# Computational Verifiable Secret Sharing Revisited*

Michael Backes[1,2], Aniket Kate[1], and Arpita Patra[3]**

[1] Max Planck Institute for Software Systems (MPI-SWS), Germany
backes@mpi-sws.org, aniket@mpi-sws.org
[2] Saarland University, Germany
[3] Aarhus University, Denmark
arpita@cs.au.dk

**Abstract.** Verifiable secret sharing (VSS) is an important primitive in distributed cryptography that allows a dealer to share a secret among $n$ parties in the presence of an adversary controlling at most $t$ of them. In the *computational* setting, the feasibility of VSS schemes based on commitments was established over two decades ago. Interestingly, all known computational VSS schemes rely on the homomorphic nature of these commitments or achieve weaker guarantees. As homomorphism is not inherent to commitments or to the computational setting in general, a closer look at its utility to VSS is called for. In this work, we demonstrate that homomorphism of commitments is not a necessity for computational VSS in the synchronous or in the asynchronous communication model. We present new VSS schemes based only on the definitional properties of commitments that are almost as good as the existing VSS schemes based on homomorphic commitments. Importantly, they have significantly lower communication complexities than their (statistical or perfect) unconditional counterparts.

Further, in the synchronous communication model, we observe that a crucial interactive complexity measure of *round complexity* has never been formally studied for computational VSS. Interestingly, for the optimal resiliency conditions, the least possible round complexity in the known computational VSS schemes is identical to that in the (statistical or perfect) unconditional setting: three rounds. Considering the strength of the computational setting, this equivalence is certainly surprising. In this work, we show that three rounds are actually not mandatory for computational VSS. We present the first two-round VSS scheme for $n \geq 2t + 1$ and lower-bound the result tightly by proving the impossibility of one-round computational VSS for $t \geq 2$ or $n \leq 3t$. We also include a new two-round VSS scheme using homomorphic commitments that has the same communication complexity as the well-known three-round Feldman and Pedersen VSS schemes.

**Keywords:** Verifiable Secret Sharing, Round Complexity, Commitments, Homomorphism

* An extended version of this paper is available [1].

** Supported by Center for Research in the Foundations of Electronic Markets (CFEM), Denmark and Center for the Theory of Interactive Computation (CTIC).

## 1 Introduction

The notion of secret sharing was introduced independently by Shamir [30] and Blakley [2] in 1979. Since then, it has remained an important topic in cryptographic research. For integers $n$ and $t$ such that $n > t \geq 0$, an $(n, t)$-*secret sharing* scheme is a method used by a *dealer* $D$ to share a secret $s$ among a set of $n$ parties (the *sharing* phase) in such a way that in the *reconstruction* phase any subset of $t + 1$ or more honest parties can compute the secret $s$, but subsets of size $t$ or fewer cannot. Since in some secret sharing applications the dealer may benefit from behaving maliciously, parties also require a mechanism to confirm the correctness of the dealt values. To meet this requirement, Chor et al. [6] introduced the concept of *verifiable secret sharing* (VSS).

VSS has remained an important area of cryptographic research for the last two decades [3, 9–11, 13, 20, 21, 23, 26, 27]. In the literature, VSS schemes are categorized based on the adversarial computational power: computational VSS schemes and unconditional VSS schemes. In the former, the adversary is computationally bounded by a security parameter, while in the latter the adversary may possess unbounded computational power. Naturally, the computational VSS schemes are significantly more practical and efficient in terms of message and communication complexities as compared to the unconditional schemes. Thus,

the majority of the recent research has been focussed on devising practical constructions for unconditional VSS. In this work, we revisit the concept of computational VSS [3,9,13,26] to settle the round complexity of computational VSS based on minimal cryptographic assumptions (which is cryptographic commitment in our case) and to investigate the role of homomorphism of commitment schemes in the context of VSS.

***Motivation and Contributions.*** The major savings in the computational VSS schemes come from the use of cryptographic commitments. Interestingly, we find that all computational VSS schemes in the literature except [13, App. A] (which satisfies weaker conditions; see related work) require these commitments to be homomorphic. However, homomorphism is not inherent to cryptographic commitments; it is an additional property provided by discrete logarithm (DLog), Pedersen [27] and few other commitment schemes. As we elaborate later in the paper, commitments can be designed from general primitives such as one-way functions or collision-free hash functions; but, homomorphism may not be guaranteed in these constructions. Furthermore, relying on as little assumptions as possible without much loss in efficiency is always a general goal in cryptography. Therefore, computational VSS schemes based only on the definitional properties of commitments can be interesting to study.

In this paper, we show that homomorphism is not a necessity for VSS in both synchronous (known and bounded message delays) and asynchronous (unbounded message delays) communication model. While our VSS schemes (in both network settings) based on any commitment scheme are almost as good as the existing computational VSS protocols using homomorphic commitment

schemes in terms of communication, they are considerably better than the unconditional VSS schemes.

In the synchronous communication model with a broadcast channel, Gennaro et al. [11] initiated the study of round complexity (number of rounds required to complete an execution) and proved a lower bound of three rounds during the sharing phase and one round during the reconstruction phase for unconditional VSS. The work was extended in [10,20] with tight polynomial time constructions, and in [21, 23] by improving the bounds in a statistical scenario where the VSS properties are held *statistically* and can be violated with a negligible probability.

The round complexity of *computational* VSS has never being formally analyzed in the synchronous VSS literature. We observe that the round complexity of all known practical computational VSS protocols [9, 27] for the optimal resilience of $n \geq 2t + 1$ is the same as that of unconditional VSS schemes: three rounds in the sharing phase.[4] This similarity is surprising considering the usage of commitments in computational VSS. We analyze the round complexity of computational VSS with homomorphic and non-homomorphic commitments.

1. We show the impossibility of 1-round computational VSS protocol in the standard communication model under consideration; specifically, we prove that a computational VSS scheme with one round in the sharing phase is impossible for $t \geq 2$ or $n \leq 3t$. However, we find that there exists a special 1-round VSS construction for $t = 1$ and $n \geq 4$, when the dealer is one of the participants; we include the construction in the full version of the paper [1].
2. We then tighten our lower-bound result by providing a 2-round computational VSS scheme for $n \geq 2t + 1$ using any commitment scheme. Existing VSS schemes [9, 13, 27] based on homomorphic commitments require three rounds for $n \geq 2t+1$. Comparing with unconditional VSS schemes, we notice that the message (the number of messages transferred) and communication (the number of bits transferred) complexities of our scheme are at least a linear factor less. Also, our scheme is better in terms of round complexity or resilience bound as compared to all known unconditional VSS schemes.
   We then provide a VSS scheme for $n \geq 2t + 1$ using homomorphic commitments that has the same message and communication complexities but requires one less round of communication as compared to [9, 13, 27].

***Organization.*** In the rest of this section, we review the related work. In Section 2, we describe our adversary model, and definitions of VSS and commitments. We present all our results for the synchronous model in Section 3 and those for the asynchronous model in Section 4. In Section 5, we discuss a few interesting open problems. Some of our proofs are shifted to the full version [1].

***Related Work.*** For our work in the synchronous setting, we closely follow the network and adversary model of the best known VSS schemes: Feldman

[4] Note that it is possible to reduce a round in sharing in [9, 27] but that asks for a sub-optimal resilience of $n \geq 3t + 1$. Further, with a much stronger assumption of non-interactive zero-knowledge (NIZK), it is possible to reduce the number of sharing rounds to one for $n \geq 2t + 1$ in the public key infrastructure [15].

VSS [9] and Pedersen VSS [27]. These schemes are called *non-interactive* as they require unidirectional private links from the dealer to the parties; non-dealer parties speak only via the broadcast channel. Our protocol assumes nearly the same network model; however, in addition, we also allow parties to send messages to the dealer over the private channels. In practice, it is reasonable to assume that private links are bidirectional. Note that we do not need any private communication links between non-dealer parties.

