Computational Verifiable Secret Sharing Revisited

Michael Backes
Saarland University and MPI-SWS
Germany
backes@mpi-sws.org

Aniket Kate
MPI-SWS
Germany
aniket@mpi-sws.org

Arpita Patra*
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 paper, we demonstrate that homomorphism of commitments is not a necessity for computational VSS in the synchronous or in the asynchronous communication setting. We present new VSS schemes based only on the definitional properties of commitments that are almost as good as existing VSS schemes based homomorphic commitments. Furthermore, they have significantly lower communication complexities than their (statistical or perfect) unconditional counterparts. Considering the feasibility of commitments from any claw-free permutation, one-way function or collision-resistant hash function, our schemes can be an excellent alternative to unconditional VSS in the future.

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 paper, 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 $n \leq 3t$. For the remaining condition of $t = 1$ and $n \geq 4$, we present a one-round VSS scheme. 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

1 Introduction

The notion of secret sharing was introduced independently by Shamir [38] and Blakley [3] 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

*The author is supported by Center for Research in Foundations of Electronic Markets (CFEM), Denmark.

of the dealt values. To meet this requirement, Chor et al. [9] introduced verifiability in secret sharing, which led to the concept of *verifiable secret sharing* (VSS).

VSS has remained an important area of cryptographic research for the last two decades [1,5,11–13,15,24–26,29,33,34]. 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 [5, 11, 15, 33] 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. For the later case, we show homomorphism of commitment schemes is not a necessity for VSS and thereby VSS can be constructed based on the definitional properties of commitment schemes.

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 [15, 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 [34] and few other commitment schemes. Results such as [39] increase our natural curiosity towards avoiding use of homomorphic discrete logarithm and Pedersen commitments. 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, they are considerably better than the unconditional VSS schemes. Therefore, if the existing computational VSS schemes become ineffective in the future possibly due to [39], our schemes will be more suitable than their unconditional counterparts in applications such as asynchronous Byzantine agreement protocols.

In the synchronous communication model with a broadcast channel, Gennaro et al. [13] 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 further extended in [12,25] with tight polynomial time constructions, and in [26, 29] by improving the bounds in a statistical scenario where the VSS properties are held *statistically* and can be violated with a negligible probability.

To the best of our knowledge, 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 [11, 34] for the optimal resilience of $n \geq 2t + 1$ is the same as that of unconditional VSS schemes: three rounds in the sharing phase.[1] 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 present a 1-round construction for $n \geq 2t + 1$ for the weaker notion of weak verifiable secret sharing (WSS), which might be of some theoretical interest. We note that our 1-round computational VSS does not differ from 1-round statistical VSS [29] in terms of possibility results.

[1] Note that it is possible to reduce a round in sharing in [11, 34] 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 [18].

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 [11, 15, 34] based on homomorphic commitments require three rounds for $n \geq 2t+1$. Comparing with unconditional VSS schemes, we notice that the message and communication complexities of our scheme are at least a linear factor better. 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 same message and communication complexities but requires one less round of communication as compared to [11,15,34]. Our scheme, therefore, is an ideal replacement for [11,15,34] in each of their applications.

Organization. In the rest of this section, we review the related work. In Section 2, we describe our adversary model and definitions of verifiable secret sharing and cryptographic commitments. For the sake of clarity, we present all our results on VSS in the synchronous communication model in Section 3. This section starts with a 2-round VSS scheme based on any commitment scheme in subsection 3.2, followed by the impossibility results for 1-round VSS in subsection 3.3 that implicitly show the round optimality of our 2-round protocol. We complete this section with a 1-round VSS for the special case of ($t = 1$ and $n \geq 4$) and an efficient 2-round VSS based on homomorphic commitment that improves the communication complexity of our 2-round VSS of subsection 3.2. We then consider VSS in asynchronous communication model in Section 4. Here, we present our *asynchronous* VSS protocol using any commitments with optimal resilience, i.e., $n \geq 3t+1$. In Section 5, we discuss a few interesting open problems. In Appendix A, we present an efficient one-round WSS scheme for $n \geq 2t+1$. Most of our proofs have been shifted to Appendix B.

