Broadcast (and Round) Efficient Verifiable Secret Sharing[*]

JUAN GARAY[†] CLINT GIVENS [‡] RAFAIL OSTROVSKY[§] PAVEL RAYKOV[¶]

September 16, 2013

Abstract

Verifiable secret sharing (VSS) is a fundamental cryptographic primitive, lying at the core of secure multi-party computation (MPC) and, as the distributed analogue of a commitment functionality, used in numerous applications. In this paper we focus on unconditionally secure VSS protocols with honest majority.

In this setting it is typically assumed that parties are connected pairwise by authenticated, private channels, and that in addition they have access to a "broadcast" channel. Because broadcast *cannot* be simulated on a point-to-point network when a third or more of the parties are corrupt, it is impossible to construct VSS (and more generally, MPC) protocols in this setting without using a broadcast channel (or some equivalent addition to the model).

A great deal of research has focused on increasing the efficiency of VSS, primarily in terms of round complexity. In this work we consider a refinement of the round complexity of VSS, by adding a measure we term *broadcast complexity*. We view the broadcast channel as an expensive resource and seek to minimize the number of rounds in which it is invoked as well.

We construct a (linear) VSS protocol which uses the broadcast channel only *twice* in the sharing phase, while running in an overall constant number of rounds.

1 Introduction

Verifiable secret sharing (VSS) [CGMA85], where a *dealer* wishes to share a secret among a group of n parties, at most t of which (possibly including the dealer) may be actively malicious, is a fundamental cryptographic primitive, lying at the core of secure multi-party computation (MPC) [GMW87, BGW88, CCD88] and used in a myriad of applications. Our focus in this paper is on *unconditionally* (a.k.a. *information-theoretically*) secure VSS protocols (meaning that the security properties are guaranteed to hold even when the malicious parties are endowed with unbounded computational power) with honest majority ($t < n/2$).

In the unconditional setting, it is typically assumed that parties are connected pairwise by authenticated, private channels, and that in addition they have access to a "broadcast" channel. Broadcast allows one party to send a consistent message to all other parties, guaranteeing consistency even if the broadcaster is corrupted. Because broadcast cannot be simulated on a point-to-point network when more than a third of the parties are corrupt [LSP82], even probabilistically, it is impossible to construct VSS (or more generally, MPC) protocols in this setting without using a "physical broadcast channel" (that is, a black-box which securely implements broadcast), or some equivalent addition to the model.

[*]An abridged version of this paper appears in *Proc. 7th International Conference on Information-Theoretic Security — ICITS 2013.*

[†]AT&T Labs – Research. Email: `juan.a.garay@gmail.com`.

[‡]Maine School of Science and Mathematics. Email: `cgivens@gmail.com`.

[§]Department of Computer Science and Department of Mathematics, UCLA. Email: `rafail@cs.ucla.edu` Supported in part by NSF grants 0830803, 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

[¶]ETH Zürich. Email: `pavel.raykov@inf.ethz.ch`.

Further, it is known that in this regime ($n/3 \leq t < n/2$), protocols are subject to some (negligibly small) error probability and cannot achieve so-called perfect security [CCD88, RB89, DDWY93], which is possible when $t < n/3$.
A great deal of research has focused on understanding the complexity as well as increasing the efficiency of VSS, primarily in terms of round complexity [GIKR02, FGG$^+$06, KKK08, PCRR09, KPC10]. Indeed, given its typical applications, such as implementing a pre-processing phase, as well as the *share* phase in the general "share-compute-reveal" shape of an MPC protocol [GMW87], or its use during the *setup* phase of information-theoretic protocols when $t \geq n/3$ (e.g., [PW96, BTHR07, HR13]; see Related work), a fast execution—namely, a (small) constant number of rounds (some specific figures given later on)—is of utmost importance.
In this work we consider a refinement of the round complexity of VSS, by incorporating an additional measure which we term *broadcast complexity.* We view the broadcast channel as an expensive resource and seek to minimize the number of rounds in which it is invoked as well. Justifiably so, high-level descriptions of VSS (and, more generally, MPC) protocols tend to treat broadcast as a black-box. When $t < n/3$, this may be viewed simply as a convenient abstraction, since broadcast in any case can be simulated in a point-to-point network using Byzantine agreement[1].
However, when $t < n/2$, the black-box treatment of broadcast is (as described above) no longer a convenience but a **requirement**, and there are compelling reasons to consider it more expensive than "mere" secure channels. Indeed, while the latter can be realized for example via the physical exchange (using trusted couriers) of large one-time pads between every pair of players, which may be done in an asynchronous preprocessing phase and without any centrally trusted party, we see no equally straightforward approach to physically implement *broadcast* without a trusted party, and when the participants are geographically scattered. Hence it is only natural to treat physical broadcast as an expensive resource, and in particular to treat a protocol's *broadcast rounds* as (substantially) more expensive than ordinary rounds. In addition, the question of how many broadcast rounds does VSS require in the $t < n/2$ regime is compelling from a theoretical perspective.

Our results. Thus motivated to better understand the broadcast requirements of verifiable secret sharing when $t < n/2$, in this work we present new upper bounds on its broadcast and round complexity. Specifically, we show a constant-round, linear VSS protocol which only uses *two* broadcasts in the sharing phase and none in reconstruction—what we call a $(2, 0)$-broadcast VSS. The overall number of rounds is $(20, 1)$, again meaning 20 rounds in the sharing phase and 1 reconstruction round.
To our knowledge, the most efficient VSS protocol in terms of broadcast rounds for $t < n/2$ is the $(2, 2)$-broadcast, $(3, 2)$-round protocol of Kumaresan *et al.* [KPC10], which is exponential-time and not (apparently) linear. The same authors also give a $(3, 2)$-broadcast, $(4, 2)$-round VSS which is polynomial-time and linear (we believe—though the authors do not claim it here either), at the expense of an additional round in the sharing phase. Hence our $(2, 0)$-broadcast protocol improves the overall broadcast complexity (although it is not as round-efficient).
Considering linear, constant-round protocols which use zero broadcasts during reconstruction (which are more suitable for VSS applications such as [broadcast-efficient] MPC), the most broadcast-efficient VSS protocol we are aware of is the $(7, 0)$-broadcast protocol described in [RB89, Rab94]. Recently, Hirt and Raykov [HR13] presented an approach allowing to construct $(1, 0)$-broadcast VSS protocols for $t < n/2$, but the overall number of protocol rounds is *linear* in n, making it not ideally fit for the natural applications of VSS mentioned above.
We derive our $(2, 0)$-broadcast, constant-round VSS protocol in two stages.

1. In the first stage, we obtain a $(3, 0)$-broadcast, constant-round protocol which is inspired by the protocol in [Rab94], but leverages a number of novelties and optimizations to reduce the broadcast complexity from 7 to 3; its overall round complexity is $(9, 1)$. This is presented in Section 3.1.
2. In the second stage, we apply a transformation to the sharing phase of the above protocol such that it uses *two* rounds of broadcast instead of three. This optimization is in turn inspired by the one

[1]Trouble comes, however, when analyzing round complexity: as observed in [KK07, Koo07, KKK08], Byzantine agreement is round-expensive, and the compilation from black-box broadcast to simulated broadcast blows up the number of rounds substantially.

presented in [KK07], and the key method is what the authors called *moderated* protocols. This method is a generic transformation which given any protocol Π employing broadcast channels, constructs a "moderated" version Π' of Π where all calls to broadcast channels are substituted with a special broadcast simulation subroutine controlled by a designated party (called *the moderator*). A new key technical element in our construction is to show how using one round of physical broadcast (the first one in our stage-1 protocol) one can prepare a *setup* which allows to invoke sufficiently many broadcast simulation routines later on. This transformation yielding the final VSS protocol is presented in Section 3.2.

As our focus in this work is on reducing the overall number of broadcast *rounds*, rather than broadcast (or otherwise) *communication complexity*, we forgo explicit treatment of the latter. We do however note that protocols described herein can be compiled via generic techniques into significantly more communication-efficient versions; see work of Fitzi and Hirt [FH06], as well as recent work by Ben-Sasson *et al.* [BFO12].

Related work. We already mentioned above the most closely related work regarding unconditionally secure VSS for $t < n/2$ [Rab94, KPC10]. The role of broadcast in multiparty protocols has been studied in a number of other previous works. Katz *et al.* [KKK08, KK07, Koo07], seeking to improve overall round complexity when broadcast is simulated over point-to-point channels, construct constant-round protocols for VSS and MPC whose descriptions use only a single broadcast round. However, for $t < n/2$ they assume a PKI infrastructure (e.g., pseudosignatures [PW96]—more on this below) is already in place.

Fitzi *et al.* [FGMR02, FGH$^+$02], as well as Goldwasser and Lindell [GL05], consider broadcast and MPC protocols for $t < n$ which do not use physical broadcast at all (nor equivalent assumptions), but instead weaken the guarantees provided by the protocol. In particular these protocols are not robust and so may fail to deliver any output at all. On the other hand, the so-called *detectable broadcast* (and *detectable MPC*) protocols of [FGMR02, FGH$^+$02] do achieve consistency among honest players: either the broadcast (MPC) succeeds and all honest parties receive output, or it fails, in which case all honest parties agree that it failed.

As mentioned above, unconditionally secure broadcast cannot be simulated on a point-to-point network when more than a third of the parties are corrupt. However, if there is a setup phase during which the parties enjoy access to a physical broadcast channel (but need not know their future inputs), Pfitzmann and Waidner [PW96] showed how to construct *pseudosignatures*, an information-theoretic authentication technique for multiparty protocols which can then be used to simulate future invocations of broadcast by running a so-called "authenticated" Byzantine agreement protocol [DS83]; this avoids any need for a physical broadcast channel during the main phase of the protocol. The number of broadcast rounds in the [PW96] setup construction is $O(n^2)$, and it works for an arbitrary number of corruptions ($t < n$). This was later improved to $O(n)$ broadcast rounds by Beerliová-Trubíniová *et al.* in [BTHR07], at the price of tolerating $t < n/2$ corruptions, and recently by Hirt and Raykov to just 1, as mentioned above. However, the overall round complexity of this construction (as well as that in [BTHR07]) is $O(n)$.

