URL: https://eprint.iacr.org/2008/172.

[53] Arpita Patra, Ashish Choudhary, Tal Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 487–504. Springer, Berlin, Heidelberg, August 2009. doi:10.1007/978-3-642-03356-8_29.

[54] Kenneth J Perry and Sam Toueg. An authenticated byzantine generals algorithm with early stopping. Technical report, Cornell University, 1984.

[55] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 73–85, New York, NY, USA, 1989. Association for Computing Machinery. doi:10.1145/73007.73014.

[56] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In *21st ACM STOC*, pages 73–85. ACM Press, May 1989. doi:10.1145/73007.73014.

[57] Rüdiger Reischuk. A new solution for the byzantine generals problem. *Information and Control*, 64(1-3):23–42, 1985.

[58] Sam Toueg, Kenneth J Perry, and TK Srikanth. Fast distributed agreement. *SIAM Journal on Computing*, 16(3):445–457, 1987.

[59] Jun Wan, Hanshen Xiao, Elaine Shi, and Srinivas Devadas. Expected constant round byzantine broadcast under dishonest majority. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part I*, volume 12550 of *LNCS*, pages 381–411. Springer, Cham, November 2020. doi:10.1007/978-3-030-64375-1_14.

[60] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982. doi:10.1109/SFCS.1982.45.

Supplementary Material

A Perfect DVSS with $O(n^2)$ communication

Next, we build a DVSS protocol using a different approach, which will allow us to improve on the communication complexity of the previous protocol by a factor of n, at the cost of not being computationally efficient.

We again start by using the same approach of giving each party horizontal and vertical projections of a two dimensional polynomial $F(x, y)$. After doing the same mutual consistency checks between each pair of parties as in the HD-DVSS protocol, instead of disqualifying pairs of parties which disagree, we directly look for a set C of size at least $n - t$ where all the parties are mutually consistent. Note that if the dealer is honest, we will always find such a set (by simply taking the set of all honest parties). Hence, if we cannot find such a set we are guaranteed that the dealer is corrupted and can simply take a default sharing. Again, all the honest parties in the set C will define a unique two dimensional polynomial $F^*(x, y)$.

On the other hand, if the dealer is corrupted, even if we find such a set we are not guaranteed that all the honest parties will be in it, since the dealer could have just given a wrong polynomial to a honest party. This is a problem because in order to ensure that we have enough parties during reconstruction for error correction to work, all the honest parties need to have polynomials which lie on the same two dimensional polynomial.

The main idea is then to let parties outside of C reconstruct a polynomial which is consistent with those of parties in C. Let P_i be a party not in C. During the pairwise consistency checks, P_i has received from each honest party in C a point on the polynomial $F^*(x, i)$. Hence, P_i can look at all the points it received from parties in C and if all but at most k (from the corrupted parties) lie on the same polynomial, it can reconstruct the polynomial $F^*(x, i)$ and take that as its share.

We still need a way to ensure that every single party knows if P_i was able to recover its share or not, i.e. if all but at most k of the points it received from parties in C lie on the same polynomial. In order to do this, rather than letting the random pads used to broadcast the points in common during the consistency checks be independent from each other, we instead let each party pick a random polynomial r_i and let the random pads be points on the polynomial r_i.

The following is the protocol in detail:

Protocol Π_{DVSS}

Sharing Phase

Dealer D

1: Chooses a random bivariate polynomial $F \in \mathcal{K}[x, y]$, of degree t in each variable, such that $F(0, 0) = s$. Send to each party P_i the polynomials $f_i(x) = F(x, i)$ and $g_i(y) = F(i, y)$.

Party P_i

1: Pick a random univariate polynomial $r_i \in \mathcal{K}[x]$. Send to each party P_j the value $r_{ij} = r_i(j)$.
2: Broadcast the following values:

- $a_{ij} = f_i(j) + r_{ij}$ (r_{ij} is the pad sent from P_i to P_j)
- $b_{ij} = g_i(j) + r_{ji}$ (r_{ji} is the pad P_i received from P_j).

Local computation:

1. Define the consistency graph $\mathcal{G} = (\mathcal{P}, E)$ on n vertices as follows: the edge $\{P_i, P_j\}$ is in E if $a_{ij} = b_{ji}$ and $a_{ji} = b_{ij}$.