Related Work. For our work in the synchronous setting, we closely follow the network and adversary model of the best known VSS schemes: Feldman VSS [11] and Pedersen VSS [34]. 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 channel. 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. This network relaxation is an advantage of computational VSS over unconditional VSS as Pedersen [34] proved that unconditional VSS schemes are impossible in the network where only the dealer is connected to the parties by private communication channel and a common broadcast medium is available.

It is also important to compare our work 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. Assuming a broadcast channel, perfect VSS is possible if and only if $n \geq 3t+1$ [2], while statistical VSS is possible for $n \geq 2t+1$ [36]. Gennaro et al. [13] initiated the study of the round complexity of unconditional VSS, which was extended by Fitzi et al. [12] and Katz et al. [25]. 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. [29] show that $n \geq 3t+1$ is necessary and sufficient for 2-round statistical VSS. Recently, Kumaresan et al. [26] 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 in this paper, the best known computational VSS protocols [11, 15, 34] 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 among these that does not mandate homomorphic commitments, [15, App. A], does not satisfies the generally required stronger commitment property of VSS. 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. We note that it is also possible to achieve 1-round VSS in the presence of a public-key infrastructure (PKI) employing NIZK proofs [18]. However, NIZK proofs requires a common reference string or a random oracle. Furthermore, the scheme of [18] 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. [5]. In the asynchronous communication setting, Cachin et al. [5], Zhou et al. [40], and more recently Schultz et al. [37]

suggested computational VSS schemes: AVSS (Asynchronous VSS), APSS (Asynchronous Proactive Secret Sharing), and MPSS (Mobile Proactive Secret Sharing) respectively. Of these, the APSS protocol is impractical for any reasonable n and t, as it has an exponential $\binom{n}{t}$ factor in the message complexity (number of messages transferred), while MPSS is developed for a more mobile setting where the set of the system nodes has to change completely between two consecutive phases. On other hand, AVSS cleverly assimilates a bivariate polynomial into Bracha's reliable broadcast [4] to construct an AVSS scheme with $O(n^2)$ message complexity and $O(n^3)$ communication complexity. Therefore, AVSS is certainly the most practical computational VSS protocol in the literature. However, all of the above schemes rely on homomorphism of the commitment scheme. We avoid the use of homomorphism, while maintaining the communication complexity of the VSS of [5]. Note that our protocol is significantly efficient in all aspects as compared to unconditional VSS schemes [6, 7, 30, 31] in the asynchronous communication model.

2 Preliminaries

We work in the computational security setting, where κ denotes the security parameter of the system, in bits. A function $\epsilon(\cdot) : \mathbb{N} \rightarrow \mathbb{R}^+$ is called *negligible* if for all $c > 0$ there exists a κ_0 such that $\epsilon(\kappa) < 1/\kappa^c$ for all $\kappa > \kappa_0$. In the rest of this paper, $\epsilon(\cdot)$ denotes a negligible function.

We assume that the dealer's secret s lies over a finite field $\mathbb{F}_p$, where p is an κ 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 $\{1, \ldots, n\}$.

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. A party is called *honest*, if it is not under the adversarial control.

We work in the synchronous as well as the asynchronous communication settings in this paper, and postpone the discussions on communication setting to the respective sections. We describe the synchronous communication model in Section 3.3 and the asynchronous communication model in Section 4.

2.2 VSS and Variants

We now present the definition of VSS [13]. 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. The sharing phase may consist of several rounds of interaction between parties. 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 $\mathsf{Rec}(v_1; \ldots; v_n)$ 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^*.

In the asynchronous communication setting, VSS also has two more termination properties: *liveness* and *agreement*. We describe them in Section 4, before we present our asynchronous VSS protocols.

A VSS protocol is considered *efficient* if the total computation and communication performed by all parties is polynomial in n and the security parameter κ. The optimal resiliency bound for VSS is $n \geq 2t+1$ [17] in the synchronous communication model and $n \geq 3t+1$ [2] in the asynchronous communication model.

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 the security 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.

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 $s^* \in \mathbb{F}_p$. As suggested in [13, 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, each party locally outputs a share of the secret chosen only from $\mathbb{F}_p$ at the end of the sharing phase, such that the joint shares output by honest parties are consistent with a specified secret sharing scheme.

   Assuming 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 $f(i)$.

   Our 1-round protocol in Section 3.4 and the first asynchronous protocol in Section 4.2 satisfy the basic VSS definition. On the other hand, our 2-round protocols in sections 3.2 and 3.5, and our second asynchronous protocol in Section 4.3 satisfy 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 1-round VSS in Section 3.4 hold 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.