It is also important to compare our results with unconditional VSS as we work towards reducing the cryptographic assumptions required for computational VSS. In unconditional or information theoretic settings, there are two different possibilities for the VSS properties; they can be held *perfectly* (i.e., error-free) or *statistically* with negligible error probability. Perfect VSS is possible if and only if $n \geq 3t+1$ [8], while statistical VSS is possible for $n \geq 2t+1$ [28], assuming a broadcast channel. Gennaro et al. [11] initiated the study of the round complexity of unconditional VSS, which was extended by Fitzi et al. [10] and Katz et al. [20]. They concentrate on unconditional VSS with perfect security and show that three rounds in the sharing phase are necessary and sufficient for $n \geq 3t+1$. In the statistical scenario, Patra et al. [23] show that $n \geq 3t+1$ is necessary and sufficient for 2-round statistical VSS. Recently, Kumaresan et al. [21] extended the result to prove that 3 rounds are enough for designing statistical VSS with $n \geq 2t + 1$.

The round complexity is never studied formally for computational VSS. In the standard model that we follow, the best known computational VSS protocols [9, 13, 27] require two rounds; however, they work only for a suboptimal resilience of $n \geq 3t + 1$. Although these schemes can also be adopted for $n \geq 2t + 1$, they then ask for *three* rounds. In addition, the only known VSS scheme among these that does not mandate homomorphic commitments, [13, App. A], does not satisfies the generally required stronger commitment property described in Section 2.2. In this paper, we improve all the above results by showing that two rounds are necessary and sufficient for (stronger) VSS with $n \geq 2t + 1$ using (homomorphic or non-homomorphic) cryptographic commitments. Note that it is also possible to achieve 1-round VSS in the presence of a public-key infrastructure (PKI) employing NIZK proofs [15]. However, NIZK proofs requires a common reference string or a random oracle. Furthermore, the scheme of [15] can only achieve computational secrecy, whereas our schemes can obtain unconditional or computational secrecy as required.

For our work in the asynchronous setting, we follow the standard model of Cachin et al. [3]. In the asynchronous setting, Cachin et al. [3], Zhou et al. [31], and more recently Schultz et al. [29] suggested computational VSS schemes. Of these, protocol by Cachin et al. is the most practical computational VSS protocol with $O(n^2)$ message complexity. However, all of these schemes rely on homomorphism of the commitment scheme. We avoid the use of homomorphism, while maintaining the message complexity of the VSS protocol by Cachin et al. [3]. Note that our protocol is significantly efficient in all aspects as compared to unconditional VSS schemes [4, 5, 24, 25] in the asynchronous setting.

## 2 Preliminaries

We work in the computational security setting, where $\kappa$ denotes the security parameter of the system, in bits. We assume that the dealer's secret $s$ lies over a finite field $\mathbb{F}_p$, where $p$ is an $\kappa$ bits long prime. Our polynomials for secret sharing belong to $\mathbb{F}_p[x]$ or $\mathbb{F}_p[x, y]$, and the indices for the parities are chosen from $\mathbb{Z}_p$. Without loss of generality, we assume these indices to be $\{1, \ldots, n\}$. A function $\epsilon(\cdot) : \mathbb{N} \to \mathbb{R}^+$ is called *negligible* if for all $c > 0$ there exists a $\kappa_0$ such that $\epsilon(\kappa) < 1/\kappa^c$ for all $\kappa > \kappa_0$. In the paper, $\epsilon(\cdot)$ denotes a negligible function.

### 2.1 Adversary Model

We consider a network of $n$ parties $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$, where a distinguished party $D \in \mathcal{P}$ works as a dealer. Our adversary $\mathcal{A}$ is *$t$-bounded* and it can compromise and coordinate actions of up to $t$ out of $n$ parties. We also assume that the adversary is *adaptive*; it may corrupt any party at any instance during a protocol execution as long as the number of corruptions is bounded by $t$.

We work in the synchronous as well as the asynchronous settings , and postpone the discussions on communication setting to the respective sections (synchronous model in Section 3 and asynchronous model in Section 4).

### 2.2 VSS and Variants

We now present a definition of VSS [11]. A VSS protocol among $n$ parties $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ with a distinguished party $D \in \mathcal{P}$ consists of two phases: a *sharing* phase and a *reconstruction* phase.

**Sharing.** Initially, $D$ holds an input $s$, referred to as the secret, and each party $P_i$ may hold an independent random input $r_i$. At the end of the sharing phase, each honest party $P_i$ holds a view $v_i$ that may be required to reconstruct the dealer's secret later.

**Reconstruction.** In this phase, each party $P_i$ publishes its entire view $v_i$ from the sharing phase, and a reconstruction function $\texttt{Rec}(v_1; \ldots; v_n)$ is applied and is taken as the protocol's output.

We call an $n$-party VSS protocol, with $t$-bounded adversary $\mathcal{A}$, an $(n, t)$-VSS protocol if it satisfies the following conditions:

**Secrecy.** If $D$ is honest then the adversary's view during the sharing phase reveals no information about $s$. More formally, the adversary's view is *identically distributed* for all different values of $s$.

**Correctness.** If $D$ is honest then the honest parties output the secret $s$ at the end of the reconstruction phase.

**Commitment.** If $D$ is dishonest, then at the end of the sharing phase there exists a value $s^* \in \mathbb{F}_p \cup \{\perp\}$, such that at the end of the reconstruction phase all honest parties output $s^*$.

The sharing phase as well as the reconstruction phase may consist of several communication rounds. A VSS protocol is considered *efficient* if the total computation and communication performed by all the honest parties is polynomial in $n$ and the security parameter $\kappa$. The optimal resiliency bound for VSS is $n \geq 2t + 1$ (in the presence of a broadcast channel) in the synchronous setting and $n \geq 3t + 1$ in the asynchronous setting.

***Variants of VSS.*** A few variants of VSS have been introduced as required in secret sharing applications. We briefly describe those below.

1. In our VSS definition, we assume that secrecy is unconditional, while correctness and commitment are computational. We can have a variation where secrecy is computational, and correctness and commitment are unconditional in nature. This is easily possible as secrecy and correctness of a VSS scheme are derived respectively from the hiding and binding of the commitment scheme under use. Our lower bound results hold for this variation as well. However, for computationally secure VSS, we can prove security only against a *static* adversary that chooses $t$ parties before a protocol execution starts.
2. In our VSS, the reconstruction may end with $\perp$. By fixing a default value in $\mathbb{F}_p$ (say 0) that will be output instead of $\perp$, it is possible to say that $s^* \in \mathbb{F}_p$. However, as suggested in [11, Sec. 2.1], there is even a stronger VSS definition possible. The stronger definition has exactly the same secrecy and correctness properties, but has a stronger commitment property:
   **Strong Commitment.** Even if $D$ is dishonest, at the end of the sharing phase, each party locally outputs a share of a secret $s^*$ chosen only from $\mathbb{F}_p$ such that shares from any $t+1$ honest parties are consistent with $s^*$.
   For Shamir's secret sharing, this property means that at the end of the sharing phase, there exists a $t$-degree polynomial $f(x)$ such that a share $s_i$ held by every honest party $P_i$ is equal to $f(i)$. While our asynchronous protocol in Section 4.2 satisfies the basic VSS definition, our 2-round protocols in sections 3.2 and 3.4 satisfy the stronger definition. In the full version [1], we present an asynchronous protocol satisfying the stronger definition.
3. Another stronger variant of VSS considers dealer $D$ to be an external party (i.e., $D \notin \mathcal{P}$) and allows the $t$-bounded adversary to corrupt the dealer and up to $t$ additional parties in $\mathcal{P}$.
   Our lower bound results and all of our protocols except our one-round VSS protocol [1] work for this variant as well. We show that 1-round VSS with an external dealer is impossible even when $t = 1$ irrespective of the value of $n$ and the number of rounds in the reconstruction phase.

We work on VSS as a standalone primitive in this paper. The required VSS properties, specially the commitment property, may change in some VSS application. We consider that to be an interesting future work and discuss in Section 5.

### 2.3 Commitment Schemes

A cryptographic commitment scheme is a two-phase cryptographic protocol between a *committer* and a *verifier*.

**Commit Phase.** Given a message $m$, a committer runs $[\mathcal{C}, (m, d)] = \mathsf{Commit}(m)$ and publishes $\mathcal{C}$ as a *commitment* that binds her to message $m$ (*binding*) without revealing it (*hiding*). The function *may* output an opening value $d$.