2. Find a maximum clique C in $\mathcal{G}$. If $|C| < n - t$, set $S = \mathcal{P}$, replace the polynomials f_i and g_i with the zero polynomials, and accept the sharing.
3. For each party P_j, let $V_j = \{(k, b_{kj})\}_{k \in C}$. We say that a party P_j in $\mathcal{P} \setminus C$ is recoverable if there is a degree t polynomial p_j such that all but k points in V_j lie on p_j. Let $A = \{P_j. \quad P_j \text{ is recoverable}\}$. Set $S = C \cup A$. If $\mathcal{P}_i \in A$, set $f_i = p_i - r_i$.
4. If $|S| \geq n - k$, output $b_i = 1$ and $v_i = (S, f_i)$. Otherwise, output $b_i = 0$.

Reconstruction Phase

Party P_i

1: Send the value $s_i = f_i(0)$ to every other party P_j.

Local computation:

1. Denote the value received from party P_j by s_{ji}. Let $SH_i = \{(j, s_{ji})\}_{j \in S}$.
2. Let $s^* = \text{RS-EC}_t(SH_i)(0)$ be the reconstructed secret.

Note that if the dealer is honest all honest parties will be part of a clique, hence if $C < n - t$, we know the dealer is corrupted and hence just accept the sharing as a default sharing of 0.

Theorem 10. *Protocol* $\Pi_{\texttt{HDDVSS}}$ *is a 2-round* (n, t, k) *HD-DVSS protocol with perfect security for* $n > 3t + k$*. Its communication complexity is* $O(n^2)$ *field elements.*

Proof. We first show some helpful properties:

Lemma 8. *If there are at most* k *corrupted parties, the set* S *contains every honest party.*

Proof. If $C < n-t$, this follows trivially. Otherwise, we have at least $n-2t \geq t+1$ honest parties in C which are mutually consistent with each other and thus define a unique degree t two dimensional polynomial $F^*(x, y)$. Fix a honest party P_j in $\mathcal{P} \setminus C$. For every honest party $P_k \in C$, we have

$$b_{kj} = g_k^*(j) + r_{jk} = F^*(k, j) + r_j(k) = f_j^*(k) + r_j(k) = (f_j^* + r_j)(k)$$

Hence the point (k, b_{kj}) lies on the polynomial $p_j = f_j^* + r_j$. It follows that the only points in V_j that do not lie on p_j are those from corrupted parties, which are at most k. Hence, P_j is recoverable. Hence, every honest party is either in C or is recoverable and thus is in A. □

Lemma 9. *All the polynomials* f_i *of honest parties in* S *lie on a unique two dimensional polynomial* F^**.*

Proof. If $C < n - t$, this follows trivially. Otherwise, as in the proof of the previous lemma, there is a polynomial $F^*(x, y)$ such that the polynomials f_i of honest parties in C lie on F^*. Recoverable honest parties set their polynomial to $f_i = p_i - r_i = f_i^* + r_i - r_i = f_i^*$, and hence their polynomial also lies on F^*. □

We show the five required properties one by one:

- **Privacy:** The adversary $\mathcal{A}$ can learn at most t polynomials f_i, g_i. For any two honest parties P_i and P_j, since the adversary knows at most t points on r_i it has no information about the value r_{ij} and thus the broadcasts a_{ij} and b_{ij} provide no additional information. Since the polynomial F has degree t, the adversary learns no information about F, and thus no information about the secret s.

- **Agreement:** Follows directly from the fact that the decision only depends on broadcasted values, and by consistency of broadcast.
- **Validity:** If $C \leq n-t$, the sharing trivially gets accepted. Otherwise, by Lemma 8 the set S contains every honest party, and thus has size at least $n-k$. Hence the sharing is accepted.
- **Commitment:** By Lemma 9, the polynomials f_i of all honest parties lie on F^*. Hence, the set SH_i will contain at least $n-k > 3t$ points, of which at most t do not lie on $F^*(0,y)$. Thus error collection uniquely recovers the polynomial $F^*(0,y)$ and the reconstructed secret $F^*(0,0)$ is unique.
- **Correctness:** If the dealer $\mathcal{D}$ is honest, then the polynomial $F^*(x,y)$ that the honest parties uniquely determine is the dealer's polynomial $F(x,y)$, and hence the reconstructed secret $s^* = F(0,0)$ is the dealer's value.