Minimizing the use of broadcast has also been considered in the related problem known as *secure message transmission by public discussion* (SMT-PD), where a Sender wants to send a message to a Receiver privately and reliably. Recall that in this problem, Sender and Receiver are connected by n channels, up to $t < n$ of which may be maliciously controlled by a computationally unbounded adversary, as well as one public channel, which is reliable but not private. SMT-PD was introduced in [GO08] as an important building block for achieving unconditionally secure MPC on sparse (i.e., not fully connected) networks. The motivation for this abstraction comes from the feasibility in partially connected settings for a subset of the nodes in the network to realize a broadcast functionality despite the limited connectivity [DPPU86, Upf92], which plays the role of the public channel. Such implementation of the public channel on point-to-point networks is costly and highly non-trivial in terms of rounds of computation and communication, as mentioned earlier. See, e.g., [GGO11] for further details.

We now turn to the presentation of the model, definitions and building blocks, followed by the new

VSS protocol (Section 3). Some of the auxiliary constructions and proofs appear in the appendix.

2 Model, Definitions and Tools

We consider a complete, synchronous network of n players $P_1, \dots, P_n$ who are pairwise connected by secure (private and authenticated) channels, and who additionally have access to a broadcast channel. Some of these players are corrupted by a centralized adversary $\mathcal{A}$ with *unbounded computing power*. The adversary is *active*, directing players under his control to deviate from the protocol in arbitrary ways. As noted, we consider only *static* rather than adaptive adversary in this work, meaning that he chooses which players to corrupt prior to the start of protocol execution. The computation evolves as a series of rounds. In a given round, honest players' messages depend only on information available to them from prior rounds; $\mathcal{A}$, however, is *rushing*, and receives all messages (and broadcasts) sent by honest players before deciding on the messages (and broadcasts) of corrupted players. Sometimes we refer to $\mathcal{A}$ thus defined as a *t-adversary.* We consider statistical security (since, as mentioned above, perfect security is unachievable in this setting), and let κ denote the error parameter, $\kappa \geq 2n$.

Information checking. An *information checking scheme (IC)* [RB89] is a triple of protocols (ICSetup, ICValidate, ICReveal) which achieves a limited signature-like functionality for three players: a *dealer* D, *intermediary* I, and *receiver* R. D holds as input a secret $s \in \mathbb{F}$, which he passes to I in ICSetup. ICValidate insures that even if D cheats, I knows a value which R will accept. In ICReveal, I sends s to R, together with some authenticating data, on the basis of which R accepts or rejects s as having originated from D. More formally, the scheme should satisfy the following guarantees:

Correctness: If D, I, and R are honest, then R will accept s in ICReveal.

Non-Forgery: If D and R are honest, then R will reject any incorrect value $s^* \neq s$ passed to him in ICReveal, except with negligible probability.

Commitment: If I and R are honest, then at the end of ICValidate I knows a value s such that R will accept s in ICReveal, except with negligible probability.

Privacy: If D and I are honest, then prior to ICReveal, a cheating R has no information on s.

We call an IC scheme *linear* if it meets the following additional condition.

Linearity: If D, I, and R have invoked ICSetup and ICValidate with respect to several secrets $\{s_i\}$, then I may (without further interaction) invoke ICReveal to authentically disclose any (public) linear combination of the s_i without revealing each s_i individually.

Our weak secret sharing and verifiable secret sharing protocols make use of a linear IC subprotocol based on that of [CDD$^+$01], with some minor adjustments to increase broadcast efficiency. For completeness, the protocol and its proof of security appear in Appendix A.

Weak secret sharing. An *(n, t)-weak secret sharing scheme (WSS)* is a pair of protocols (WSS-Share, WSS-Rec) for a set of players $\mathcal{P} = \{P_1, \dots, P_n\}$, one of whom, the *dealer* D, holds input $s \in \mathbb{F}$. It must satisfy the following guarantees in the presence of an unbounded adversary corrupting up to t of the parties:

Weak Commitment: W.h.p., at the end of WSS-Share there exists a fixed value $s^* \in \mathbb{F} \cup \{\perp\}$, defined by the joint view of the honest parties, such that all honest parties will output the same value, either s^* or $\perp$, in WSS-Rec. If D is honest, then w.h.p. all honest parties will output $s^* = s$.

Privacy: If D is honest, then prior to WSS-Rec $\mathcal{A}$ gains no information on s (i.e., his view is statistically independent of s).

WSS is useful as an information-theoretic, distributed commitment for the dealer D. Thus, we may say that a dealer who completes WSS-Share has *committed* to his (effective) input s, and that upon completing WSS-Rec he *decommits* (if a proper value is reconstructed). We call a commitment to a value in $\mathbb{F}$ a *proper commitment* (regardless of whether it equals the dealer's actual input), and a commitment to $\perp$ an *improper* (or *garbage*) *commitment.* We will also need a slightly relaxed version of WSS called *WSS-without-agreement* (or *very weak secret sharing* [BPW91]), in which the Commitment property above is replaced by

WEAK COMMITMENT WITHOUT AGREEMENT: W.h.p., at the end of WSS-Share there exists a fixed value $s^* \in \mathbb{F} \cup \{\perp\}$, defined by the joint view of the honest parties, such that each honest party will output either s^* or $\perp$ in WSS-Rec (but some may output s^* and others $\perp$). If D is honest, then w.h.p. all honest parties will output $s^* = s$.

Furthermore, we will call a WSS(-without-agreement) *linear* if it satisfies the following in addition:

LINEARITY: If D has properly committed to several secrets $\{s^{(k)}\}$, then he may (without further interaction) invoke WSS-Rec to decommit to any (public) linear combination of the $s^{(k)}$. If some of the commitments are garbage, there still exists a fixed value $s^* \in \mathbb{F} \cup \{\perp\}$ which is reconstructed as the "linear combination" (w.h.p.).

We can slightly strengthen this requirement in the case of the sum of two values, to say that if one is properly committed and the other is garbage, their sum is garbage also (as opposed to any fixed value, which Linearity gives). We will use this property later on in the construction of VSS protocols.

PROPER + IMPROPER: If D has committed separately to $s \in \mathbb{F}$ and to $\perp$, then the reconstruction of the sum $s + \perp$ (or $\perp + s$) will yield $\perp$ (w.h.p.).

Our WSS(-without-agreement) protocol is presented in Section 3.1. It has a single sharing phase, which uses two broadcasts, and two different reconstruction phases: one which uses a single broadcast round and achieves ordinary WSS, and one which uses no broadcast but achieves only WSS-without-agreement.

Verifiable secret sharing. An (n, t)-*verifiable secret sharing scheme* [CGMA85] is a pair of protocols (VSS-Share, VSS-Rec) for a set of players $\mathcal{P} = \{P_1, \ldots, P_n\}$, one of whom, the *dealer* D, holds input $s \in \mathbb{F}$. In addition to the Privacy property above in the WSS case, VSS must satisfy the following, stronger guarantee in the presence of an unbounded adversary corrupting up to t of the parties:

COMMITMENT: W.h.p., at the end of VSS-Share there exists a fixed value $s^* \in \mathbb{F}$, defined by the joint view of the honest parties, such that all honest parties will output s^* in VSS-Rec. If D is honest, then $s^* = s$.

VSS strengthens WSS by guaranteeing that even when a cheating D does not cooperate in the Reconstruction phase, the honest players can still recover the value he committed to (which we now require to be a proper field element, not $\perp$). This makes possible a stronger variant of linearity, in which honest players can reconstruct linear combinations of secrets shared by *different dealers*. This strong linearity property is crucial for MPC applications of VSS.

We say that the parties *verifiably share* a secret s if each (honest) party maintains some state such that, when the honest parties invoke VSS-Rec on that joint state, they will reconstruct the value s (w.h.p.). Clearly, if a dealer D has just completed VSS-Share with effective input s, then the parties verifiably share s.

LINEARITY: If the parties verifiably share secrets $\{s^{(k)}\}$, then they also (without further interaction) verifiably share any (public) linear combination of the secrets.

We now turn to broadcast-type [LSP82] primitives over point-to-point channels (and slightly extended communication models; see below) which will become useful when further reducing the number of physical broadcasts (Section 3.2).

Gradecast. *Graded broadcast* (a.k.a. "gradecast") was introduced by Feldman and Micali [FM88]. It allows to broadcast a value among the set of recipients but with weaker consistency guarantees than in the case of standard broadcast, where all honest recipients are required to output the same value. In addition to the value v_i each recipient P_i also outputs a grade $g_i \in \{0, 1\}$.

Formally, a protocol achieves graded broadcast if it allows the dealer $D \in \mathcal{P}$ to distribute a value $v \in \mathcal{D}$ among parties $\mathcal{P}$ with every party P_i outputting a value $v_i \in \mathcal{D}$ with a grade $g_i \in \{0, 1\}$ such that:

VALIDITY: If the dealer D is correct, then every correct $P_i \in \mathcal{P}$ outputs $(v_i, g_i) = (v, 1)$.

GRADED CONSISTENCY: If a correct $P_i \in \mathcal{P}$ outputs (v_i, g_i) with $g_i = 1$, then every correct $P_j \in \mathcal{P}$ outputs (v_j, g_j) with $v_j = v_i$.

In [Fit03] gradecast is considered in different communication models. First, it is shown that gradecast is achievable from point-to-point channels if and only if $t < n/3$. Second, an extended communication model is considered where each player can broadcast to a (every) pair of other players. Such a primitive is called *2-cast*. A construction is then given which tolerates $t < n/2$ and achieves binary gradecast given 2-cast channels.

In this paper we will make use of a round-efficient gradecast protocol allowing *arbitrary* domains $\mathcal{D}$ based on that construction. Our construction works as follows: First, we construct a *weak broadcast* primitive (see next) given 2-cast; then, based on weak broadcast we build gradecast.

Weak broadcast. *Weak broadcast* (a.k.a. *Crusader agreement* [Dol82]) is another weak form of broadcast, where the recipients either decide on the value sent by the dealer or on a special symbol $\perp$ indicating that the dealer is malicious.