**Open Phase.** The committer opens commitment $\mathcal{C}$ by revealing $(m, d)$ to a verifier. The verifier can then check if the message is consistent with the commitment (i.e., $m \stackrel{?}{=} \mathsf{Open}(C, m, d)$).

We note that the commitment schemes also require a setup that generally involves choosing the cryptographic parameters. This can easily be included in the VSS setup and thus we do not consider it in detail.

A commitment scheme cannot be unconditional (perfect or statistical) binding and hiding at the same time. As a result, commitments come in two flavors: perfect (or statistical) binding but computational hiding commitments, and perfect (or statistical) hiding but computational binding commitments. There are many applications of commitments where they may never be opened or opened only after a while. In such scenarios, commitments of the second type are generally considered advantageous over the first type, since the committed values are hidden in information theoretic sense in the second type.

Perfect hiding but computational binding (under the DLog assumption) Pedersen commitment scheme [27] is the most commonly used commitment scheme in computational VSS. It has an interesting additive homomorphic property that a product of two commitments $\mathcal{C}_1$ and $\mathcal{C}_2$ (associated respectively with messages $m_1$ and $m_2$) commits to an addition of the committed messages $(m_1 + m_2)$. However, with its reliance on the DLog assumption, this commitment scheme will not be suitable once quantum computers arrive.

On the other hand, commitments of both types can be achieved from any one-way function (see [16] and references within). In this paper, we concentrate on the commitments of the second type, whose efficient constructions are possible from any claw-free permutation [14], any one-way permutation [22] or any collision-free hash function [17]. Along with being non-homomorphic, some of these commitment constructions are also interactive in the nature. We restrict ourselves to the non-interactive commitment constructions (e.g., [14] and [17]) as the interactive commitment constructions may increase the rounds complexity of our VSS schemes.

## 3 VSS in the Synchronous Network Model

Before presenting our results in the synchronous setting, we describe our synchronous communication model in detail.

### 3.1 Synchronous Communication Model

We closely follow the bounded-synchronous communication model in [9, 13, 27]. Here, the dealer is connected to every other party by a private, authenticated and bidirectional link. We do not require communication links between any two

non-dealer parties in $\mathcal{P}$. We further assume that all parties have access to a common broadcast channel that allows a party to send a message to all other parties and every party is assured that all parties have received the same message in the same round.

In the synchronous model, the distributed protocols operate in a sequence of rounds. In each *round*, a party performs some local computation, sends messages (if any) to the dealer through the private and authenticated link, and broadcasts some information over the broadcast channel. By the end of the round, it also receives all messages sent or broadcast by the other parties in the same round.

Along with being adaptive and $t$-bounded, we allow the adversary to be *rushing*: in every round of communication it can wait to hear the messages of the honest parties before sending (or broadcasting) its own messages. By round complexity of VSS, we mean the number of rounds in the sharing phase only, since all of our protocols ask for single round during reconstruction.

### 3.2 2-Round VSS for $n \geq 2t + 1$ from any Commitment

Here, we present a 2-round sharing and 1-round reconstruction VSS protocol for $n \geq 2t + 1$. Our 2-round VSS protocol allows any form of commitment. Feldman and Pedersen VSS schemes require three rounds for $n \geq 2t + 1$. The general structure of the sharing phase of their three round VSS schemes is: In the first (distribution) round, the dealer sends shares to parties and publishes a commitment on these shares. In the second round, parties may accuse (through broadcast) the dealer of sending inconsistent shares, which he resolves (through broadcast) in the third round. It is impossible to have distribution and accusation in the same round. Therefore, in order to reduce the number of rounds to two, the accusation and resolution rounds in VSS are collapsed into one round. To achieve this, the set of parties (in addition to dealer) performs some communication in the first round. We then employ a commitment-based modification of standard round-reduction technique from unconditional VSS protocols [11, Sect. 3.1]. It involves every party publicly committing to some randomness and sending that randomness to the dealer in the first round. The dealer uses this randomness as a blinding pad to broadcast the shares in the next round. Further, we use bivariate polynomial instead of univariate polynomials used in Feldman or Pedersen VSS. In the absence of homomorphism and without using bivariate polynomial, we do not know how the parties can check if the degree of a shared univariate polynomial is $t$ without using expensive NIZK proofs.

***Overview.*** In our 2-round protocol, dealer $D$ chooses a $t$-degree symmetric bivariate polynomial $F(x, y)$ such that $F(0, 0) = s$, the secret that he wants to distribute. Note that all of our protocols in this paper work also with the asymmetric bivariate polynomials. However, for ease of understanding, we always use *symmetric* polynomials in our descriptions. Dealer $D$ gives the univariate polynomial $f_i(x) = F(x, i)$ to every party $P_i$ and publicly commits to evaluations $f_i(j)$ for $j \in [1, n]$. As already mentioned, we allow every party to communicate to $D$ independently in the first round. Specifically, every party $P_i$ sends $n$ random

**Protocol 2-Round-VSS$(D, \mathcal{P}, s)$: Sharing Phase (Two Rounds)**

**Round 1:** Dealer $D$
- chooses a random symmetric bivariate polynomial $F(x, y)$ of degree-$t$ such that $F(0, 0) = s$
- computes $[\mathtt{Com}_{ij}, (f_{ij}, r_{ij})] = \mathsf{Commit}(f_{ij})$ for $i, j \in [1, n]$ and $i \geq j$, where $f_{ij} = F(i, j)$
- assigns $\mathtt{Com}_{ij} = \mathtt{Com}_{ji}$ and $r_{ij} = r_{ji}$ for $i, j \in [1, n]$ and $i < j$
- sends $(f_{ij}, r_{ij})$ to $P_i$ for $j \in [1, n]$ and broadcasts $\mathtt{Com}_{ij}$ for $i, j \in [1, n]$

Every other party $P_i$
- chooses two sets of $n$ random values $(p_{i1}, \ldots, p_{in})$ and $(g_{i1}, \ldots, g_{in})$.
- computes $[\mathtt{PCom}_{ij}, (p_{ij}, q_{ij})] = \mathsf{Commit}(p_{ij})$ and $[\mathtt{GCom}_{ij}, (g_{ij}, h_{ij})] = \mathsf{Commit}(g_{ij})$ for $j \in [1, n]$.
- sends $(p_{ij}, q_{ij})$ and $(g_{ij}, h_{ij})$ for $j \in [1, n]$ to $D$, and broadcasts $\mathtt{PCom}_{ij}$ and $\mathtt{GCom}_{ij}$ for $j \in [1, n]$.

**Round 2:** Dealer $D$, for every party $P_i$,
- verifies if $p_{ij} \stackrel{?}{=} \mathsf{Open}(\mathtt{PCom}_{ij}, p_{ij}, q_{ij})$ and $g_{ij} \stackrel{?}{=} \mathsf{Open}(\mathtt{GCom}_{ij}, g_{ij}, h_{ij})$ for $j \in [1, n]$
- broadcasts $(\alpha_{ij}, \beta_{ij})$ for all $j \in [1, n]$ such that $\alpha_{ij} = f_{ij} + p_{ij}$ and $\beta_{ij} = r_{ij} + g_{ij}$ if the verification succeeds, and broadcasts $(f_{ij}, r_{ij})$ for all $j \in [1, n]$ otherwise.

Party $P_i$
- verifies if $\deg(f_i(x)) \stackrel{?}{=} t$ and $f_{ij} \stackrel{?}{=} \mathsf{Open}(\mathtt{Com}_{ij}, f_{ij}, r_{ij})$ for $j \in [1, n]$, where $f_i(x)$ is the polynomial defined by $f_{ij}$s for $j \in [1, n]$.
- broadcasts nothing if the verifications succeeds, and broadcasts $(p_{ij}, q_{ij})$ and $(g_{ij}, h_{ij})$ for $j \in [1, n]$ otherwise.

$P_i$ is said to be `happy` if she broadcasts nothing, and considered `unhappy` otherwise.