In the VSS literature, a strictly weaker primitive called weak verifiable secret sharing (WSS) has also been studied. It is generally used as a stepping stone toward the main goal of obtaining a VSS scheme. In Appendix A, we discuss our findings about WSS in the computational setting. Also, note that 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 briefly discuss in Section 5.

2.3 Commitment Schemes

Commitment schemes are important components of many cryptographic protocols. 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 a specific message m (*binding*) without revealing it (*hiding*). The function *may* output an opening value (or a witness) 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) binding and hiding at the same time, and the impossibility also holds in the statistical case. As a result, commitments come in two dual flavors:

perfect (or statistical) binding but computational hiding commitments, and perfect (or statistical) hiding but computational binding commitments. In many applications of commitments, they may never be opened or opened only after a long time. In such scenarios, the binding property must be satisfied while the application is running, however the committed values should remain hidden forever, and thus commitments of the second type are generally considered advantageous as compared to commitments of the first type.

Perfect hiding but computational binding (under the DLog assumption) Pedersen commitment [34] is most commonly used 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, Pedersen commitment will not be suitable once quantum computers arrive. On the other hand, commitments of both types can be achieved from any one-way function [19–21, 23, 27]. In this paper, we concentrate and use the commitments of the second type, whose efficient constructions are possible from any claw-free permutation [8, 16], any one-way permutation [28] or any collision-free hash function [22]. Note that, along with being non-homomorphic, some of the above commitment constructions are also interactive in the nature and involve two or more rounds of communication. However, we restrict ourselves to the non-interactive commitment constructions as the interactive commitment constructions may increase the rounds complexity of the 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 with private and authenticated links and a broadcast channel [11,15,34]. Here, the dealer is connected to every other party by a private, authenticated and bidirectional link. Note that 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 satisfies the terminating reliable broadcast properties [32]: it 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.

In our synchronous communication model, 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 and reconstruction phases of any execution. Although, it is possible to have more than one round during the reconstruction phase [29], all of our protocols ask for single round during reconstruction. Therefore, in this paper, we denote the round complexity of a VSS protocol as the number of rounds in its sharing phase.

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

In this section, 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.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 [13, 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 values privately to D and publicly commits them. At the end of the first round, 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. if D is discarded, then every honest party knows it, and similarly, every honest party will have identical copy of $\mathbb{Q}$, the set of parties allowed to take part in the reconstruction phase.

In the reconstruction phase, every party discloses (or broadcasts) 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 Figure 1. We prove that the 2-Round-VSS protocol satisfies the stronger variant of VSS defined in Section 2.2.

Theorem 3.1. *Protocol 2-Round-VSS is a 2-round computational VSS scheme for $n \geq 2t + 1$.*

We prove the secrecy, correctness and strong commitment properties of VSS to show that the above theorem holds. For a detailed proof, refer to Appendix B.

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. 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 3.5, 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 2. 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 3.2. *1-round VSS is impossible for $t > 1$ and $n \geq 4$, irrespective of the number of rounds in the reconstruction phase.*

Proof Outline. The proof of this theorem is very similar to the proof of Theorem 7 of [29]. We prove the theorem by contradiction. So we assume that 1-round VSS, say Π, 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 Π, since P_1 could be the sole corrupted party and can distinguish the secret when he receives the particular information.

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 $i, j \in [1, n]$.
- 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 $j \in [1, n]$
- 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, while considered `unhappy` otherwise.

Local Computation: Every party P_k

1. discards D and halts the execution of 2-Round-VSS, if
   - $\mathtt{Com}_{ij} \neq \mathtt{Com}_{ji}$ for some i and j
   - D broadcasts (f_{ij}, r_{ij}) such that $f_{ij} \neq \mathsf{Open}(\mathtt{Com}_{ij}, f_{ij}, r_{ij})$ for some i and j
   - D broadcasts f_{ij} for $j = [1, n]$ such that they define polynomial of degree $> t$ for some i
   - D broadcasts (f_{ij}, r_{ij}) and (f_{ji}, r_{ji}) for some i and j such that $(f_{ij} \neq f_{ji})$ or $(r_{ij} \neq r_{ji})$
   - D broadcasts $(\alpha_{ij}, \beta_{ij})$ 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 $\mathbb{Q}$ 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 $\mathbb{Q}$. If she is `unhappy` and belongs to $\mathbb{Q}$ 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 $r_{kj} = \beta_{kj} - g_{kj}$.