□

B DVSS for General Adversaries and Perfect Security

In the previous section we have seen how to construct DVSS protocols for threshold-based adversaries, which are allowed to corrupt parties up to a specific amount. We can generalize this and instead define an adversary by an adversarial structure which specifies which subsets of players it is allowed to corrupt:

Definition 8. *Given an adversarial structure $\mathcal{Z} \subseteq P(\mathcal{P})$, we say that an adversary A is a $\mathcal{Z}$-adversary if the set Z of parties he corrupts is in $\mathcal{Z}$.*

Note that a threshold based t-adversary is simply a special case of a $\mathcal{Z}$-adversary, where the set $\mathcal{Z}$ consists of all subsets of $\mathcal{P}$ of size at most t.

We now look at DVSS protocols for general adversaries. In addition to an adversarial structure $\mathcal{Z}$ which determines which sets of parties the adversary is allowed to corrupt, we will have an additional structure $\mathcal{Z}_e \subseteq \mathcal{Z}$ which determines which sets the adversary can corrupt such that we are guaranteed that the sharing succeeds. The definitions of (n,t) VSS, (n,t,k) OVSS and (n,t,k) DVSS protocol then easily generalize to $(n,\mathcal{Z})$ VSS, $(n,\mathcal{Z},\mathcal{Z}_e)$ OVSS and $(n,\mathcal{Z},\mathcal{Z}_e)$ DVSS.

B.1 The $\mathcal{Q}$ predicate

In the threshold adversary case, in order to prove the correctness of our protocols we needed some conditions relating the values n, t and k. Similarly, in the general adversary case we will need some specific conditions relating the sets in the adversarial structures $\mathcal{Z}$ and $\mathcal{Z}_e$ to the full party set $\mathcal{P}$. These conditions are expressed using the Q predicate, first introduced By Hirt and Maurer [39]:

Definition 9. *Let M denote a base set and let $\mathcal{A},\mathcal{B} \subseteq P(M)$. We say that the $Q^{(m,n)}(M,\mathcal{A},\mathcal{B})$ condition is satisfied if the following holds:*

$$\forall A_1 \ldots, A_m \in \mathcal{A} \quad \forall B_1, \ldots, B_n \in \mathcal{B} \quad M \not\subseteq \left(\bigcup_{i=1}^{m} A_i\right) \cup \left(\bigcup_{i=1}^{n} B_i\right)$$

B.2 General Adversary DVSS

In this section we build a perfect security 2 round DVSS for general adversaries. This can then be combined with already existing 3 round VSS protocols for general adversaries [4] in order to construct a $(2,3)$ round OVSS protocol.

We proceed similarly as in the threshold based adversary case, by first building a $2-$round perfect-security $(n, \mathcal{Z}, \mathcal{Z}_e)-$ HD-DVSS protocol.

The idea is as follows: instead of sharing the secret using polynomials, we instead share it using an additive sharing. Afterwards, for each possible set $Z \in \mathcal{Z}$ that the adversary is allowed to corrupt, we send one share to all parties in $\mathcal{P} \setminus Z$. This ensures Privacy of the protocol, as no matter what set the adversary corrupts, he will always miss one share.

Again, we need to perform consistency checks in order to make sure that all the parties in $\mathcal{P} \setminus Z$ have received the same share in case that the dealer is corrupted. We use the same method of sharing random pads during the first round and then having players broadcast the padded share they received from the dealer. If there is a disagreement between two parties either one of them is corrupted or the dealer is corrupted. Thus, for each share if we can find a set $Z \in \mathcal{Z}_e$ such that all the disagreements could have been caused by those players, we simply disqualify them for that specific share. Then, we still have enough parties during reconstruction such that we can perform simple error correction and reconstruct all the additive shares correctly:

Protocol $\Pi_{\texttt{HDDVSS}}$

Sharing Phase

Dealer D

1: Let $\mathcal{Z} = \{Z_1, \ldots, Z_{|\mathcal{Z}|}\}$. Additively share the secret s into uniformly random shares s_i such that $s = \sum_{k=1}^{|\mathcal{Z}|} s_k$. Send s_k to every player in $\mathcal{P} \setminus Z_k$.

Party P_i

1: For every $1 \leq k \leq |\mathcal{Z}|$ and every $j > i$ such that $P_j \in \mathcal{P} \setminus Z_k$: send a uniformly random value r_{kij} to party P_j.