**Local Computation:** Every party $P_k$
1. discards $D$ and halts the execution of 2-Round-VSS, if $D$ broadcasts
   - $\mathtt{Com}_{ij} \neq \mathtt{Com}_{ji}$ for some $i$ and $j$
   - $(f_{ij}, r_{ij})$ such that $f_{ij} \neq \mathsf{Open}(\mathtt{Com}_{ij}, f_{ij}, r_{ij})$ for some $i$ and $j$
   - $f_{ij}$ for $j = [1, n]$ that define polynomial of degree $> t$ for some $i$
   - $(f_{ij}, r_{ij})$ and $(f_{ji}, r_{ji})$ for some $i$ and $j$ such that $(f_{ij} \neq f_{ji})$ or $(r_{ij} \neq r_{ji})$
   - $(\alpha_{ij}, \beta_{ij})$ and $P_i$ broadcasts $(p_{ij}, q_{ij})$ and $(g_{ij}, h_{ij})$ such that $p_{ij} = \mathsf{Open}(\mathtt{PCom}_{ij}, p_{ij},\ q_{ij}), g_{ij} = \mathsf{Open}(\mathtt{GCom}_{ij}, g_{ij}, h_{ij})$ for all $j$; and $(f'_{ij} \neq \mathsf{Open}(\mathtt{Com}_{ij}, f'_{ij}, r'_{ij})$ or $\deg(f'_i(x)) > t)$ where $f'_{ij} = \alpha_{ij} - p_{ij}$, $r'_{ij} = \beta_{ij} - g_{ij}$ and $f'_i(x)$ is the polynomial defined by $f'_{ij}$s for $j \in [1, n]$.
2. discards an `unhappy` party $P_i$, if she broadcasts $p_{ij}$ and $g_{ij}$ for $j \in [1, n]$ such that $p_{ij} \neq \mathsf{Open}(\mathtt{PCom}_{ij}, p_{ij}, q_{ij})$ or $g_{ij} \neq \mathsf{Open}(\mathtt{GCom}_{ij}, g_{ij}, h_{ij})$ for some $j$. Let $\mathcal{Q}$ be the set of non-discarded parties.
3. outputs $(f_{kj}, r_{kj})$ for $j \in [1, n]$ as received in round 1, if $P_k$ is `happy` and in $\mathcal{Q}$. If she is `unhappy` and belongs to $\mathcal{Q}$ then she outputs $(f_{kj}, r_{kj})$ for $j \in [1, n]$ if they are broadcasted in round 2. Otherwise, $P_k$ computes $(f_{kj}, r_{kj})$ for $j \in [1, n]$ as $f_{kj} = \alpha_{kj} - p_{kj}$ and $r_{kj} = \beta_{kj} - g_{kj}$.

**Fig. 1.** Sharing Phase of Protocol 2-Round-VSS$(D, \mathcal{P}, s)$ for $n \geq 2t + 1$

values privately to $D$ and publicly commits them. At the end of the first round,

**Protocol 2-Round-VSS$(D, \mathcal{P}, s)$: Reconstruction Phase (One Round)**

1. Each $P_i$ in $\mathcal{Q}$ broadcasts $(f'_{ij}, r'_{ij})$ for $j \in [1, n]$

**Local Computation:** For every party $P_k$,

1. Party $P_i \in \mathcal{Q}$ is said to be *confirmed* if $\deg(f'_i(x)) = t$ and $f'_{ij} = \mathsf{Open}(\mathtt{Com}_{ij}, f'_{ij}, r'_{ij})$ for $j \in [1, n]$, where $f'_i(x)$ is the polynomial defined by $f'_{ij}$'s for all $j \in [1, n]$.
2. Consider $f'_i(x)$ polynomials of any $t+1$ *confirmed* parties. Interpolate $F'(x, y)$ and output $s' = F'(0, 0)$.

**Fig. 2.** Reconstruction Phase of Protocol 2-Round-VSS$(D, \mathcal{P}, s)$ for $n \geq 2t + 1$

every party checks the consistency of his received univariate polynomial with the commitments of $D$ and $D$ checks consistency of his received values with the corresponding commitments of the individual parties. The second round communication consists of only broadcasts. Any inconsistency between the public commitments and private values as well as the pairwise inconsistencies in the bivariate polynomial distribution (i.e, $f_i(j) \stackrel{?}{=} f_j(i)$) are sorted out in the second round. Note that there will be agreement among the parties at the end of local computation of sharing phase; i.e. every honest party knows if $D$ is discarded, otherwise every honest party has identical copy of $\mathcal{Q}$, the set of parties allowed to participate in the reconstruction phase.

In the reconstruction phase, every party discloses their respective univariate polynomials. They are verified with respect to the public commitments and the consistent polynomials are used for the reconstruction of the bivariate polynomial and consequently the committed secret $s$. We present the protocol in Fig. 1 and Fig. 2. We prove that the 2-Round-VSS protocol satisfies the stronger variant of VSS defined in Section 2.2.

**Theorem 1.** *Protocol 2-Round-VSS is a VSS scheme for $n \geq 2t + 1$.*

*Proof.* We prove the secrecy, correctness and strong commitment properties of VSS to show that the above theorem holds.

**Secrecy.** The secrecy of the scheme follows from the unconditional hiding property of the underlying commitment function and the property of symmetric bivariate polynomial. $D$'s public commitments $\mathtt{Com}_{ij}$'s will be uniformly distributed given the unconditional hiding property of the underlying commitment function. Moreover, the $\alpha_{ij}$, $\beta_{ij}$ values for $j \in [1, n]$ corresponding to honest $P_i$'s will be uniformly distributed. Now the secrecy of the constant term of the $D$'s degree-$t$ bivariate polynomial follows from the standard information-theoretic argument [27] against an adversary controlling at most $t$ parties, i.e.,

$$\Pr[\mathcal{A} \text{ computes } s | \{V_i \text{ for any } t \text{ parties}, \text{Public Information}\}] = \Pr[\mathcal{A} \text{ computes } s],$$

where $V_i$ represents all the information available at or computable by party $P_i$ at the end of the sharing phase.

**Correctness.** If $D$ is honest, then he will never be discarded. Moreover, all the honest parties will be `happy`. Now, correctness will follow if we show that a corrupted $P_i \in \mathcal{Q}$ is considered as *confirmed* only when she broadcasts correct polynomials in the reconstruction phase. Assume that corrupted $P_i$ is considered to be *confirmed* even when she broadcasts $f'_{ij}$ and $r'_{ij}$ for $j \in [1, n]$, where these values are not equal to $f_{ij}$ and $r_{ij}$ (as given by $D$). We can then devise an algorithm to break the computational binding property of the commitment function using this adversary. Therefore, given that the commitment function achieves computational binding, all the *confirmed* parties disclose proper $f_{ij}$ and $r_{ij}$ for $j \in [1, n]$. Therefore, every honest party will correctly reconstruct $F(x, y)$ and consequently $s = F(0, 0)$.

**Strong Commitment.** We have to consider the case of a corrupted $D$. If $D$ is discarded in the sharing phase, then every party may assume some default predefined value as $D$'s secret. So we consider the case when $D$ is not discarded.

Firstly, note that an honest party will never be discarded. Moreover at the end of sharing phase honest $P_i$ will output $n$ points (i.e. $f_{ij}$'s for all $j \in [1, n]$) on a degree-$t$ polynomial $f_i(x)$ and $n$ values $r_{ij}$ such that for every honest $P_j$, it holds that $f_{ij} = f_{ji}$ and $r_{ij} = r_{ji}$. We show this by considering all the three cases for any pair of honest parties $(P_i, P_j)$:

**If $P_i$ and $P_j$ are `happy`,** then we have $\texttt{Com}_{ij} = \texttt{Com}_{ji}$. Now $P_i$ verified consistency of $(\texttt{Com}_{ij}, f_{ij}, r_{ij})$, and $P_j$ verified consistency of $(\texttt{Com}_{ji}, f_{ji}, r_{ji})$. This implies the pair $(f_{ij}, r_{ij})$ is same as $(f_{ji}, r_{ji})$, unless corrupted $D$ had broken the binding property of the commitment function.