Formally, a protocol achieves weak broadcast if it allows the dealer D to distribute a value $v \in \mathcal{D}$ among parties $\mathcal{P}$ with every party P_i outputting a value $v_i \in \mathcal{D} \cup \{\perp\}$ such that:

VALIDITY: If the dealer D is correct, then every correct $P_i \in \mathcal{P}$ outputs $v_i = v$.

WEAK CONSISTENCY: If a correct $P_i \in \mathcal{P}$ outputs $v_i \neq \perp$, then every correct $P_j \in \mathcal{P}$ outputs $v_j \in \{v_i, \perp\}$.

In Appendix B we present modifications to the gradecast and weak broadcast protocols in [Fit03] to allow for arbitrary domains $\mathcal{D}$, instead of just the binary domain.

3 A Broadcast- and Round-Efficient VSS Protocol

In this section we present our new $(2, 0)$-broadcast, constant-round VSS protocol for $t < n/2$. Its overall round complexity is $(20, 1)$. This is the first linear VSS protocol enjoying such a small number of broadcast rounds without trusted setup, while running in an overall constant number of rounds. We first obtain a $(3, 0)$-broadcast, $(9, 1)$-round protocol which, at a high level, is inspired by the ($(7, 0)$-broadcast) protocol in [RB89]; we then apply a moderated-protocol transformation to shave off one additional broadcast round.

3.1 A $(3, 0)$-broadcast, constant-round VSS protocol

Our VSS protocol's sharing phase uses a WSS protocol, which we now describe.

Our WSS(-without-agreement) protocol uses two broadcasts in its sharing phase, and admits two different reconstruction phases: one which uses a single broadcast round and achieves ordinary WSS, and one which uses no broadcast but achieves only WSS-without-agreement. In turn, the protocol makes use of a linear IC subprotocol based on that in [CDD$^+$01] (Appendix A). The WSS protocol(s) is shown below. Since its sharing and reconstruction phases are invoked at different rounds of the VSS protocol's sharing phase, we specify them as separate protocols for convenience. The WSS protocol, with its two different reconstruction phases.

Protocol $\mathsf{WSS\text{-}Share}(\mathcal{P}, D, s)$

1. D chooses a random polynomial $f(x)$ of degree $\leq t$ such that $f(0) = s$, and sets $s_i := f(i)$; this will be P_i's share. For each pair $P_i, P_j \in \mathcal{P} - \{D\}$, run $\mathsf{ICSetup}(D, P_i, P_j, s_i)$.

2–5. **2 x BROADCAST in 4,5:** For each $P_i, P_j \in \mathcal{P} - \{D\}$, run $\mathsf{ICValidate}(D, P_i, P_j, s_i)$.

Protocol $\mathsf{WSS\text{-}Rec}(\mathcal{P}, D, s)$

1. For each pair $P_i, P_j \in \mathcal{P} - \{D\}$, run $\mathsf{ICReveal}(P_i, P_j, s_i)$.
2. **BROADCAST:** D broadcasts the polynomial $f(x)$ which he used to share the secret. P_i broadcasts the list of pieces $\{(j, s_j)\}$ which he accepted in $\mathsf{ICReveal}$ in the previous step.

Let HAPPY denote the set of players who accept at least $n-t$ pieces, and all of whose accepted pieces lie on the polynomial $f(x)$. If $|\mathsf{HAPPY}| \geq n-t$, all players take $s = f(0)$ to be the secret, otherwise $\perp$.

Protocol $\mathsf{WSS\text{-}Rec\text{-}NoBC}(\mathcal{P}, D, s)$

1. For each pair $P_i, P_j \in \mathcal{P} - \{D\}$, run $\mathsf{ICReveal}(P_i, P_j, s_i)$. If P_i accepts at least $n-t$ pieces, and all accepted pieces lie on a polynomial $f(x)$ of degree $\leq t$, then P_i takes $s = f(0)$ to be the secret, otherwise $\perp$.

Theorem 1. $\mathsf{WSS} = (\mathsf{WSS\text{-}Share}, \mathsf{WSS\text{-}Rec})$ *is a linear weak secret sharing scheme secure against a static, unbounded adversary corrupting* $t < n/2$ *players. Furthermore,* $\mathsf{WSS}^* = (\mathsf{WSS\text{-}Share}, \mathsf{WSS\text{-}Rec\text{-}NoBC})$ *is a linear WSS-without-agreement scheme, secure against a static, unbounded adversary who corrupts* $t < n/2$ *players.*

Proof. Commitment. First consider a cheating D. At the end of $\mathsf{WSS\text{-}Share}$, an honest P_i holds s_i which all honest parties will accept (due to the Commitment property of the IC protocol). Now these pieces s_i held by the honest parties define a polynomial $f^*(x)$; if $\deg f^*(x) > t$, then each honest party will accept pieces not lying on the dealer's broadcast polynomial $f(x)$. Therefore we will have $|\mathsf{HAPPY}| < n-t$, and $\perp$ will be reconstructed. Note that this situation is precisely a garbage commitment.
Otherwise $\deg f^*(x) \leq t$ (and the commitment is proper). If the dealer's broadcast polynomial $f(x) \neq f^*(x)$ then again each honest party will accept pieces not on $f(x)$, and so $\perp$ will be reconstructed. If $f^*(x) = f(x)$ then it may be the case that $\perp$ is reconstructed (depending on the values honest parties accept from dishonest parties), or that $s^* = f(0) = f^*(0)$ is reconstructed. Regardless, there is only one non-$\perp$ value which may be reconstructed, and it is fixed by the joint view of the honest parties at the end of $\mathsf{WSS\text{-}Share}$.
Now if D is honest, then by the IC Non-Forgery property no cheating party can fool an honest party into accepting a value other than s_i during $\mathsf{ICReveal}$ (except with negligible probability). It follows that each honest player will accept $\geq n-t$ pieces, and all their accepted values will lie on the dealer's polynomial $f(x)$. Thus $|\mathsf{HAPPY}| \geq n-t$ and the parties output $s = f(0)$.

Privacy. If D is honest, then by the IC Privacy property, the adversary has no information on any s_i value held by an honest player P_i prior to $\mathsf{ICReveal}$. Hence the adversary learns only the t points on the polynomial $f(x)$ corresponding to dishonest players' shares, and in particular has no information on $f(0) = s$ prior to $\mathsf{WSS\text{-}Rec}$.

Commitment Without Agreement. Define $f^*(x)$ as above, by the shares of the honest parties. As before if $\deg f^*(x) > t$, all honest parties will accept a set of pieces which do not lie on any degree t polynomial, and they will all output $\perp$.
If $\deg f^*(x) \leq t$, then honest party P_i will output $s^* = f^*(0)$ only if all the pieces he accepts from dishonest parties lie on $f^*(x)$; otherwise the set of pieces he accepts will lie on no polynomial of degree t, and he will output $\perp$.
For an honest D, the argument is the same as in the with-agreement case: Due to IC Non-Forgery, all honest parties will (w.h.p.) accept only values which lie on $f(x)$, and so all will output the correct value $s = f(0)$.

Linearity. Suppose D has properly committed to values $\{s^{(k)}\}$, using polynomials $f_k(x)$. Then for each value $s^{(k)}$, player P_i holds a share $s_i^{(k)}$. To decommit to a linear combination of the $s^{(k)}$, in $\mathsf{WSS\text{-}Rec}$ P_i reveals the linear combination of his $s_i^{(k)}$ during $\mathsf{ICReveal}$ (in place of "s_i"), and D broadcasts the linear combination of these polynomials (in place of "$f(x)$"). Then the properties of commitment and privacy remain in place, since taking a linear combination of polynomials of degree at most t results in a new polynomial of degree at most t.
If some of the commitments were garbage, this means exactly that some of the polynomials (defined by the shares of the honest players) were of degree $> t$. Nevertheless, taking a linear combination of these polynomials results in a single, fixed polynomial whose free term is the only possible non-$\perp$ value which honest parties will reconstruct (and then only if the new polynomial has degree $\leq t$).

Proper + Improper. A proper commitment is associated with a polynomial of degree $\leq t$, and an improper commitment with one of degree $> t$. Thus the sum of the two has degree $> t$, corresponds to another improper commitment, and will yield $\perp$ (w.h.p.). □

We are now ready to present $\mathsf{VSS}_{3\mathrm{bc}}$, our $(3, 0)$-broadcast, $(9, 1)$-round VSS protocol, which uses the WSS protocol above in its sharing phase. Regarding the presentation of our protocol(s), many VSS protocol descriptions rely on a bivariate-polynomial approach; others are univariate-based. We opt for the latter, since we feel that this protocol's structure lends itself best to a univariate polynomial description. At a high level, the protocol is inspired by that of Rabin and Ben-Or [RB89], and has a similar structure. First D distributes shares of a t-degree polynomial f where $s := f(0)$ and of additional random t-degree polynomials g_k. Each player P_i commits to all shares via WSS. Then the parties jointly carry out a cut-and-choose process in which the players are challenged to reconstruct either g_k or $f + g_k$ for each k, which must be degree t. Players who complain of incompatible shares, or fail to participate, have their shares broadcast (and hence fixed) by D.
As mentioned earlier, Rabin and Ben-Or's VSS requires 7 broadcast rounds in the share phase. One novelty which allows us to reduce broadcast round complexity to 3 is that we require the *dealer* as well as the players to commit via WSS to the shares he distributed, which constrains the misbehavior of a cheating dealer. After all commitments are in place, the players broadcast a round of cut-and-choose challenges in step 7. In step 8, parties respond to the challenges by using WSS-without-agreement to reconstruct the shares of the appropriate polynomials. In the final step 9, a broadcast is used to confirm the results of the WSS-without-agreement; at the same time D has a chance (and is obligated) to broadcast shares of players for whom he did not reconstruct the correct share in step 8.
An additional trick which saves us a broadcast round can be seen in step 6, which is inserted between the last two rounds of the WSS share phase. In this step, the parties perform a *pre-broadcast* by sending each other player their intended WSS final-round broadcast on point-to-point channels. In step 7, they officially complete WSS by echoing the pre-broadcast. This forces a cheating player to "semi-commit" in step 6 to one of at most $n-t$ possible final-round broadcasts for WSS, since a majority (including at least one honest player) must confirm his pre-broadcast. Luckily, semi-commitment restricts cheaters' options enough that players are able to broadcast the cut-and-choose challenges in the *same round*—step 7—rather than waiting for full commitment and then using another broadcast. (Note that in the case of a non-rushing adversary, step 6 is unnecessary.)