Reconstruction Phase: One Round

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

Local Computation: For every party P_k,

1. Party $P_i \in \mathbb{Q}$ 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 $s' = F'(0, 0)$.

Figure 1: 2-Round VSS for $n \geq 2t + 1$

Now for proving the above fact we use a hybrid argument and the fact that two parties can be corrupted (possibly including the dealer). So we start with an honest execution G of Π for a secret s^G. Naturally, in this case the views of the parties in reconstruction phase will reconstruct secret s^G. Next we claim that if the view of P_n is replaced by any arbitrary view and the rest of the parties view remain same, the reconstruction phase still will output the same secret s^G. This is justified by the correctness property of Π, as D could be honest and P_n could be the corrupted party who inputs an arbitrary view in the reconstruction phase. We then continue to claim that if the views of P_{n-1} and P_n are replaced by any arbitrary views and the rest of the parties' view remain same, the reconstruction phase will still output the same secret s^G. This is argued as follows: Assume D is corrupted and he distributes proper information to all the parties (proper means as per G which is assumed to be an honest execution) except P_n to whom he simply delivers arbitrary information. Assuming P_n to be honest, the joint views in the reconstruction phase of this execution will be identical to when D was honest and P_n was corrupted. Hence s^G should be reconstructed. Now assume that apart from D, P_{n-1} is also corrupted who inputs wrong view in the reconstruction phase. By commitment of Π, the arbitrary junk views of P_{n-1} (who is corrupted and inputs junk view) and P_n (who is honest but receives junk view from D in sharing phase) do not stop s^G to be reconstructed. This is exactly the place where we exploit our assumption that $t = 2$. In the same way as above, we can show that if the views of P_{n-2}, P_{n-1} and P_n are replaced by any arbitrary views and rest of the parties' view remain same, still the reconstruction phase outputs the same secret s^G. We may proceed this way to finally prove that even junk views of parties $P_2, \ldots, P_n$ do not stop the reconstruction of s^G. This clearly implies that the views of $P_2, \ldots, P_n$ are independent of secret s^G and the view of P_1 solely determines the secret. Now if P_1 is the corrupted party, then the secrecy of Π is no longer guaranteed, a contradiction to the fact that Π is a VSS protocol. For a detailed proof, refer to Appendix B. □

Theorem 3.3. *1-round VSS is impossible for $n \leq 3t$, irrespective of the number of rounds in the reconstruction phase.*

Proof Outline. 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. For a detailed proof, refer to Appendix B. □

In Theorem 3.3, 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 3.2, 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 $n \geq 4$.

Corollary 3.4. *1-round VSS is possible only if $t = 1$ and $n \geq 4$.*

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 3.2 and arrive at the same contradiction while assuming $t = 1$ and the dealer is corrupted. Hence, we have the following theorem.

Theorem 3.5. *1-round VSS with an external dealer is impossible for $t > 0$ irrespective of the number of parties and the number of rounds in the reconstruction phase.*

3.4 1-Round VSS for $t = 1$ and $n \geq 4$

In this section, we prove the sufficiency of Corollary 3.4 by describing a simple 1-round VSS protocol when $t = 1$, $n \geq 4$ and $D \in \mathcal{P}$.

In the sharing phase, dealer $D \in \mathcal{P}$ does a Shamir-sharing of its secret and publicly commits the individual shares held by the parties using efficient commitment schemes. Since there is a single round in the sharing phase, parties may not be able to raise any accusation against the dealer in case the shares are not consistent with respect to D's public commitment. In spite of this, our protocol achieves the correctness and commitment properties of VSS . The key aspect is that D cannot participant in the reconstruction phase. Now the unique secret committed by the dealer D is defined to be $\perp$ if shares of two or more reconstructing parties are not consistent with the public commitment, or the consistent shares define a polynomial of degree

Protocol 1-Round-VSS$(D, \mathcal{P}, s)$

Sharing Phase: One Round

1. $D \in \mathcal{P}$ selects a random polynomial $f(x)$ of degree one, such that $f(0) = s$.
2. For every $i \in [1, n]$, D computes $[\mathcal{C}_i, (f_i, r_i)] = \mathsf{Commit}(f_i)$ where $f_i = f(i)$. He then sends (f_i, r_i) to P_i over the private channel and broadcasts C_i for $i \in [1, \ldots, n]$.