**If $P_i$ is `happy` and $P_j$ is `unhappy`,** then $(\texttt{Com}_{ij}, f_{ij}, r_{ij})$ is consistent and also $\texttt{Com}_{ij} = \texttt{Com}_{ji}$. For $P_j$, we have two cases: (1) $D$ has broadcasted $f_j(k)$ and $r_{jk}$ for $k \in [1, n]$; (2) $D$ broadcasted $\alpha_{ik}$, $\beta_{ik}$ for $k \in [1, n]$ and $P_j$ computed $f_{ik} = \alpha_{ik} - p_{ik}$, $r_{ik} = \beta_{ik} - g_{ik}$. However, in both the above cases, $f_{ik}$ and $r_{ik}$ are consistent with $\texttt{Com}_{jk}$ for all $k \in [1, n]$ (for otherwise $D$ would have been discarded). This also implies that tuple $(\texttt{Com}_{ji}, f_{ji}, r_{ji})$ is consistent. Again unless corrupted $D$ had broken the binding property of the commitment function, the pairs $(f_{ij}, r_{ij})$ and $(f_{ji}, r_{ji})$ are identical.

**If $P_i$ and $P_j$ are `unhappy`,** then $D$ would have been discarded if the pairs $(f_{ij}, r_{ij})$ and $(f_{ji}, r_{ji})$ are not identical.

So unless corrupted $D$ breaks the binding property of commitment function, the polynomials of the honest parties define symmetric bivariate polynomials, say $F(x, y)$. Now in the reconstruction phase, every honest party will be considered as *confirmed*. However, a corrupted party will be considered as *confirmed* if she broadcasts points on degree-$t$ polynomial $f_i(x) = F(x, i)$ (assuming she does not break binding of commitment function). Let $P_i$ broadcasts $n$ points, say $f'_{ij}$'s, corresponding to $f'_i(x)$ that is different from $f_i(x)$. Then $f_{ij}$ must be different from $f'_{ij}$ at least for one $j$ where $P_j$ is honest. Then $f'_{ij}$ will not be consistent with $\texttt{Com}_{ij}$ and $P_i$ will not be *confirmed*. Now it follows that the parties will reconstruct $D$'s committed secret $s = F(0, 0)$ in the reconstruction phase. □

The sharing phase of our 2-Round VSS protocol requires $O(n^2\kappa)$ bits of broadcast and $O(n^2\kappa)$ bits of private communication, while the reconstruction

phase requires $O(n^2\kappa)$ bits of broadcast. This communication complexity is at least a linear factor lower than the unconditional VSS schemes for $n \geq 2t+1$ [21]. On the other hand, it is also a linear factor higher than the communication complexity of 3-round Pedersen or Feldman VSS. This difference arises due to the use of bivariate polynomial in our protocol, which results from the lack of homomorphism in the commitment scheme under use. We suppose this increase in the communication complexity is a price paid for a reduction in the assumptions. In subsection 3.4, we present a more efficient VSS protocol using homomorphic commitments that has same communication complexity as Pedersen or Feldman VSS, but requires one less round of communication.

### 3.3 (Im)possibility Results for 1-Round VSS

Here, we prove the impossibility of 1-Round VSS except when $t = 1$ and $n \geq 4$, which lower-bounds computational VSS for $n \geq 2t + 1$ and any $t$ to a round complexity of *two*. Our 2-round protocol presented in the previous section thus has an optimal round complexity. Our results hold irrespective of computational or unconditional nature of the secrecy property.

**Theorem 2.** *1-round VSS is impossible for $t > 1$ and $n \geq 4$, irrespective of the number of rounds in the reconstruction phase.*

*Proof (Sketch).* The proof of this theorem is very similar to the proof of Theorem 7 of [23]. We prove the theorem by contradiction. So we assume that 1-round VSS, say $\Pi$, with $t = 2$ exists. Without loss of generality, we assume $D$ to be some party other than $P_1$. We then show that for any execution if party $P_1$ receives some particular piece of information from the dealer, then she will reconstruct a particular secret in the reconstruction phase irrespective of what $P_2, \ldots, P_n$ has received from the dealer. This of course allows us to show a breach of secrecy of $\Pi$, since $P_1$ could be the sole corrupted party and can distinguish the secret when he receives the particular information. We note that the proof does not make any assumption on the computational power of $P_1$ i.e. even a polynomial time $P_1$ can breach the secrecy. Since the proof strategy is very similar to the proof of Theorem 7 of [23], we skip the details here and present a detailed proof in the full version of the paper [1].

**Theorem 3.** *1-round VSS is impossible for $n \leq 3t$, irrespective of the number of rounds in the reconstruction phase.*

*Proof (Sketch).* This theorem is also proved by contradiction. In brief, we show that if such a scheme exists, then the the view of any $t$ parties in the sharing phase must determine the secret. This further implies a breach of secrecy, since adversary $\mathcal{A}$ can corrupt and coordinate any $t$ parties. A detailed proof appears in the full version of the paper [1].

In Theorem 3, we show that 1-round VSS is impossible for $n \leq 3t$, which implies the impossibility of 1-round VSS for $t = 1$ and $n \leq 3$. Further, in

Theorem 2, we show that 1-round VSS is impossible for $t > 1$ and $n \geq 4$. Therefore, 1-round VSS, if possible, will work for $t = 1$ and $n \geq 4$. We present a 1-round protocol in support of the corollary in the full version of the paper.

***VSS with an External Dealer.*** Here it can be shown that 1-round sharing VSS is impossible even in the presence of a single corruption apart from the dealer irrespective of the total number of parties and number of rounds in the reconstruction phase. Basically, we can follow the proof of Theorem 2 and arrive at the same contradiction while assuming $t = 1$ and the dealer is corrupted. Hence, we have the following theorem.

**Theorem 4.** *1-round VSS with external dealer is impossible for $t > 0$ irrespective of the number of parties and the number of rounds in reconstruction phase.*

### 3.4 An Efficient 2-round VSS using Homomorphic Commitments

We now present a 2-round sharing, 1-round reconstruction VSS protocol for $n \geq 2t + 1$ using homomorphic commitments. It has the same message and communication complexities as that of Feldman and Pedersen VSS schemes, and requires one less round of interaction. The protocol is similar to our 2-round protocol in Section 3.2; however, we do not need bivariate polynomials here.

Without loss of generality, we use the Pedersen commitment scheme as a representative homomorphic commitment scheme. In the sharing phase, dealer $D$ chooses two random degree-$t$ polynomials $f(x)$ and $r(x)$ such that $f(0) = s$. Dealer $D$ then sends $f_i = f(i)$ and $r_i = r(i)$ to each $P_i$ over the private links and broadcasts commitments on the coefficients of $f(x)$ (using the coefficients of $r(x)$ as random strings). By the end of the second round, every honest party must hold the correct point on the committed polynomial. To ensure that every $P_i$ sends two pairs $(p_i, q_i)$ and $(g_i, h_i)$ in $\mathbb{F}_p^2$ to dealer $D$ and publicly commits $p_i$ (using $q_i$ as a random element) and $g_i$ (using $h_i$ as a random element). Broadcasts and local computations in the second round are very similar to 2-Round-VSS in Section 3.2. The protocol is presented in Fig. 3. Similar to 2-Round-VSS, we note that there will be agreement among the parties at the end of local computation of sharing phase on whether $D$ is discarded or not. If $D$ is not discarded, then every honest party will have identical copy of $\mathcal{Q}$.

**Theorem 5.** *Protocol 2-Round-VSS-Hm is a VSS scheme for $n \geq 2t + 1$.*

The proof of the theorem closely follows from the proof of Theorem 1, and we include it in the full version of the paper.

The sharing phase requires $O(n\kappa)$ bits of communication over both the private links and the broadcast channel. The reconstruction phase requires $O(n\kappa)$ bits of communication over the broadcast channel.