2: For every $1 \leq k \leq |\mathcal{Z}|$ and every $P_j \in \mathcal{P} \setminus Z_k$: if $j > i$, let $r' = r_{kij}$, otherwise let $r' = r_{kji}$. Broadcast the value $s_{kij} = s_k + r'$.

Local computation: For every $1 \leq k \leq |\mathcal{Z}|$, build an auxiliary graph G_k with vertex set $\mathcal{P} \setminus Z_k$, and add en edge between player P_i and P_j if $s_{kij} = s_{kji}$. If there exists a set $Z_e \in \mathcal{Z}_e$ such that the vertices $\mathcal{P} \setminus Z_k \setminus Z_e$ form a clique in G_k, set $S_k = \mathcal{P} \setminus Z_k \setminus Z_e$. If there is no such set, the sharing has failed.

Reconstruction Phase

Party P_i

1: For every $1 \leq k \leq |\mathcal{Z}|$: if $P_i \in S_k$, send the value s_k to every other player P_j.

Local computation:

1. For every $1 \leq k \leq |\mathcal{Z}|$: denote the value obtained for share s_k from player $P_j \in S_k$ by s'_{kj}. If there exists a set $Z \in \mathcal{Z}$ such that all the values s'_{kj} for $P_j \in S_k \setminus \mathcal{Z}$ are equal, set s^*_k to be that value. Otherwise set $s^*_k = 0$.
2. Set $s^* = \sum_{i=1}^{|\mathcal{Z}|} s^*_k$.

Lemma 10. *Protocol $\Pi_{\texttt{HDDVSS}}$ is a perfectly secure $(n, \mathcal{Z}, \mathcal{Z}_e)-$ HD-DVSS protocol if the $Q^{(3,1)}(\mathcal{P}, \mathcal{Z}, \mathcal{Z}_e)$ condition is satisfied.*

Proof. We prove the five required properties:

- **Privacy:** The values broadcasted in the second round are all obfuscated by uniformly random pads and hence are distributed independently from the secret. For any set $Z_k \in \mathcal{Z}_p$ that can be corrupted, the adversary receives no information about the share s_k and hence no information about the secret s.
- **Agreement:** The decision on the success of the sharing is made solely based on broadcasted values, hence every honest party will make the same decision.
- **Validity:** Suppose the dealer is honest and the dealer corrupts a set $Z_e \in \mathcal{Z}_e$. Hence all players in $\mathcal{P} \setminus Z_k \setminus Z_e$ will have received the same value s_k from the dealer, and since they behave according to protocol we also have that $s_{kij} = s_{kji}$ for any $P_i, P_j \in \mathcal{P} \setminus Z_k \setminus Z_e$. Hence the set $\mathcal{P} \setminus Z_k \setminus Z_e$ will always form a clique in G_k and the sharing will succeed, as required.
- **Commitment:** First note that since the vertices S_k form a clique in G_k, all honest parties in it have received the same value from the dealer. Denote this value by s_k^*. Denote the set of corrupted players by Z^*. It follows that the values s'_{kj} for $P_j \in S_k \setminus Z^*$ are all equal to s_k^*. Now suppose that there is another set $Z_2^* \in \mathcal{Z}$ such that the values s'_{kj} for $P_j \in S_k \setminus Z_2^*$ are all equal to a different value s_{k2}^*. Since the honest players received s_k^*, it follows that $S_k \setminus Z_2^* \in \mathcal{Z}$. But we then have that

  $$\mathcal{P} \subseteq (S_k / Z_2^*) \cup Z_2^* \cup Z_e \cup Z_k$$

  contradicting the $Q^{(3,1)}(\mathcal{P}, \mathcal{Z}, \mathcal{Z}_e)$ condition.

  Hence, we will uniquely reconstruct the secret $s^* = \sum_{i=1}^{|\mathcal{Z}|} s_k^*$.
- **Correctness:** If the dealer D is honest the values s_k^* are an additive sharing of the dealer's secret s, and thus we will correctly reconstruct the dealer's secret.