Reconstruction Phase: One Round— D is not allowed to participate

1. Each party P_i broadcasts (f_i, r_i).

Local Computation: For every party P_k,

1. Party $P_i \in \mathcal{P} \setminus \{D\}$ is said to be *confirmed* if $f_i \stackrel{?}{=} \mathsf{Open}(\mathcal{C}_i, f_i, r_i)$.
2. If there are two or more *non-confirmed* parties then output $\perp$. Otherwise, interpolate f_i of all *confirmed* parties to a polynomial $f'(x)$. If $\deg(f') \leq 1$ then output $s = f'(0)$ else output $\perp$.

Figure 2: 1-Round VSS for $t = 1$ and $n \geq 4$

more than t. Otherwise, the unique secret is the constant term of the polynomial defined by the consistent shares. A detailed description of the protocol appears in Figure 2.

Theorem 3.6. *Protocol 1-Round-VSS is a 1-round computational VSS scheme for $t = 1$ and $n \geq 4$.*

We prove the secrecy, correctness and commitments properties of VSS to prove the theorem. For a detailed proof, refer to Appendix B. This 1-Round VSS protocol is efficient in terms of the communication complexity. The sharing phase requires $O(n\kappa)$ bits of broadcast and $O(n\kappa)$ bits of private communication, while the reconstruction phase requires $O(n\kappa)$ bits of broadcast.

3.5 An Efficient 2-round VSS using Homomorphic Commitments

Here we present a 2-round sharing, 1-round reconstruction VSS protocol for $n \geq 2t + 1$ using homomorphic commitments. As already mentioned, it has exactly the same message and communication complexity as that of Feldman and Pedersen VSS schemes and requires one less round of interaction for $n \geq 2t + 1$. Conceptually this protocol is similar to our 2-round protocol in Section 3.2 except that we do not need bivariate polynomials here. We present a brief overview of the protocol and the protocol description below, and move the proofs to Appendix B.

Overview. 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 our 2-Round-VSS protocol in Section 3.2. The protocol is now presented in Figure 3. Before we proceed to prove the properties of the protocol, we first note that there will be agreement among the parties at the end of local computation of sharing phase. That is, if D is discarded, then every honest party knows it. Similarly, every honest party will have identical copy of $\mathbb{Q}$.

Theorem 3.7. *Protocol 2-Round-VSS-Hm in Figure 3 is a 2-round VSS scheme for $n \geq 2t + 1$.*

For a proof, refer to Appendix B.

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.

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 $f(x) = a_0 + a_1 x + \ldots + a_t x^t$ and $r(x) = b_0 + b_1 x + \ldots + b_t x^t$.
2. For every $i \in [1, n]$, D sends $f_i = f(i)$ and $r_i = r(i)$ to P_i and broadcasts $\texttt{Com}_i = \mathsf{Commit}(a_i, b_i)$ 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 $\texttt{PCom}_i = \mathsf{Commit}(p_i, q_i)$ and $\texttt{GCom}_i = \mathsf{Commit}(g_i, h_i)$.

Round 2:

1. D checks if $\texttt{PCom}_i$ and $\texttt{GCom}_i$ 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 $\mathsf{Commit}(f_i, r_i) = \prod_{j=0}^{t} (\texttt{Com}_i)^{i^j}$. 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 at the end of Round 2: 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 $\mathsf{Commit}(f_i, r_i) \neq \prod_{j=0}^{t} (\texttt{Com}_i)^{i^j}$.
   (b) α_i, β_i; and P_i broadcasts (p_i, q_i) and (g_i, h_i) such that $\texttt{PCom}_i = \mathsf{Commit}(p_i, q_i)$ and $\texttt{GCom}_i = \mathsf{Commit}(g_i, h_i)$; and $\mathsf{Commit}(f_i', r_i') \neq \prod_{j=0}^{t} (\texttt{Com}_i)^{i^j}$ 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 $\texttt{PCom}_i \neq \mathsf{Commit}(p_i, q_i)$ or $\texttt{GCom}_i \neq \mathsf{Commit}(g_i, h_i)$. Let $\mathbb{Q}$ 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 $\mathbb{Q}$ and `happy`. An `unhappy` P_k in $\mathbb{Q}$ 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