## 4 VSS in the Asynchronous Communication Model

We now shift our focus to the asynchronous communication setting where VSS is possible for $n \geq 3t+1$. As we discuss in the related work, all known computa-

**Protocol 2-Round-VSS-Hm**$(D, \mathcal{P}, s)$

**Sharing Phase:** Two Rounds

**Round 1:**

1. $D$ selects two random polynomials $f(x)$ and $r(x)$ of degree-$t$, such that $f(0) = s$. Let $f(x) = a_0 + a_1x + \ldots + a_tx^t$ and $r(x) = b_0 + b_1x + \ldots + b_tx^t$.
2. For every $i \in [1, n]$, $D$ sends $f_i = f(i)$ and $r_i = r(i)$ to $P_i$ and broadcasts $\mathtt{Com}_i = \mathsf{Commit}(a_i, b_i)$ for $i = 0, \ldots, t$.
3. Every party $P_i$ sends two pairs $(p_i, q_i)$ and $(g_i, h_i)$ in $\mathbb{F}_p^2$ to $D$ and broadcasts commitments $\mathtt{PCom}_i = \mathsf{Commit}(p_i, q_i)$ and $\mathtt{GCom}_i = \mathsf{Commit}(g_i, h_i)$.

**Round 2:**

1. $D$ checks if $\mathtt{PCom}_i$ and $\mathtt{GCom}_i$ are consistent with the received pairs $(p_i, q_i)$ and $(g_i, h_i)$. If they are not consistent, then $D$ broadcasts $(f_i, r_i)$; else he broadcasts $\alpha_i = f_i + p_i$ and $\beta_i = r_i + g_i$.
2. Party $P_i$ checks if $\mathsf{Commit}(f_i, r_i) = \prod_{j=0}^{t} (\mathtt{Com}_i)^{i^j}$. If not, then $P_i$ broadcasts pairs $(p_i, q_i)$ and $(g_i, h_i)$, else she broadcasts nothing. Party $P_i$ is considered `happy` in the later case while she is `unhappy` in the former case.

**Local Computation:** Every party $P_k$

1. discards $D$ and halts the execution of 2-Round-VSS-Hm, if $D$ broadcasts
   (a) $f_i$, $r_i$ for some $i$ and $\mathsf{Commit}(f_i, r_i) \neq \prod_{j=0}^{t} (\mathtt{Com}_i)^{i^j}$.
   (b) $\alpha_i$, $\beta_i$; and $P_i$ broadcasts $(p_i, q_i)$ and $(g_i, h_i)$ such that $\mathtt{PCom}_i = \mathsf{Commit}(p_i, q_i)$ and $\mathtt{GCom}_i = \mathsf{Commit}(g_i, h_i)$; and $\mathsf{Commit}(f_i', r_i') \neq \prod_{j=0}^{t} (\mathtt{Com}_i)^{i^j}$ where $f_i' = \alpha_i - p_i$ and $r_i' = \beta_i - g_i$.
2. discards an `unhappy` party $P_i$ if she broadcasts $(p_i, q_i)$ and $(g_i, h_i)$ such that $\mathtt{PCom}_i \neq \mathsf{Commit}(p_i, q_i)$ or $\mathtt{GCom}_i \neq \mathsf{Commit}(g_i, h_i)$. Let $\mathcal{Q}$ be the set of non-discarded parties.
3. outputs $f_k$, $r_k$ as received from $D$ in round 1, if $P_k$ is in $\mathcal{Q}$ and `happy`. An `unhappy` $P_k$ in $\mathcal{Q}$ outputs $f_k$, $r_k$ if they are directly broadcasted by $D$ in round 2. Else $P_k$ computes $f_k$ and $r_k$ as $f_k = \alpha_k - p_k$ and $r_k = \beta_k - g_k$.

**Reconstruction Phase:** One Round

**Round 1:**

1. Each $P_i \in \mathcal{Q}$ broadcasts $f_i'$ and $r_i'$.

**Local Computation:** For every party $P_k$,

1. Party $P_i \in \mathcal{Q}$ is said to be *confirmed* if $\mathsf{Commit}(f_i', r_i') = \prod_{j=0}^{t} (\mathtt{Com}_i)^{i^j}$.
2. Consider $f_i'$ values of any $t + 1$ *confirmed* parties and interpolate $f'(x)$. Output $s' = f'(0)$.

**Fig. 3.** Protocol 2-Round-VSS-Hm for $n \geq 2t + 1$ with Homomorphic Commitments

tional VSS scheme [3,29,31] in the asynchronous communication setting rely on homomorphism of commitments. In this section, we show that homomorphism is not necessary for computational VSS in the asynchronous communication setting. We build our protocol from asynchronous VSS of [3] as it is the only generic and efficient asynchronous VSS scheme known in the literature. Further, with its $O(n^2)$ messages complexity, it is extremely efficient in terms of the number of messages. We modify this scheme so that it satisfies the VSS properties when the underlying commitment need not be homomorphic. This protocol does

not guarantee that every honest party receive his share of the secret. However, it guarantees that even a corrupted $D$ can not commit to $\perp$ instead of a secret from $\mathbb{F}_p$ (which is stronger than the basic definition given in section 2.2). We present another protocol in the full version that achieves the stronger definition where every party receives his share of the secret. Although this protocol increases the communication complexity by a linear factor in $n$, it is highly efficient in terms of communication when compared with the unconditional schemes [4, 5, 24, 25].

### 4.1 Asynchronous Communication Model

We follow the communication model of [3] and assume an asynchronous network of $n$ parties $P_1, \ldots, P_n$ such that every pair of parties is connected by an authenticated and private communication link. We work against a $t$-bounded adaptive adversary that we defined in Section 2.1. In the asynchronous communication setting, we further assume that the adversary controls the network and may delay messages between any two honest parties. However, it cannot read or modify these messages as the links are private and authenticated, and it also has to eventually deliver all the messages by honest parties. In the asynchronous communication setting, a VSS scheme has to satisfy the liveness and agreement properties (also called as the termination conditions) along with the secrecy, correctness and commitment properties described in Section 2.2.

**Liveness.** If the dealer $D$ is honest in the sharing phase, then all honest parties complete the sharing phase.

**Agreement.** If some honest party completes the sharing phase, then all honest parties complete the sharing phase eventually. If all honest parties subsequently start the reconstruction, then all honest parties will complete it.

### 4.2 VSS for $n \geq 3t + 1$ from any Commitment

We observe that VSS of [3] heavily relies on homomorphism of the underlying commitment schemes and does not satisfy VSS properties if we replace the homomorphic commitments by non-homomorphic commitments (agreement property will not be satisfied). The incapability stems from the fact that verifying the following with respect to non-homomorphic commitment is not easy: given commitments on $n$ values (associated with $n$ indices), the underlying values define a degree-$t$ polynomial. However, we find that with subtle enhancements to VSS of [3], one can obtain an asynchronous VSS protocol. In our enhanced protocol, a majority ($t + 1$ or more) of the honest parties receives proper share of the secret ($t$-degree univariate polynomial), while the remaining honest parties are assured that there are $t + 1$ or more honest parties that have received $t$-degree univariate polynomial and can complete the reconstruction phase. The message and communication complexities of our protocol are same as that of VSS of [3].

In our protocol, $D$ chooses a symmetric bivariate polynomial $F(x, y)$ satisfying $F(0, 0) = s$. He then computes an $n \times n$ commitment matrix, `Com` such that

**Protocol AsynchVSS**$(D, \mathcal{P}, s)$

**Sharing Phase:**

**Code for** $D$**:**

- Choose a random symmetric bivariate polynomial $F(x, y)$ of degree-$t$ such that $F(0, 0) = s$.
- Compute $[\mathtt{Com}_{ij}, (f_{ij}, r_{ij})] = \mathsf{Commit}(f_{ij})$ for $i, j \in [1, n]$ and $i \geq j$, where $f_{ij} = F(i, j)$.
- Assign $\mathtt{Com}_{ij} = \mathtt{Com}_{ji}$ and $r_{ij} = r_{ji}$ for $i, j \in [1, n]$ and $i < j$. Let $\mathtt{Com}$ be the $n \times n$ matrix containing $\mathtt{Com}_{ij}$ for $j \in [1, n]$ in the $i^{th}$ row.
- Send $(\mathtt{send}, \mathtt{Com}, f_i(x), r_i(x))$ to $P_i$, where $f_i(x) = F(x, i)$, $r_i(x)$ is the degree-$(n-1)$ polynomial defined by the points $((1, r_{i1}), \ldots, (n, r_{in}))$.