Protocol $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}(\mathcal{P}, D, s)$

1. D chooses a random polynomial $f(x)$ of degree $\leq t$ such that $f(0) = s$, and sets $s_i := f(i)$. Also for $1 \leq k \leq \kappa n$, D chooses random polynomials $g_k(x)$ of degree $\leq t$, and sets $t_{ki} := g_k(i)$. D sends $(s_i, \{t_{ki}\}_k)$ to P_i.

2–5. **BROADCAST:** P_i and D will now each act as WSS dealers to commit to P_i's share s_i. We reserve s_i to denote the value D commits to, and let s_i^* denote that which P_i commits to (these may be different if D and/or P_i is dishonest). D and P_i act as dealer in steps 1–4 of $\mathsf{WSS\text{-}Share}(D, s_i)$, $\mathsf{WSS\text{-}Share}(P_i, s_i^*)$, $\mathsf{WSS\text{-}Share}(D, t_{ki})$, and $\mathsf{WSS\text{-}Share}(P_i, t_{ki}^*)$ $(1 \leq k \leq \kappa n)$.

6. The parties have just completed WSS-Share step 4/ICValidate step 3. In the next step (corresponding to WSS-Share step 5/ICValidate step 4) the WSS/IC dealer will resolve conflicts. Instead of doing so immediately, let BC_i denote the broadcast which P_i would make. P_i first sends-to-all BC_i.
Also, if D conflicted with any P_i in the previous step (namely in ICValidate step 3) then in the following round D will broadcast *all* the values $(s_i, \{t_{ki}\}_k)$. For now, D sends-to-all these values, which we call *public pieces.*

7. **BROADCAST:** Now P_i broadcasts BC_i, which completes WSS-Share step 5/ICValidate step 4, and D broadcasts the values $(s_i, \{t_{ki}\}_k)$ which he sent-to-all in the previous step. Of course each player broadcasts his view of the previous step; if it is not the case that at least $t + 1$ players agree that P_i's broadcast this round matches what he told them in the previous round, then P_i is disqualified.

Additionally, each $P_i \neq D$ broadcasts a random challenge $C_i \in \{0,1\}^\kappa$ for D and for the other P_j's. The challenge indicates, for each index $k \in [\kappa n]$ assigned to P_i (κ such in total), whether:

(1) D and P_j should reveal $f(x) + g_k(x)$, in which case set $v_{kj} = s_j + t_{kj}$ and $v^*_{kj} = s^*_j + t^*_{kj}$; or

(2) D and P_j should reveal $g_k(x)$, in which case set $v_{kj} = t_{kj}$ and $v^*_{kj} = t^*_{kj}$.

8. $\forall k \in [\kappa n]$, $j \in [n]$, P_i participates in WSS-Rec-NoBC(D, v_{kj}) and WSS-Rec-NoBC(P_j, v^*_{kj}). P_i's outputs from these protocols are $v^{(i)}_{kj}$ and $v^{*(i)}_{kj}$, respectively.
9. **BROADCAST:** Each P_i broadcasts his view of the previous round—namely, the reconstructed shares $v^{(i)}_{kj}$ and $v^{*(i)}_{kj}$, for all k, j.
If a majority of players agrees on a non-$\perp$ reconstructed value for v_{kj} (resp. v^*_{kj}), then such value is the *broadcast (BC) consensus* for the given commitment, and the players who agree *participate in the consensus.* If no BC consensus exists, or if the player who shared the value does not participate, then the sharing player is disqualified. Consequently, if D is not so disqualified, then there exists a BC consensus (which D endorses) for all v_{kj}. Assuming this is the case, then D is nevertheless disqualified if for any k, the set of shares $\{v_{kj}\}_j$, together with appropriate public pieces, does not lie on a polynomial of degree $\leq t$.
In addition to broadcasting his view as described above, D also accuses player P_j, by publicly broadcasting the shares $(s_j, \{t_{kj}\}_k)$, if either of the following occurred:
(1) D output $\perp$ in any WSS-Rec-NoBC instance for which P_j was dealer; or
(2) D reconstructed an incorrect value for P_j's share of any challenge polynomial ($v^{*(D)}_{kj} \neq v_{kj}$).
If any such public pieces fail to lie on the appropriate degree-t polynomial, or if D neglects to accuse P_j when there exists a BC consensus that $v^*_{kj} \neq v_{kj}$, then D is disqualified.
Let HAPPY denote the set of **non-disqualified** players who were **not accused** by D. If $|\text{HAPPY}| < n - t$, then D is disqualified.

Protocol VSS-Rec$_{0\text{bc}}(\mathcal{P}, s)$

1. Each player $P_i \in$ HAPPY invokes WSS-Rec-NoBC(P_i, s_i).
Each player $P_i \in \mathcal{P}$ creates a list of shares consisting of those s_j which he accepts from any WSS-Rec-NoBC(P_j, s_j) (including his own), together with all public pieces s_j. He takes any $t+1$ shares from the list, interpolates a polynomial $f(x)$, and outputs $s := f(0)$ as the secret.

Theorem 2. *Protocol* VSS$_{3\text{bc}}$ = (VSS-Share$_{3\text{bc}}$, VSS-Rec$_{0\text{bc}}$) *is a* $(3,0)$*-broadcast,* $(9,1)$*-round, linear verifiable secret sharing scheme secure against an unbounded adversary who corrupts* $t < n/2$ *players.*

The number of broadcast rounds, as well as total number of rounds, is easily verified by inspection. In particular, broadcast is used in rounds 5, 7 and 9. We will specifically reference these rounds in the next section, where we only keep first and third broadcasts. The proof of Theorem 2 is broken up into three lemmas, as follows.

Lemma 3 (Privacy)**.** *If D is honest, then w.h.p. the adversary $\mathcal{A}$ gains no information on s prior to* VSS-Rec$_{0\text{bc}}$*.*

Proof. The secret-sharing properties of degree-t polynomials assure that the joint distribution of all shares handed by D to the corrupted parties in step 1, is uniformly random, in particular independent of s.
By the privacy property of protocol WSS employed in steps 2–7, the individual shares $(s_i, \{t_{ki}\}_k)$ of any honest party remain independent of the adversary's view. If in step 7 D broadcasts $(s_i, \{t_{ki}\}_k)$ for some P_i who conflicted with D in an instance of ICValidate, then that P_i must have been corrupt and hence $\mathcal{A}$ already knew these values (as well as the fact that D would broadcast them).
In step 7, $\mathcal{A}$ learns the honest parties' random challenges, which are independent of s and its shares, and thus yield no additional information.
The values reconstructed in step 8 are, for each challenge, either $f(x) + g_k(x)$ or $g_k(x)$. The $g_k(x)$'s themselves were chosen uniformly at random, and until step 8 $\mathcal{A}$ knew nothing about them except for

the shares held by corrupt parties, by WSS Privacy. Hence, conditioned on $\mathcal{A}$'s view up to that point, the revealed polynomial is uniformly random subject to consistency with the shares held by corrupted parties. Since D is honest he will answer all challenges correctly, and so $\mathcal{A}$ knows in advance that all honest parties will "accept" D's responses.
In step 9, $\mathcal{A}$ knows in advance what each honest party reconstructed from each WSS, so those broadcasts reveal nothing. Additionally, if D accuses P_j, then D output either $\perp$ or an incorrect value in some instance $\mathsf{WSS\text{-}Rec\text{-}NoBC}(P_j, *)$. This implies w.h.p. that P_j was dishonest (WSS Commitment Without Agreement), in which case $\mathcal{A}$ learns nothing when D broadcasts the shares $(s_j, \{t_{kj}\}_k)$.[2] □

Lemma 4 (Commitment)**.** *With high probability, at the end of* $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}$ *there exists a fixed* $s^* \in \mathbb{F}$ *such that all honest players output* s^* *during* $\mathsf{VSS\text{-}Rec}_{0\mathrm{bc}}$*. If* D *is honest, then* $s^* = s$*.*

The proof of this lemma is a bit laborious, and is presented in Appendix C.

Lemma 5 (Linearity)**.** *If the parties verifiably share secrets* $\{s^{(k)}\}$*, then they also (without further interaction) verifiably share any (public) linear combination of the* $s^{(k)}$*.*

Proof. Consider the situation when parties verifiably share a secret s according to the protocol, for a dealer D who was not disqualified. By Claim 13 of the Commitment proof, we know that w.h.p. D's WSS-commitment to s_i is proper for all i, and Claim 15 ensures that each happy player has properly WSS-committed to the same value. Since happy P_i have made proper WSS-commitments, the linearity of WSS-commitment implies that such P_i can reveal (and are committed to) any linear combination thereof.
Now consider secrets $s^{(k)}$ which are verifiably shared with shares $s_i^{(k)}$, interpolating polynomials $f_k(x)$ all of degree $\leq t$. (We ignore the "shares" of players who are disqualified in some execution of $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}$—such players must be corrupt and without loss of generality other players simply ignore their messages and shares during $\mathsf{VSS\text{-}Rec}_{0\mathrm{bc}}$.) Then any $t{+}1$ of the summed shares $\sum_k s_i^{(k)}$ interpolate the polynomial $f(x) = \sum_k f_k(x)$, which is of degree $\leq t$ with free term $\sum_k s^{(k)}$.
For any given non-disqualified P_i each share $s_i^{(k)}$ associated with that player is (w.h.p.) either (1) properly WSS-shared among all parties; or (2) publicly known. If *all* the $s_i^{(k)}$ for P_i are publicly known, then other players simply use the public sum $\sum_k s_i^{(k)}$ as P_i's share during $\mathsf{VSS\text{-}Rec}_{0\mathrm{bc}}$. Otherwise, by the linearity property of WSS P_i can reveal any sum of $s_i^{(k)}$'s. In particular, he can reveal exactly the sum of those $s_i^{(k)}$'s which are not already public, and this is what he does when revealing his "share" in $\mathsf{VSS\text{-}Rec}_{0\mathrm{bc}}$. This is of course the functional equivalent of revealing the sum of all the $s_i^{(k)}$'s since the other players need only add the public values to the reconstructed value to obtain the "true" share $\sum_k s_i^{(k)}$ of $\sum_k s^{(k)}$. (And revealing the sum of all shares reveals exactly the same information as revealing the sum of the non-public shares.) □

3.2 Further reducing the number of broadcast rounds

We now show how to modify protocol $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}$ so that only *two* rounds of broadcast suffice. This improvement is inspired by the transformation presented in Section 3.3 of [KK07].
We execute the transformation in three steps. First, we show how using one round of physical broadcast one can prepare a *setup* (details below) which allows to simulate sufficiently many gradecast channels later on. Second, we consider a *moderated* version of $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}$ where the dealer acts as the moderator. Third, we instruct the players to use one more round of physical broadcast in order to agree on whether the moderator behaved correctly or not. The overall construction results in a constant-round share protocol which uses physical broadcast in two rounds only—one for gradecast setup generation and one for agreeing whether the dealer's moderation was honest.
First, we describe two additional building blocks used in our transformation.