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

Local Computation: For every party P_k,

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

Figure 3: 2-Round VSS for $n \geq 2t + 1$ using Homomorphic Commitments

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 computational VSS scheme [5, 37, 40] 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 [5] 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. However this protocol does not guarantee that every honest party receive their shares of the secret. Therefore, we present another protocol that achieves this stronger definition using the previous one as a building block. Our final VSS although increases the communication complexity by a linear factor in n, it is still highly efficient in all complexity measures as compared to the unconditional asynchronous VSS schemes [6, 7, 30, 31].

4.1 Asynchronous Communication Model

We follow the communication model of [5] 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 will eventually complete the sharing phase. If all honest parties subsequently start the reconstruction phase, then all honest parties will complete the reconstruction phase.

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

We observe that VSS of [5] heavily relies on homomorphism of the underlying commitment schemes. It is not even a WSS scheme if we replace the homomorphic commitments by non-homomorphic commitments as the agreement property is not 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 [5], one can obtain an asynchronous VSS protocol that satisfies the standard VSS definition in Section 2.2. 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 the exactly same as that of VSS of [5].

In our protocol, D, as usual, chooses a symmetric bivariate polynomial $F(x, y)$ satisfying $F(0, 0) = s$. He then computes an $n \times n$ commitment matrix, $\texttt{Com}$ such that $(i, j)^{th}$ entry in $\texttt{Com}$ is the commitment on $F(i, j)$. Now D delivers $f_i(x) = F(x, i)$ and $\texttt{Com}$ to every P_i. In the rest of the protocol the parties try to agree on $\texttt{Com}$ and check whether their polynomials are consistent with $\texttt{Com}$ 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 $\texttt{Com}$ can be achieved. Because, the parties can now perform local consistency checking of their polynomial with $\texttt{Com}$. In our protocol, some honest parties may not receive polynomials consistent with $\texttt{Com}$, however, they still help to reach agreement on $\texttt{Com}$ sensing that majority of the honest parties have received a common $\texttt{Com}$ and also the polynomials received by them are consistent with $\texttt{Com}$. We describe the protocol in Figure 4.

Lemma 4.1. *Let P_i be the first honest party to send $\texttt{ready}$ message containing $\texttt{Com}$. Then for every other honest party P_j that sends $\texttt{ready}$ message containing $\overline{\texttt{Com}}$, $\overline{\texttt{Com}} = \texttt{Com}$.*

Proof. We prove this by contradiction. Let $\texttt{Com} \neq \overline{\texttt{Com}}$. The honest P_i has communicated $\texttt{Com}$ after receiving $(\texttt{echo}, \texttt{Com})$ from at least $2t + 1$ parties in which at least $t + 1$ are honest. Note that an honest party sends $\texttt{echo}$ and $\texttt{ready}$ messages to all parties including *itself*. P_j has communicated $\overline{\texttt{Com}}$ after one of the following two events has occurred. We show that in every case we arrive at a contradiction: (a) P_j received $(\texttt{echo}, \overline{\texttt{Com}})$ from at least $2t+1$ parties: This implies that there is at least $t+1$ parties who communicated $\texttt{echo}$ signals of two types, one type containing $\texttt{Com}$ and another type containing $\overline{\texttt{Com}}$. However, this implies that the *honest* parties in the set of those $t + 1$ parties communicate $\texttt{echo}$ message of two types. This is impossible. (b) P_j received $(\texttt{ready}, \cdot, \overline{\texttt{Com}})$ from at least $t + 1$ parties, where $\cdot$ can be either $\texttt{share-holder}$ or $\star$: this implies that there is at least one honest party, say P_k who communicated the above to P_j. By a chain of arguments, this case also boils down to the case that there must be some honest party who communicated $\texttt{echo}$ signals of two types, which is a contradiction. Hence, we prove the lemma. □

Lemma 4.2. *If some honest party P_i has agreed on $\texttt{Com}$, then every honest party will eventually agree on $\texttt{Com}$.*

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

Reconstruction Phase:
Code for P_i**:**

1. Send $(f_i(x), r_i(x))$ to every P_j if you had sent $(\texttt{ready}, \texttt{share-holder}, \texttt{Com})$ 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 $f_j(k) = \mathsf{Open}(\texttt{Com}_{jk}, f_j(k), r_j(k))$ 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.