**Code for** $P_i$**:**

- On receiving $(\mathtt{send}, \mathtt{Com}, f_i(x), r_i(x))$ from $D$, send $(\mathtt{echo}, \mathtt{Com})$ to every $P_j$ if (a) $\mathtt{Com}$ is an $n \times n$ symmetric matrix and (b) $f_i(j) \stackrel{?}{=} \mathsf{Open}(\mathtt{Com}_{ij}, f_i(j), r_i(j))$.
- On receiving $(\mathtt{echo}, \mathtt{Com})$ from at least $2t + 1$ parties (possibly including itself) satisfying that $\mathtt{Com}$ received from $P_j$ is same as received from $D$, send $(\mathtt{ready}, \mathtt{share\text{-}holder}, \mathtt{Com})$ to every $P_j$, if you have already sent out $\mathtt{echo}$ messages.
- If you have not sent out any $\mathtt{ready}$ signal before:
  1. on receiving $\mathtt{ready}$ messages from at least $t+1$ $P_j$'s satisfying that $\mathtt{Com}$ received from $P_j$ is same as received from $D$, send $(\mathtt{ready}, \mathtt{share\text{-}holder}, \mathtt{Com})$ to every $P_j$, if you have already sent out $\mathtt{echo}$ messages.
  2. on receiving $(\mathtt{ready}, \mathtt{share\text{-}holder}, \mathtt{Com})$ from at least $t + 1$ $P_j$'s such that all the $\mathtt{Com}$ are same but do not match with the copy received from $D$, update your $\mathtt{Com}$ with this new matrix, delete everything else received from $D$ and send $(\mathtt{ready}, \star, \mathtt{Com})$ to every $P_j$.
- On receiving $\mathtt{ready}$ signals from at least $2t+1$ parties such that all of them contain same $\mathtt{Com}$ as yours and at least $t+1$ $\mathtt{ready}$ signals contain $\mathtt{share\text{-}holder}$, agree on $\mathtt{Com}$ and terminate.

**Reconstruction Phase:**

**Code for** $P_i$**:**

1. Send $(f_i(x), r_i(x))$ to every $P_j$ if you had sent $(\mathtt{ready}, \mathtt{share\text{-}holder}, \mathtt{Com})$ in the sharing phase.
2. Wait for $t+1$ $(f_j(x), r_j(x))$ messages such that $f_j(x)$ is degree-$t$ polynomial, $r_j(x)$ is degree-$(n-1)$ polynomial and $f_j(k) = \mathsf{Open}(\mathtt{Com}_{jk}, f_j(k), r_j(k))$ for all $k \in [1, n]$, interpolate $F(x, y)$ using those $t+1$ $f_j(x)$ polynomials, compute $s = F(0, 0)$ as the secret.

**Fig. 4.** Asynchronous VSS for $n \geq 3t + 1$ (optimal resilience)

$(i, j)^{th}$ entry in $\mathtt{Com}$ is the commitment on $F(i, j)$. Now $D$ delivers $f_i(x) = F(x, i)$ and $\mathtt{Com}$ to every $P_i$. In the rest of the protocol the parties try to agree on $\mathtt{Com}$ and check whether their polynomials are consistent with $\mathtt{Com}$ or not. We observe that the parties do not need to exchange and verify their common points on the

bivariate polynomial, given that agreement on Com can be achieved. Because, the parties can now perform local consistency checking of their polynomial with Com. In our protocol, some honest parties may not receive polynomials consistent with Com, however, they still help to reach agreement on Com sensing that majority of the honest parties have received a common Com and also the polynomials received by them are consistent with Com. We describe the protocol in Fig. 4.

**Lemma 1.** *If an honest party $P_i$ sends a* ready *message containing* Com *and a distinct honest party $P_j$ sends a* ready *message containing* $\overline{\texttt{Com}}$*, then* Com $= \overline{\texttt{Com}}$*.*

*Proof.* We prove this by contradiction. Let there exists an honest pair $(P_i, P_j)$ such that $\texttt{Com} \neq \overline{\texttt{Com}}$. The honest $P_i$ communicates ready with Com if: (a) it receives $(\texttt{echo}, \texttt{Com})$ from at least $2t+1$ parties OR b) it receives $(\texttt{ready}, \cdot, \texttt{Com})$ from at least $t+1$ parties, where $\cdot$ can be either share-holder or $\star$. Similar reasons apply for $P_j$ who sends $\overline{\texttt{Com}}$. If $P_i$ and $P_j$ send ready messages due to (a), then it implies that there is at least one honest party who communicates echo messages with Com as well as with $\overline{\texttt{Com}}$. This is impossible, since an honest party communicates echo with a unique matrix. For all other cases, we arrive at the contradiction that there is at least one honest party who sends echo with two different matrices or ready with two different matrices. We show this by considering the case when $P_i$ sends ready due to (a) and $P_j$ sends due to (b). The other cases will follow. $P_j$ sends ready due to (b) implies that there is at least one honest party, say $P_k$ who communicated ready with $\overline{\texttt{Com}}$ to $P_j$. Then by chain of arguments, we either get that honest $P_i$ has sent ready with $\overline{\texttt{Com}}$ or get an honest party (possibly including $P_i$) who communicates ready with $\overline{\texttt{Com}}$ due to (a). In both cases, we arrive at contradiction, since no honest party can send echo/ready with two different matrices. Hence, we prove the lemma. □

**Lemma 2.** *If some honest party $P_i$ has agreed on* Com*, then every honest party will eventually agree on* Com*.*

*Proof.* To prove the lemma, it is enough to prove the following: If some honest party $P_i$ has received $2t+1$ ready messages with Com such that at least $t+1$ of them contain share-holder, then every honest party will eventually receive the same. If $P_i$ receives ready messages as above, then there are at least $t+1$ honest parties who send out ready messages with Com and at least one of the honest party's ready message must contain share-holder. An honest party sends out ready with share-holder in two cases: (a) She received at least $2t+1$ echo message with Com and it has sent out echo with Com. Among these $2t+1$ parties $t+1$ are honest and they will eventually receive ready message from all the $t+1$ honest parties who also sent the same to $P_i$ (also by Lemma 2 if some honest party has sent a ready message with Com, then no other honest party will send ready with $\overline{\texttt{Com}}$). Hence these $t+1$ honest parties will eventually send out ready with share-holder. Hence eventually every honest party will receive $2t+1$ ready messages with Com such that at least $t+1$ of them contain share-holder. (b) She received at least $(t+1)$ ready messages with Com and she has sent out echo with Com. Among these $(t+1)$, there is at least one honest party, say $P_k$. If

$P_k$ has sent `ready` with `share-holder`, then by recursive argument this case will boil down to case (a). However if $P_k$ sends `ready` *without* `share-holder`, then he has received at least $t+1$ `ready` massages with `share-holder` which ensures existence of another honest $P_l$ who sent `ready` massage with `share-holder`. Now again by recursive argument, this case will boil down to case (a). □

**Lemma 3.** *If some honest party $P_i$ has agreed on `Com`, then there is a set $\mathcal{H}$ of at least $t+1$ honest parties each holding degree-$t$ polynomial $f_j(x)$ such that it is consistent with `Com` and there is a symmetric bivariate polynomial $F(x,y)$ such that $F(x,i) = f_i(x)$.*

*Proof.* If honest $P_i$ has agreed on `Com`, then she has received $2t+1$ `ready` messages with `Com` such that at least $t+1$ of them contain `share-holder`. From the previous proof, eventually $t+1$ honest parties (possibly including $P_i$) will eventually send out `ready` with `share-holder`. So there will be a set of at least $t+1$ honest parties who send out `ready` with `share-holder`. We claim that this set of honest parties, denoted by $\mathcal{H}$ will satisfy the conditions mentioned in the lemma statement. We notice that the honest parties in $\mathcal{H}$ never update `Com` and by previous lemma they eventually agree on the same. Also they send out `echo` well before sending out `ready`. This implies each honest party $P_i$ in $\mathcal{H}$ ensures that her polynomial $f_i(x)$ (i.e. the points on it) are consistent with `Com`. Now we proceed to show that there is a symmetric bivariate polynomial $F(x,y)$ such that $F(x,i) = f_i(x)$. This can be shown by showing for every pair $(P_i, P_j)$ from $\mathcal{H}$, $f_i(j) = f_j(i)$ holds good. This follows from the fact that $P_i$ and $P_j$ has same `Com` where they checked $\texttt{Com}_{ij} = \texttt{Com}_{ji}$ holds and then $P_i$ and $P_j$ individually ensured $f_i(j) \stackrel{?}{=} \mathsf{Open}(\texttt{Com}_{ij}, f_i(j), r_i(j))$ and $f_j(i) \stackrel{?}{=} \mathsf{Open}(\texttt{Com}_{ji}, f_j(i), r_j(i))$ respectively. If the above arguments do not hold then corrupted $D$ has broken binding property of underlying commitment, as he knows how to open $\texttt{Com}_{ij}$ in two different ways. □