□

Next, we use protocol $\Pi_{\texttt{HDDVSS}}$ in order to construct a DVSS protocol. The idea is similar to the threshold adversary case: we use $\Pi_{\texttt{HDDVSS}}$ in order to make sure that at the end of the sharing phase every party is either disqualified or has a unique value that they will be able to provide as their share during reconstruction. Instead of the polynomial Shamir sharing, we use an additive Shamir sharing as in $\Pi_{\texttt{HDDVSS}}$:

Protocol $\Pi_{\texttt{DVSS}}$

Sharing Phase

Dealer D

1: Let $\mathcal{Z} = \{Z_1, \ldots, Z_{|\mathcal{Z}|}\}$. Additively share the secret s into uniformly random shares s_i such that $s = \sum_{k=1}^{|\mathcal{Z}|} s_k$. Send s_k to every player in $\mathcal{P} \setminus Z_k$.

Party P_i

1: For every party $P_j \in \mathcal{P}$ and set $Z_k \in \mathcal{Z}$, initiate an instance Π_{jk} of protocol $\Pi_{\texttt{HDDVSS}}$. Execute round 1 of every instance Π_{jk}, acting as a dealer in instance Π_{ik} sharing a uniformly random value r_{ik} from $\mathcal{K}$.

2: Execute round 2 of every instance Π_{jk}. For every set Z_k such that $P_i \in Z_k$, broadcast the value $a_{ik} = s_k + r_{ik}$.

Local computation:

1. Define the set S to be all parties P_j such that all the sharings in instances Π_{jk} have succeeded.
2. If there exists a set $Z_e \in \mathcal{Z}_e$ such that $S = \mathcal{P} \setminus Z_e$, the sharing is accepted. Otherwise it is rejected.

Reconstruction Phase

Party P_i

1: For every party $P_j \in S$, execute the reconstruction phase of instance Π_{jk} in order to reconstruct the values r_{jk}.

Local computation:

1. For $P_j \in S$, let $s'_{jk} = a_{jk} - r_{jk}$. If there exists a set $Z \in \mathcal{Z}$ such that all the values s'_{jk} for $P_j \in S \setminus Z_k \setminus Z$ are equal, set s^*_k to be that value. Otherwise set $s^*_k = 0$.
2. Set $s^* = \sum_{i=1}^{|\mathcal{Z}|} s^*_k$.

Theorem 11. *Protocol* $\Pi_{\texttt{DVSS}}$ *is a* 2*-round* $(n, \mathcal{Z}, \mathcal{Z}_e)$ *DVSS protocol if the* $Q^{(3,1)}(\mathcal{P}, \mathcal{Z}, \mathcal{Z}_e)$ *condition is satisfied.*

Proof. We show the five required properties one by one:

- **Privacy:** The values broadcasted in the second round are all obfuscated by uniformly random pads and hence, by Privacy of $\Pi_{\texttt{HDDVSS}}$, are distributed independently from the secret. For any set $Z_k \in \mathcal{Z}$ that can be corrupted, the adversary receives no information about the share s_k and hence no information about the secret s.
- **Agreement:** Follows directly from the agreement property of the $\Pi_{\texttt{HDDVSS}}$ protocol.
- **Validity:** By the HD-Validity of $\Pi_{\texttt{HD-DVSS}}$, the sharing in instance Π_{ik} succeeds for every honest party P_i. It follows that S contains every honest party and thus has size at least $n-k$.
- **Commitment:** By commitment of $\Pi_{\texttt{HD-DVSS}}$, at the end of the sharing phase there are unique values r^*_{ik} that will be reconstructed for all parties in S. Hence, the values s^*_k and thus the reconstructed secret s^* are fixed at the end of the sharing phase.
- **Correctness:** If the dealer $\mathcal{D}$ is honest, for a fixed k we have for honest parties P_i that $s'_{ik} = s_k$. Hence, if the set of corrupted players is Z^*, we have for all $P_j \in S \setminus Z_k \setminus Z^*$ that $s'_{jk} = s_k$. Moreover, suppose that there is another set $Z^*_2 \in \mathcal{Z}$ such that the values s'_{jk} for $P_j \in S \setminus Z_k \setminus Z^*_2$ are all equal to a different value $s'_k \neq s_k$. It follows that $S \setminus Z_k \setminus Z^*_2 \in \mathcal{Z}$. But we then have that
$$\mathcal{P} \subseteq (S \setminus Z_k \setminus Z^*_2) \cup Z^*_2 \cup Z_e \cup Z_k$$
contradicting the $Q^{(3,1)}(\mathcal{Z}, \mathcal{Z}_e)$ condition.

□