[2]As this discussion suggests, it may happen that D broadcasts an honest party's shares in step 9; this can only happen if $\mathcal{A}$ succeeds in an IC forgery attempt (hence with negligible probability). As a consequence, our protocol achieves statistical but not perfect privacy. On the other hand, privacy *is* perfect conditioned on the event that $\mathcal{A}$ is unsuccessful in all forgery attempts, as a failed forgery by itself reveals nothing about s.

Gradecast from setup. In [HR13], Hirt and Raykov recently showed how to prepare a *setup* allowing to simulate 2-cast channels (protocols Setup_3 and $\mathsf{Broadcast}_3$ in [HR13]). The setup protocol Setup_3 takes 3 rounds, where in the first two rounds point-to-point channels are used and in the third round a physical broadcast is used. The protocol $\mathsf{Broadcast}_3$ simulating 2-cast from the prepared setup uses point-to-point channels during 3 rounds.
Since gradecast is achievable from 2-cast channels in settings with $t < n/2$ (recall the description of gradecast in Section 2; see also Lemma 10), we can interpret the setup for 2-cast channels as a setup for gradecast channels. Let the protocol SetupGradecast be defined to generate such a setup, i.e., SetupGradecast runs sufficiently many instances of Setup_3 for each triple of players in parallel. The following lemma summarizes the security achieved by the pair of protocols (SetupGradecast, Gradecast).

Lemma 6. *Protocol* Gradecast *is a* 6*-round protocol achieving gradecast from a setup and point-to-point channels tolerating* $t < n/2$ *malicious parties. Moreover, the setup used by protocol* Gradecast *is prepared using the* 3*-round protocol* SetupGradecast*, where in the first two rounds point-to-point channels are used and in the third round physical broadcast is used.*

Moderated VSS. In [KK06], Katz and Koo proposed a new primitive called *moderated VSS* which allows to execute VSS under the supervision of a designated party called *the moderator.* If the moderator is honest, then the resulting protocol actually achieves the security properties of VSS; otherwise no security is guaranteed.
Formally, a two-phase protocol for parties $\mathcal{P}$, where there is a distinguished dealer $D \in \mathcal{P}$ who holds an initial input s and a moderator $P^{**} \in \mathcal{P}$ (who may possibly be the dealer), is a moderated VSS protocol tolerating t malicious parties if the following conditions hold for any adversary controlling at most t parties:

- Each honest party P_i outputs a bit f_i at the end of the sharing phase, and a value s_i at the end of the reconstruction phase.
- If the moderator is honest during the sharing phase, then each honest party P_i outputs $f_i = 1$ at the end of this phase.
- If there exists an honest party P_i who outputs $f_i = 1$ at the end of the sharing phase, then the protocol achieves VSS; specifically: (1) if the dealer is honest then all honest parties output s at the end of the reconstruction phase, and the joint view of all the malicious parties at the end of the sharing phase is independent of s, and (2) the joint view of the honest parties at the end of the sharing phase defines a value s' such that all honest parties output s' at the end of the reconstruction phase.

Theorem 7 ([KK06])**.** *Assume there exists a constant-round VSS protocol* Π*, using a broadcast channel in the sharing phase only, which tolerates* t *malicious parties. Then there exists a constant-round moderated VSS protocol* Π'*, using a gradecast channel and tolerating the same number of malicious parties.*

The compilation of Π into Π' is achieved by requiring the players to use a "moderated broadcast subroutine" to simulate broadcast. Each time players invoke the subroutine they update their flag f_i indicating whether the broadcast simulation has been successful. Players start executing Π' with f_i set to 1. The moderated subroutine Modercast for party P_i broadcasting a message m is defined as following.

Protocol Modercast$(\mathcal{P}, P^{**}, P_i, m)$

1. P_i gradecasts the message m.
2. The moderator P^{**} gradecasts the message he output in the previous step.
3. Let (m_j, g_j) and (m'_j, g'_j) be the outputs of party P_j in steps 1 and 2, respectively. Within the underlying execution of Π', party P_j will use m'_j as the message "broadcast" by P_i.
4. Furthermore, P_j sets $f_j := 0$ if (1) $g'_i \neq 1$, or (2) $m'_i \neq m_i$ and $g_i = 1$.[3]

We are now ready to show the enhanced VSS protocol.

[3]We note that in the description of the compilation from [KK06] gradecast with grades in $\{0, 1, 2\}$ is used. Here we

A $(2,0)$-broadcast, constant-round VSS protocol. In order to reduce the number of rounds where physical broadcast is used we apply the following transformation to the protocol $\mathsf{VSS\text{-}Share}_{3\text{bc}}$:

1. First, we generate gradecast setup using protocol $\mathsf{SetupGradecast}$.
2. We then run a *moderated* version of the protocol $\mathsf{VSS\text{-}Share}_{3\text{bc}}$, where the dealer acts as a moderator. The $\mathsf{Modercast}$ subroutine uses two sequential gradecast invocations that are simulated using the setup prepared by the protocol $\mathsf{Gradecast}$.
3. Finally, each player broadcasts (using physical broadcast) his flag f_i indicating whether he trusts the moderator (who is also the dealer). If the number of players broadcasting 1 is greater than $n/2$ then the sharing phase was successful, otherwise the dealer is disqualified.

The second and the third steps of the transformation have been already proposed by Katz and Koo in [KK07], while in the first step they make use of a pre-distributed PKI acting as a setup for gradecast. In our transformation, instead of assuming a PKI we generate a setup for gradecast using the protocol $\mathsf{SetupGradecast}$. We call the modified sharing phase $\mathsf{VSS\text{-}Share}_{2\text{bc}}$.
Furthermore, in $\mathsf{VSS\text{-}Share}_{2\text{bc}}$ we optimize the round complexity of the transformation by parallelizing the beginning and the end of $\mathsf{VSS\text{-}Share}_{3\text{bc}}$ with $\mathsf{SetupGradecast}$ and broadcasting the flags, respectively.

Protocol $\mathsf{VSS\text{-}Share}_{2\text{bc}}(\mathcal{P}, D, s)$

1-2. Players execute rounds 1 and 2 of the protocol $\mathsf{SetupGradecast}$ in parallel with rounds 1 and 2 of $\mathsf{VSS\text{-}Share}_{3\text{bc}}$.

3-5. **BROADCAST:** Players execute round 3 of the protocol $\mathsf{SetupGradecast}$ and rounds 3-5 of $\mathsf{VSS\text{-}Share}_{3\text{bc}}$. Each player broadcasts the concatenation of the values resulting from protocols $\mathsf{SetupGradecast}$ and $\mathsf{VSS\text{-}Share}_{3\text{bc}}$.

6. Players execute round 6 of the protocol $\mathsf{VSS\text{-}Share}_{3\text{bc}}$.

7-18. **MODERCAST:** Players execute round 7 of $\mathsf{VSS\text{-}Share}_{3\text{bc}}$ where the $\mathsf{Modercast}$ subroutine is used instead of broadcast. The subroutine invokes two gradecast channels sequentially. Each call to the gradecast channel is simulated using the protocol $\mathsf{Gradecast}$, which takes 6 rounds.

19. Players execute round 8 of the protocol $\mathsf{VSS\text{-}Share}_{3\text{bc}}$.

20. **BROADCAST:** Players execute round 9 of $\mathsf{VSS\text{-}Share}_{3\text{bc}}$. Each player additionally broadcasts flag f_i indicating whether $\mathsf{Modercast}$ was successful. If the number of $f_i = 1$ is greater than $n/2$, then the sharings generated by $\mathsf{VSS\text{-}Share}_{3\text{bc}}$ are accepted; otherwise, the dealer is disqualified.