**Theorem 6.** *Protocol AsynchVSS is an asynchronous VSS for $n \geq 3t+1$.*

*Proof.* **Liveness.** If $D$ is honest, then every honest party will eventually send out `echo` and then `ready` with `share-holder`. Since there are at least $2t+1$ honest parties, every honest party will eventually agree on `Com`.
**Agreement.** Agreement follows from Lemma 2.
**Correctness.** Correctness follows from Lemma 2 and 3. Honest dealer case is easy to follow. For a corrupted dealer the unique secret determined in the sharing phase is nothing but the constant term of $F(x,y)$ defined by $\mathcal{H}$ in Lemma 3. In the reconstruction phase, all the parties will reconstruct $D$'s secret using the polynomials sent by the honest parties in $\mathcal{H}$. Specifically, every honest party will definitely consider $f_j(x), r_j(x)$ sent by party $P_j$ in $\mathcal{H}$. However, we will be done if we show that any wrong degree-$t$ polynomial $\overline{f_j(x)}$ sent by a corrupted party $P_j$ will never be considered (unless corrupted $P_j$ breaks binding of commitment). This is ensured by the following check performed by an honest party before considering $P_j$'s polynomial for the reconstruction of $F(x,y)$: $f_j(k) = \mathsf{Open}(\texttt{Com}_{jk}, f_j(k), r_j(k))$ for all $k \in [1,n]$. This check ensures that $\overline{f_j(x)}$

must match with $f_j(x)$ at the $t+1$ positions corresponding to $\mathcal{H}$. But then it implies $\overline{f_j(x)} = f_j(x)$.
**Secrecy.** Follows from the properties of bivariate polynomial and the hiding of underlying commitment scheme. □

## 5 Discussion and Future Work

In this paper, we considered computational VSS as a standalone primitive. Our VSS schemes may also be easily leveraged in applications such as asynchronous Byzantine agreement protocols [5]. However, other VSS applications such as proactive share renewal and share recovery schemes [3, 18] and distributed key generation [12, 19] heavily rely on homomorphism of the commitments. It represents an interesting open problem if we can do better than in the unconditional case (e.g., [7]) for these applications. Further, most of the threshold cryptographic protocols also rely on homomorphism to verify the correctness. It will be interesting to check the feasibility of these threshold protocols based our VSS schemes without using expensive zero-knowledge proofs.

Finally, our schemes based on the definitional properties of commitments are expensive (by a linear factor) in terms of communication complexity in comparison to the respective schemes employing homomorphic commitments. It is worthwhile to study whether this gap in communication complexity is inevitable.

***Acknowledgements.*** We thank Jonathan Katz and our anonymous reviewers for their comments and suggestions on an earlier draft. We are also grateful to Ian Goldberg and Mehrdad Nojoumian for interesting initial discussions.

## References

1. Backes, M., Kate, A., Patra, A.: Computational Verifiable Secret Sharing Revisited. Cryptology ePrint Archive, Report 2011/281 (2011)
2. Blakley, G.R.: Safeguarding Cryptographic Keys. In: the National Computer Conference. pp. 313–317 (1979)
3. Cachin, C., Kursawe, K., A.Lysyanskaya, Strobl, R.: Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems. In: ACM CCS'02. pp. 88–97 (2002)
4. Canetti, R.: Studies in Secure Multiparty Computation and Applications. Ph.D. thesis, The Weizmann Institute of Science (1996)
5. Canetti, R., Rabin, T.: Fast Asynchronous Byzantine Agreement with Optimal Resilience. In: ACM STOC'93. pp. 42–51 (1993)
6. Chor, B., Goldwasser, S., Micali, S., Awerbuch, B.: Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In: IEEE FOCS'85. pp. 383–395 (1985)
7. D'Arco, P., Stinson, D.R.: On Unconditionally Secure Robust Distributed Key Distribution Centers. In: ASIACRYPT'02. pp. 346–363 (2002)
8. Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly secure message transmission. J. ACM 40(1), 17–47 (1993)
9. Feldman, P.: A Practical Scheme for Non-interactive Verifiable Secret Sharing. In: IEEE FOCS'87. pp. 427–437 (1987)

10. Fitzi, M., Garay, J.A., Gollakota, S., Rangan, C.P., Srinathan, K.: Round-Optimal and Efficient Verifiable Secret Sharing. In: TCC'06. pp. 329–342 (2006)
11. Gennaro, R., Ishai, Y., Kushilevitz, E., Rabin, T.: The round complexity of verifiable secret sharing and secure multicast. In: ACM STOC'01. pp. 580–589 (2001)
12. Gennaro, R., Jarecki, S., Krawczyk, H., Rabin, T.: Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. J. of Cryptology 20(1), 51–83 (2007)
13. Gennaro, R., Rabin, M.O., Rabin, T.: Simplified VSS and Fact-Track Multiparty Computations with Applications to Threshold Cryptography. In: ACM PODC'98. pp. 101–111 (1998)
14. Goldreich, O., Kahan, A.: How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. J. Cryptology 9(3), 167–190 (1996)
15. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. J. ACM 38(3), 691–729 (1991)
16. Haitner, I., Reingold, O.: Statistically-hiding commitment from any one-way function. In: ACM STOC'07. pp. 1–10 (2007)
17. Halevi, S., Micali, S.: Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. In: CRYPTO'96. pp. 201–215 (1996)
18. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In: CRYPTO'95. pp. 339–352 (1995)
19. Kate, A., Goldberg, I.: Distributed Key Generation for the Internet. In: Proc. Intl. Conf. on Distributed Computing Systems (ICDCS). pp. 119–128 (2009)
20. Katz, J., Koo, C., Kumaresan, R.: Improving the Round Complexity of VSS in Point-to-Point Networks. In: ICALP(2)'08. pp. 499–510 (2008)
21. Kumaresan, R., Patra, A., Rangan, C.P.: The Round Complexity of Verifiable Secret Sharing: The Statistical Case. In: ASIACRYPT'10. pp. 431–447 (2010)
22. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect Zero-Knowledge Arguments for $P$ Using Any One-Way Permutation. J. Cryptology 11(2), 87–108 (1998)
23. Patra, A., Choudhary, A., Rabin, T., Rangan, C.P.: The Round Complexity of Verifiable Secret Sharing Revisited. In: CRYPTO'09. pp. 487–504 (2009)
24. Patra, A., Choudhary, A., Rangan, C.P.: Efficient Asynchronous Byzantine Agreement with Optimal Resilience. In: ACM PODC'09. pp. 92–101 (2009)
25. Patra, A., Choudhary, A., Rangan, C.P.: Efficient Statistical Asynchronous Verifiable Secret Sharing with Optimal Resilience. In: ICITS'09. pp. 74–92 (2009)
26. Pedersen, T.P.: A Threshold Cryptosystem without a Trusted Party. In: Eurocrypt'91. pp. 522–526. Springer-Verlag (1991)
27. Pedersen, T.P.: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: CRYPTO'91. pp. 129–140 (1991)
28. Rabin, T., Ben-Or, M.: Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (Extended Abstract). In: ACM STOC'89. pp. 73–85 (1989)
29. Schultz, D.A., Liskov, B., Liskov, M.: MPSS: Mobile Proactive Secret Sharing. ACM Trans. Inf. Syst. Secur. 13(4), 34 (2010)
30. Shamir, A.: How to Share a Secret. Commun. ACM 22(11), 612–613 (1979)
31. Zhou, L., Schneider, F.B., van Renesse, R.: APSS: Proactive Secret Sharing in Asynchronous Systems. ACM Trans. Inf. Syst. Secur. 8(3), 259–286 (2005)