Theorem 8. *Protocol* $\mathsf{VSS}_{2\text{bc}} = (\mathsf{VSS\text{-}Share}_{2\text{bc}}, \mathsf{VSS\text{-}Rec}_{0\text{bc}})$ *is a* $(2,0)$*-broadcast,* $(20,1)$*-round, linear verifiable secret sharing scheme secure against an unbounded adversary who corrupts* $t < n/2$ *players.*

Proof sketch. Due to Theorem 2 the protocol $\mathsf{VSS}_{3\text{bc}} = (\mathsf{VSS\text{-}Share}_{3\text{bc}}, \mathsf{VSS\text{-}Rec}_{0\text{bc}})$ is a linear verifiable secret sharing scheme secure against an unbounded adversary who corrupts $t < n/2$ players. Hence, due to Theorem 7 the protocol $\mathsf{VSS\text{-}Share}_{2\text{bc}}$ obtains a moderated VSS protocol when substituting broadcasts in $\mathsf{VSS\text{-}Share}_{3\text{bc}}$ with $\mathsf{Modercast}$. Finally, due to the definition of moderated VSS, if there exists at least one honest party with $f_i = 1$ then the moderated version of $\mathsf{VSS\text{-}Share}_{3\text{bc}}$ achieves VSS. Hence, since $t < n/2$, if more than $n/2$ parties broadcast $f_i = 1$ then $\mathsf{VSS\text{-}Share}_{2\text{bc}}$ achieves VSS; otherwise the dealer is corrupt and hence can be disqualified. □

4 Summary and Open Problems

In the $t < n/2$, unconditional security regime, just because protocols treat broadcast as a black-box should not, from a theoretical perspective or otherwise, entitle us to consider it a *free resource*, as there are compelling reasons to consider it more expensive than "mere" secure channels. In this paper we proposed a refinement of the round complexity of VSS, by adding a measure we term *broadcast*

use gradecast with grades in $\{0,1\}$ because during the compilation it is only required to distinguish the maximal grade from all other grades (so we put maximal grade to 1 instead of 2).

complexity, and seeked to minimize the number of rounds in which it is invoked as well, presenting a $(2, 0)$-broadcast, constant-round VSS protocol for $t < n/2$. This is the first linear VSS protocol enjoying such a small number of broadcast rounds without trusted setup, while running in an overall constant number of rounds.
One drawback of our resulting VSS protocol is that it is only proved secure for static adversaries, since our WSS protocol, based on [CDD$^+$01]'s, is not adaptively secure. It is possible that the VSS protocol is adaptively secure, even though the WSS protocol is not (per [Rab94, CDD$^+$01]). We leave this corroboration for future work. We also leave open the question of whether $(1, 0)$-broadcast, constant-round VSS protocols exist.

References

[BFO12] E. Ben-Sasson, S. Fehr, and R. Ostrovsky. Near-linear unconditionally-secure multiparty computation with a dishonest minority. In *CRYPTO*, pages 663–680, 2012.

[BGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. 20th Annual ACM Symposium of the Theory of Computation*, pages 1–10, May 1988.

[BH06] Z. Beerliová-Trubíniová and M. Hirt. Efficient multi-party computation with dispute control. In S. Halevi and T. Rabin, editors, *Theory of Cryptography*, volume 3876 of *Lecture Notes in Computer Science*, pages 305–328. Springer Berlin / Heidelberg, 2006.

[BPW91] Birgit Baum-Waidner, Birgit Pfitzmann, and Michael Waidner. Unconditional byzantine agreement with good majority. In *STACS*, pages 285–295, 1991.

[BTHR07] Zuzana Beerliová-Trubíniová, Martin Hirt, and Micha Riser. Efficient byzantine agreement with faulty minority. In Kaoru Kurosawa, editor, *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 393–409. Springer, 2007.

[CCD88] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings 20th Annual Symposium on Theory of Computing, STOC.* Association for Computing Machinery, May 1988.

[CDD$^+$01] R. Cramer, I. Damgård, S. Dziembowski, M. Hirt, and T. Rabin. Efficient multiparty computations secure against an adaptive adversary. In *Advances in Cryptology–EUROCRYPT'01*, Lecture Notes in Computer Science. Springer Verlag, May 2001.

[CGMA85] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *Proceedings of Twenty Sixth IEEE Symposium in Foundations of Computer Science*, pages 383–395, 1985.

[DDWY93] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly secure message transmission. *Journal of ACM*, 1(40):17–47, 1993.

[Dol82] Danny Dolev. The Byzantine generals strike again. *Journal of Algorithms*, (3):14–30, 1982.

[DPPU86] Cynthia Dwork, David Peleg, Nicholas Pippenger, and Eli Upfal. Fault tolerance in networks of bounded degree (preliminary version). In *STOC, Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing,* 28-30 *May* 1986*, Berkeley, California, USA*, pages 370–379, 1986.

[DS83] D. Dolev and H. Strong. Authenticated algorithms for Byzantine agreement. *SIAM Journal on Computing*, 12(4):656–666, 1983.

[FGG$^+$06] M. Fitzi, J. Garay, S. Gollakota, C. Pandu Rangan, and K. Srinathan. Round-optimal and efficient verifiable secret sharing. In *Proc. 3rd Theory of Cryptography Conference (TCC'06)*, Lecture Notes in Computer Science, pages 329–342. Springer Verlag, March 2006.

[FGH$^+$02] M. Fitzi, D. Gottesman, M. Hirt, T. Holenstein, and A. Smith. Detectable byzantine agreement secure against faulty majorities. In *Proceedings of the twenty-first annual symposium on Principles of distributed computing*, PODC '02, pages 118–126, New York, NY, USA, 2002. ACM.

[FGMR02] M. Fitzi, N. Gisin, U. Maurer, and O. von Rotz. Unconditional byzantine agreement and multi-party computation secure against dishonest minorities from scratch. In *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '02, pages 482–501, London, UK, 2002. Springer-Verlag.

[FH06] M. Fitzi and M. Hirt. Optimally efficient multi-valued Byzantine agreement. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, PODC '06, pages 163–168, New York, NY, USA, 2006. ACM.

[Fit03] Matthias Fitzi. *Generalized Communication and Security Models in Byzantine Agreement.* PhD thesis, ETH Zurich, March 2003. Reprint as vol. 4 of *ETH Series in Information Security and Cryptography*, ISBN 3-89649-853-3, Hartung-Gorre Verlag, Konstanz, 2003.

[FM88] Paul Feldman and Silvio Micali. Optimal algorithms for byzantine agreement. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 148–161, New York, NY, USA, 1988. ACM.

[GGO11] Juan A. Garay, Clint Givens, and Rafail Ostrovsky. Secure message transmission by public discussion: A brief survey. In *Coding and Cryptology — Third International Workshop (IWCC) IWCC*, volume 6639 of *Lecture Notes in Computer Science*, pages 126–141. Springer, 2011.

[GIKR02] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. On 2-round secure multiparty computation. In *Advances in Cryptology–CRYPTO'02*, Lecture Notes in Computer Science. Springer Verlag, August 2002.

[GL05] Shafi Goldwasser and Yehuda Lindell. Secure multi-party computation without agreement. *J. Cryptology*, 18(3):247–287, 2005.

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proc. 19th Annual ACM Symposium on Theory of Computation*, pages 218–229, May 1987.

[GO08] Juan A. Garay and Rafail Ostrovsky. Almost-everywhere secure computation. In *Advances in Cryptology–Eurocrypt'08*, Lecture Notes in Computer Science 4965, pages 307–323. Springer, April 2008.

[HR13] Martin Hirt and Pavel Raykov. On the complexity of broadcast setup. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 552–563. Springer, 2013.

[KK06] J. Katz and C.-Y. Koo. On expected constant-round protocols for Byzantine agreement. In *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 445–462. Springer, 2006.

[KK07] J. Katz and C.-Y. Koo. Round-efficient secure computation in point-to-point networks. In *Proceedings of the 26th annual international conference on Advances in Cryptology*, EUROCRYPT '07, pages 311–328, Berlin, Heidelberg, 2007. Springer-Verlag.

[KKK08] J. Katz, C.-Y. Koo, and R. Kumaresan. Improving the round complexity of VSS in point-to-point networks. In *ICALP '08: Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part II*, pages 499–510, Berlin, Heidelberg, 2008. Springer-Verlag.

[Koo07] C.-Y. Koo. *Studies on Fault-Tolerant Broadcast and Secure Computation.* PhD thesis, 2007.

[KPC10] R. Kumaresan, A. Patra, and C. Pandu Rangan. The round complexity of verifiable secret sharing: The statistical case. In *ASIACRYPT*, pages 431–447, 2010.

[LSP82] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, pages 382–401, July 1982.

[PCRR09] Arpita Patra, Ashish Choudhary, Tal Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 487–504. Springer, 2009.

[PW96] B. Pfitzmann and M. Waidner. Information-theoretic pseudosignatures and byzantine agreement for $t \geq n/3$. Technical Report RZ 2882 (#90830), IBM Research, 1996.

[Rab94] T. Rabin. Robust sharing of secrets when the dealer is honest or cheating. *J. ACM*, 41(6):1089–1109, 1994.

[RB89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proc. 21st ACM Symposium on the Theory of Computing*, pages 73–85, 1989.

[Upf92] Eli Upfal. Tolerating linear number of faults in networks of bounded degree. In *PODC*, pages 83–89, 1992.

A Information Checking Protocol

Here we give the information checking subprotocol used in our WSS and VSS constructions. It is based on that in [CDD+01] with some minor adjustments to increase broadcast efficiency. The three protocols ICSetup, ICValidate, and ICReveal are given below. We first need the following definition.

Definition A.1. Let $s, y, z, \alpha \in \mathbb{F}$. We say that the triple (s, y, z) is 1_α-*consistent* provided that the three points $(0, s)$, $(1, y)$, and (α, z) are colinear over $\mathbb{F}$.[4] One easily verifies that if (s, y, z) and (s', y', z') are 1_α-consistent, then linear combinations of these two vectors are as well.

Remark. Since we ultimately want to run many invocations of the IC protocol in parallel, some of the protocol steps allow events in parallel instances to affect the current instance. Such instructions are set off in square brackets (and can be ignored when considering the scheme as a stand-alone protocol).

Protocol $\mathsf{ICSetup}(D, I, R, s)$**:**

1. Dealer D chooses a random value $\alpha \in \mathbb{F} - \{0, 1\}$ and additional values $y, z \in \mathbb{F}$ such that (s, y, z) is 1_α-consistent. [D uses the same α for all parallel instances.] Also he chooses random values $s', y', z' \in \mathbb{F}$ such that (s', y', z') is 1_α-consistent. D sends (s, s', y, y') to the intermediary I, and (α, z, z') to recipient R.

Protocol $\mathsf{ICValidate}(D, I, R, s)$

1. I chooses a random value $d \in \mathbb{F}$ and sends it to D.
2. D sends the triple $(d, s' + ds, y' + dy)$ to R.
3. **BROADCAST:** Each player broadcasts the values he sent and received in the previous two rounds. I broadcasts his view of the triple $(d, s' + ds, y' + dy)$. Additionally, R checks that $(s' + ds, y' + dy, z' + dz)$ is 1_α-consistent; if not R broadcasts "reject values."

   Based on these broadcasts D may be *in conflict* with I and/or R:

   (1) D and I are in conflict if they disagree about the value of the triple $(d, s' + ds, y' + dy)$. [Or if they conflict in a parallel instance.]

   (2) D and R are in conflict if they disagree about what D sent in step 2, or if D is *not* in conflict with I, and R broadcast "reject values." [Or if they conflict in a parallel instance.]

   If no such conflicts arise, then all parties are satisfied and the phase ends here. Otherwise continue to step 4.
4. **BROADCAST:** If D, I are in conflict, then D broadcasts (s, y) and R adjusts z if necessary so that (s, y, z) is 1_α-consistent. This is done regardless whether D, R are in conflict or not, and the phase ends here.

   Otherwise, it must be that D, R are in conflict, but D, I are not. In this case D broadcasts (z, α) and I adjusts y if necessary so that (s, y, z) is 1_α-consistent.

Protocol $\mathsf{ICReveal}(I, R, s)$

1. I sends (s, y) to R, who accepts s if and only if (s, y, z) is 1_α-consistent.

Theorem 9. $\mathsf{IC} = (\mathsf{ICSetup}, \mathsf{ICValidate}, \mathsf{ICReveal})$ *is an IC scheme which remains secure when polynomially many instances of the* ICSetup*, and then* ICValidate*, phases are run in parallel. Additionally, it is linear with respect to such invocations.*

The proof is similar to that in [CDD+01] with some minor adjustments. We include it for completeness.

[4]That is, for some line $L(x) = bx + c$ over $\mathbb{F}$, we have $L(0) = s$, $L(1) = y$, and $L(\alpha) = z$.

Proof. CORRECTNESS. It is easy to see that if D, I, and R are all honest, then they will be satisfied in step 3 of $\mathsf{ICValidate}$, and R will accept s in $\mathsf{ICReveal}$.

NON-FORGERY. Since D and R are honest, they will never be in conflict. We claim that I gains no information on α during the $\mathsf{ICSetup}$ and $\mathsf{ICValidate}$ phases. The values he receives in $\mathsf{ICSetup}$ are independent of α. After choosing d in step 1 of $\mathsf{ICValidate}$, I knows exactly what D and R will broadcast in step 3 and so learns nothing. Moreover if D and I conflict, then D broadcasts the values s, y which I again already possessed.
Now consider the situation I is in upon invoking $\mathsf{ICReveal}$ for the first time with value s. He knows that if he sends the proper values (s, y), R will accept s, and if he alters just one of them, R will reject. Either way I learns nothing about α, which still appears uniform in $\mathbb{F} - \{0, 1\}$ to him. If he alters both values, to (s^*, y^*), R will accept only if (s^*, y^*, z) is 1_α-consistent. If that is the case then, since (s, y, z) is also 1_α-consistent, it follows that $(s - s^*, y - y^*, 0)$ is as well. From this fact I can deduce the value of α. In other words, in order to get R to accept a false value, I must guess the value of α, and if R does not accept, I learns only that this particular α was incorrect. Thus if I makes ℓ attempts, he has at best an $\ell/(|\mathbb{F}| - \ell - 2)$ chance of ever making R accept a false value, which is negligible for polynomial ℓ.

COMMITMENT. Here I and R are honest. If D conflicts with either one of them (or both), the property is trivial. Otherwise, D, I, and R all agree on the values $(d, s' + ds, y' + dy)$, and R accepted $(s' + ds, y' + dy, z' + dz)$ as 1_α-consistent. If for some $e \neq d$, $(s' + es, y' + ey, z' + ez)$ is also 1_α-consistent, then their difference $((d - e)s, (d - e)y, (d - e)z)$ is as well, and hence (s, y, z). Taking the contrapositive: if (s, y, z) distributed by D were *not* 1_α-consistent, there is at most one d which would have led R to accept the values. Since I chooses d randomly, an inconsistent (s, y, z) is detected (and a conflict occurs) except with negligible probability $1/|\mathbb{F}|$.

PRIVACY. Finally, assume D and I are honest. In the course of $\mathsf{ICSetup}$ and $\mathsf{ICValidate}$, a cheating R learns the values $\alpha, z, z', d, s' + ds, y' + dy$. However, he knows that the values $(s' + ds, y' + dy, z' + dz)$ are 1_α-consistent since D and I are honest. This implies that the value of $y' + dy$ can be computed based on $\alpha, z, z', d, s' + ds$, hence it can be removed from the view. This leaves only $\alpha, z, z', d, s' + ds$. Now fixing α, z, z' still leaves s' uniformly random (since y' may be any value), and since s' is used to blind ds, R learns no information on s.

LINEARITY. The protocol guarantees that R uses the same α value in all instances (for which the first two phases are parallel). The property follows immediately from the prior observation that linear combinations preserve 1_α-consistency. □

B Gradecast and Weak Broadcast for Arbirary Domains

We first give a straightforward modification of Protocol 5.1 in [Fit03] allowing for arbitrary domains $\mathcal{D}$ instead of the binary domain only:

Protocol $\mathsf{WeakBroadcast}(\mathcal{P}, D, v)$:

1. Dealer D 2-casts v to every pair of players in $\mathcal{P} \setminus \{D\}$.
2. $\forall P_i \in \mathcal{P} \setminus \{D\}$: If all values received in the previous step are the same (equal to some $u \in \mathcal{D}$) then output $v_i = u$, otherwise output $v_i = \bot$.
   Dealer D: Output v.

We now present a modification to Protocol 4.9 in [Fit03] which allows for arbitrary domains $\mathcal{D}$ instead of the binary domain only:

Protocol $\mathsf{Gradecast}(\mathcal{P}, D, v)$:

1. Dealer D: Weak broadcasts v. Denote output of each player P_i with w_i.
2. $\forall P_i \in \mathcal{P}$: Weak broadcast w_i. Denote output of each player P_j with w_{ij}.
3. $\forall P_i \in \mathcal{P}$: $\forall u \in \mathcal{D}$ let $T_i^u = \{P_j \in \mathcal{P} \mid w_{ji} = u\}$. Let v_i be u with maximal $|T_i^u|$ (break ties arbitrarily); if $|T_i^{v_i}| > n/2$ then $g_i = 1$, otherwise $g_i = 0$. Output (v_i, g_i).

Lemma 10. *The protocol* Gradecast *tolerates $t < n/2$ corruptions and achieves gradecast from 2-cast channels.*

Proof. It is easy to see that the protocol Gradecast satisfies Validity property of gradecast. In the following we prove that it achieves Graded Consistency as well. Consider any correct player P_i outputting (v_i, g_i) with $g_i = 1$. We have that $|T_i^{v_i}| > n/2$. Hence, there is at least one correct player P_j who weak-broadcasts v_i at Step 2. This implies that $w_j = v_i$ and any other correct player P_k has $w_k = v_i$ or $\perp$. Therefore, at Step 2 all correct players weak-broadcast v_i or $\perp$. Let H^{v_i} denote the set of correct players weak-broadcasting v_i and $H^{\perp}$ weak-broadcasting $\perp$. Then $M^{v_i} = T_i^{v_i} \setminus H^{v_i}$ is the set of corrupted parties who weak-broadcast v_i and $M^* = \mathcal{P} \setminus (H^{v_i} \cup H^{\perp} \cup M^{v_i})$ is the set of the remaining malicious players. We prove that $|M^*| < |H^{v_i}|$. Since $|T_i^{v_i}| > n/2$ we have that $|H^{v_i}| + |M^{v_i}| > n/2$. Since the majority of players are correct $|H^{v_i}| + |H^{\perp}| > n/2$. Combining two inequalities we have that $|H^{v_i}| + (|H^{v_i}| + |M^{v_i}| + |H^{\perp}|) > n$. Finally, we substitute the second term in the left part of the last inequality with $(n - |M^*|)$ and prove the claim.
In order to prove that Graded Consistency is satisfied we need to show that for any correct player P_j holds $|T_j^{v_i}| > |T_j^u|$ for any $u \neq v_i$. Since no player in $H^{v_i} \cup M^{v_i} \cup H^{\perp}$ weak-broadcasts u we have that $T_j^u \subseteq M^*$, hence $|T_j^u| \leq |M^*|$. Since all players in H^{v_i} are correct and weak-broadcast v_i we have that $|H^{v_i}| \leq |T_j^{v_i}|$. So, we have that $|T_j^u| \leq |M^*|$, $|H^{v_i}| \leq |T_j^{v_i}|$ and $|M^*| < |H^{v_i}|$ implies $|T_j^u| < |T_j^{v_i}|$. □

C Proofs Omitted from the Main Body

Here we give the complete proofs for statements which were omitted in the text. For convenience, we restate the results.

Lemma 4 (Commitment). *With high probability, at the end of* $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}$ *there exists a fixed $s^* \in \mathbb{F}$ such that all honest players output s^* during* $\mathsf{VSS\text{-}Rec}_{0\mathrm{bc}}$. *If D is honest, then $s^* = s$.*

Proof. Consider a cheating D. As in the protocol description, we let (s_i, t_{ki}) denote the values which D commits to in steps 2–7, and (s_i^*, t_{ki}^*) the values which P_i commits to. For dishonest players, these may of course be improper commitments. Nevertheless:

Claim 11. *If in step 9 there exists a BC consensus that P_i (or D) reconstructed $z \neq \perp$ in step 8 (not a sum of two shared values), then w.h.p. P_i (D) properly committed to z in steps 2–7.*

Proof. Since BC consensus requires a majority, at least one honest player must be part of the BC consensus. By the Commitment Without Agreement property of WSS, if any honest player reconstructs a non-$\perp$ value z, this value must have been properly committed to by the WSS dealer (w.h.p.). □

Claim 12. *If in step 9 there exists a BC consensus that P_i (or D) reconstructed a sum of two values, $z = z_1 + z_2 \neq \perp$ in step 8, then w.h.p. either (1) z_1, z_2 were each properly committed to in steps 2–7; or (2) z_1, z_2 were both* improperly *committed to in steps 2–7 via polynomials $h_1(x), h_2(x)$ of degree $> t$, where $h_1 + h_2$ is of degree $\leq t$. In particular, P_i (D) was committed at the end of step 7 to a fixed, unique value which honest players would reconstruct as the "sum" $z_1 + z_2$.*

Proof. Again at least one honest player must have joined the consensus. Then the claim follows directly from the Proper + Improper property of WSS. □

Recall that a player is disqualified if any value which he ostensibly committed to, lacks a BC consensus in step 9. In light of this, we will say that player P "decommitted" to value v, provided that a BC consensus agrees (in step 9) that P reconstructed v in step 8. In the case that there is no BC consensus (or the consensus is $\perp$), we say P "failed to decommit" and according to the protocol, P is disqualified.

Claim 13. (1) *If any of the shares s_i^* committed to by a party P_i is improper, i.e. $s_i^* = \perp$, then w.h.p. P_i will be disqualified.*
(2) *If any of the shares s_i committed to by D is improper, i.e. $s_i = \perp$, then w.h.p. D will be disqualified.*

Proof. (1) Technically, P_i is not committed to s_i^* until the end of step 7, since that is when the last step of the WSS protocol takes place. Nevertheless, at the end of step 6, a P_i who will not be disqualified is committed to a set of at most $n-t$ possible polynomials which must be his final commitment. Why? Because P_i is required in step 6 to send-to-all his upcoming broadcast (not including challenges), which fixes his commitment. According to the protocol, he will be disqualified in step 7 unless at least $t+1$ players agree that he has faithfully re-broadcast the message he sent them in step 6; therefore at least one honest party must have received in step 6 the broadcast which P_i makes in step 7.
Since honest parties also broadcast their *challenges* in step 7, a rushing adversary can make corrupt P_i's commitment depend on these challenges, from among the $n-t$ possibilities fixed in step 6. We must argue that this is not enough freedom for the adversary to dishonestly circumvent the challenges. Consider a *single* possible broadcast, of the potentially $n-t$ which P_i can make in step 7 without being immediately disqualified, and suppose this broadcast represents an improper commitment to $s_i^* = \perp$ (and various t_{ki}^*, each of which may or may not be improper). If some associated $t_{ki}^* \neq \perp$ is a proper commitment, then the Proper + Improper property of WSS implies that w.h.p. P_i will fail to decommit if he must reconstruct $s_i^* + t_{ki}^*$. On the other hand, if P_i made a garbage commitment $t_{ki}^* = \perp$, then he will fail to decommit if challenged to reconstruct t_{ki}^*. It follows that P_i can successfully decommit to *at most one of* $s_i^* + t_{ki}^*$ or t_{ki}^*.
But each honest challenge consists of κ independent requests to reveal either $s_i^* + t_{ki}^*$ or t_{ki}^* (for varying t_{ki}^*). Thus for a fixed honest challenge C_j, P_i will fail to correctly decommit with probability $\geq 1-2^{-\kappa}$. Thus the probability that C_j does *not* detect $s_i^* = \perp$ is $\leq 2^{-\kappa}$. Now since P_i can choose from $n-t$ possible broadcasts in step 7, the probability that there exists *at least one* improper commitment which C_j fails to detect, is $\leq (n-t)\cdot 2^{-\kappa} = \mathsf{negl}$. Therefore w.h.p if P_i makes *any* improper commitment (from among the up to $n-t$ potentially available to him in step 7), he will be disqualified w.h.p.
(2) The same argument works if we just replace P_i with D, s_i^* with s_i, and t_{ki}^* with t_{ki}. □

Claim 14. *If any of the shares t_{ki} committed to by D is improper, i.e. $t_{ki} = \perp$, then w.h.p. D will be disqualified.*

Proof. By the preceding claim, a non-disqualified D must have properly committed to all s_i (w.h.p.). Condition on an execution where this holds, and suppose D made a garbage commitment to $t_{ki} = \perp$. If D must reconstruct $v_{ki} = s_i + t_{ki}$ in step 8, then by the Proper + Improper property of WSS, he will w.h.p. fail to decommit, and be disqualified. Alternatively, if D must reconstruct $v_{ki} = t_{ki}$ in step 8, he will fail w.h.p. since t_{ki} is itself improperly committed, and be disqualified. He's required to do one of these by the relevant challenge broadcast in step 7. □

Claim 15. *Assume D is not disqualified. If in steps 2–7, P_i committed to $s_i^* \neq s_i$ (i.e., a share different from the one committed to by D), then w.h.p. either P_i will be disqualified, or P_i will be accused by D in step 9.*

Proof. Assume P_i is not disqualified—then there exists a BC consensus in step 9 for the value v_{ki}^*. Since D is not disqualified, there exists also a BC consensus for v_{ki} (which D participates in). Each consensus is (w.h.p.) correct in the sense that P_i and D were committed already in steps 2–7 to reconstruct these values (Claims 11 and 12).
If D does not participate in the BC consensus for v_{ki}^*, then he must accuse P_i. Then P_i will be unhappy, and we are done. Otherwise D does participate in the BC consensus for v_{ki}^*. Then since $s_i^* \neq s_i$, we have that for all k corresponding to some fixed honest challenge, either

$$s_i + t_{ki} \neq s_i^* + t_{ki}^* \qquad \text{or} \qquad t_{ki} \neq t_{ki}^*. \tag{1}$$

Now as in the previous claims, if D and/or P_i are dishonest, they are only partially committed to their shares at the time they see the honest challenges: At the end of step 6, there exist $\leq n-t$ possible broadcasts which each one can make in step 7, finally committing them to a fixed set $(s_i, \{t_{ki}\}_k)$ or $(s_i^*, \{t_{ki}^*\}_k)$. Nevertheless, for any fixed pair of possible broadcasts D and P_i could make in step 7 (and not be disqualified), the probability that a given honest challenge will force D and P_i to reconstruct some unequal values $v_{ki} \neq v_{ki}^*$ is $\geq 1-2^{-\kappa}$. There are (at most) $(n-t)^2$ possible pairs of step 7

broadcasts, hence with probability $\geq 1-(n-t)^2 2^{-\kappa} = 1 - \mathsf{negl}$, D and P_i will be committed to reconstruct unequal values regardless of which broadcasts they choose in step 7.
Since BC consensus exists for each of these unequal values (otherwise one or both of D, P_i are disqualified), D will be disqualified unless he accuses P_i in step 9. □

In light of Claims 13 and 14, a cheating D who is not disqualified must have (w.h.p.) committed to genuine field elements $(s_i, \{t_{ki}\}_k)$ for all k, i. Given that this is the case, and with slight abuse of notation, let $f(x)$ denote the polynomial interpolating all s_i, and for each k let $g_k(x)$ be the one interpolating all t_{ki}.

Claim 16. *If* $\deg f(x) > t$*, then w.h.p.* D *is disqualified.*

Proof. Suppose that $\deg f(x) > t$. For each fixed k, we have a guarantee that either

$$\deg(f(x) + g_k(x)) > t \qquad \text{or} \qquad \deg g_k(x) > t. \tag{2}$$

D is partially committed to the set $(s_i, \{t_{ki}\}_k)$ at the end of step 6, in that there are at most $n-t$ possible broadcasts he can successfully make in step 7 to conclude the WSS. In step 8, based on the step 7 challenges, he must reconstruct either the values $v_{ki} = s_i + t_{ki}$, i.e. the polynomial $f(x) + g_k(x)$, or the values $v_{ki} = t_{ki}$, i.e. the polynomial $g_k(x)$.
Again, for any *single, fixed* broadcast D could make, and given honest challenge, the probability that he can successfully decommit in step 8 (leading to a BC consensus in step 9 involving only polynomials of degree $\leq t$) is negligible; hence even allowing him to choose from among the $n-t$, his success probability remains negligible, and w.h.p. he will be disqualified. □

Claim 17. *No honest player* P_i *is disqualified. (Hence at the end of* $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}$*, each honest player is either happy, or has been accused by* D *and his shares made public.)*

Proof. Honest P_i will not be disqualified for misbehaving during any WSS subprotocol. In step 6, P_i will faithfully send his pre-broadcast to all other honest players, hence at least $t+1$ players will confirm it in step 7, and he will not be disqualified there. In step 8, the properties of $\mathsf{WSS\text{-}Rec\text{-}NoBC}$ ensure that for an honest dealer all honest players will correctly reconstruct v^*_{ki}, thus these values will have BC consensus in step 9 (in which P_i participates), and he will not be disqualified there.
As for honest D, the same holds. Additionally, since he will have shared s_i and $\{t_{ki}\}_k$ using polynomials of correct degree, and announce correct public pieces in step 7, then all values v_{ki} which he reconstructs in $\mathsf{WSS\text{-}Rec\text{-}NoBC}$ in step 8 will lie on polynomials of appropriate degree. □

Recall that in $\mathsf{VSS\text{-}Rec}_{0\mathrm{bc}}$, each happy party whose share s_i is not yet public is supposed to invoke $\mathsf{WSS\text{-}Rec\text{-}NoBC}(P_i, s_i)$ to reveal it. By Claim 15, we see that all parties in HAPPY can reveal only the share $s^*_i = s_i$ (or possibly no share at all, if dishonest). In short, all public pieces and all committed values of happy parties are equal to s_i and thus lie on $f(x)$, which is of degree $\leq t$ (Claim 16). Since only such shares are used during $\mathsf{VSS\text{-}Rec}_{0\mathrm{bc}}$ to construct $f(x)$—and since at least all $\geq t+1$ shares associated with honest parties will be recovered by every honest party—each honest player will reconstruct $f(x)$ and output $s = f(0)$, a value which is fixed by the joint view of the honest parties at the end of $\mathsf{VSS\text{-}Share}_{3\mathrm{bc}}$.
It is easy to see that if in fact D is honest, then the polynomial $f(x)$ will satisfy $f(0) = s$, and all honest parties will reconstruct the correct value.
Finally, as shown above, if D is dishonest and not disqualified, then w.h.p. the secret s^* which he commits to is $f(0)$, where f interpolates all values s_i which D committed to via WSS. In turn, the values s_i are computable from D's view of the shares D distributed in the WSS sharing phase, which are themselves computable from D's messages in the underlying IC protocols. Hence the criterion for input independence holds as well. □