Figure 4: Asynchronous VSS for $n \geq 3t + 1$ (optimal resilience)

Proof. To prove the lemma, it is enough to prove the following: If some honest party P_i has received $2t + 1$ $\texttt{ready}$ messages with $\texttt{Com}$ such that at least $t + 1$ of them contain $\texttt{share-holder}$, then every honest party will eventually receive the same. If P_i receives $\texttt{ready}$ messages as above, then there are at least $t + 1$ honest parties who send out $\texttt{ready}$ messages with $\texttt{Com}$ and at least one of the honest party's $\texttt{ready}$ message must contain $\texttt{share-holder}$. An honest party sends out $\texttt{ready}$ with $\texttt{share-holder}$ in two cases: (a) She received at least $2t + 1$ $\texttt{echo}$ message with $\texttt{Com}$ and it has sent out $\texttt{echo}$ with $\texttt{Com}$. Among these $2t + 1$ parties $t + 1$ are honest and they will eventually receive $\texttt{ready}$ message from all the $t + 1$ honest parties who also sent the same to P_i. Hence these $t + 1$ honest parties will eventually send out $\texttt{ready}$ with $\texttt{share-holder}$. Hence eventually every honest party will receive $2t + 1$ $\texttt{ready}$ messages with $\texttt{Com}$ such that at least $t + 1$ of them contain $\texttt{share-holder}$. (b) She received at least $(t + 1)$ $\texttt{ready}$ messages with $\texttt{Com}$ and she has sent out $\texttt{echo}$ with $\texttt{Com}$. Among these $(t + 1)$, there is at least one honest party, say P_k. If P_k has sent $\texttt{ready}$ with $\texttt{share-holder}$, then by recursive argument this case will boil down to case (a). However if P_k sends $\texttt{ready}$ *without* $\texttt{share-holder}$, then he has received at least $t + 1$ $\texttt{ready}$ massages with $\texttt{share-holder}$ which ensures existence of another honest P_l who sent $\texttt{ready}$ massage with $\texttt{share-holder}$. Now again by recursive argument, this case will boil down to case (a). Hence, we prove the lemma. □

The commitment property is the most interesting to prove. It follows from Lemma 4.3.

Lemma 4.3. *If some honest party P_i has agreed on $\texttt{Com}$, 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 $\texttt{Com}$ 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 $\texttt{Com}$, then she has received $2t+1$ $\texttt{ready}$ messages with $\texttt{Com}$ such that at least $t+1$ of them contain $\texttt{share-holder}$. From the previous proof, eventually $t+1$ honest parties (possibly including P_i) will eventually send out $\texttt{ready}$ with $\texttt{share-holder}$. So there will be a set of at least $t+1$ honest parties who send out $\texttt{ready}$ with $\texttt{share-holder}$. 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 $\texttt{Com}$ and by previous lemma they eventually agree on the same. Also they send out $\texttt{echo}$ well before sending out $\texttt{ready}$. 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 $\texttt{Com}$. 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 $\texttt{Com}$ 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 have broken binding property of underlying commitment scheme, as he knows how to open $\texttt{Com}_{ij}$ in two different ways. □

Using the above lemmas, we can prove the properties of AsynchVSS. We do that in Appendix B.5.

4.3 VSS with the Strong Commitment Property for $n \geq 3t+1$

The AsynchVSS protocol in Figure 4 does not satisfy the strong commitment property. Here, we design a protocol that satisfies the strong commitment property. We ensure that every honest party receives her proper share by making the dealer to execute $n+1$ parallel inter-related instances of AsynchVSS. We now briefly discuss our idea and defer the protocol in Appendix B.

Dealer D would like to share his secret s using a symmetric bivariate polynomial $F(x,y)$ such that $F(0,0) = s$. Now apart from $F(x,y)$, D also selects n random symmetric bivariate polynomials $F^1(x,y), \ldots, F^n(x,y)$ satisfying $F(x,i) = F^i(x,0)$. Now, D runs $n+1$ instance of AsynchVSS for $F(x,y), F^1(x,y), \ldots, F^n(x,y)$ respectively. Let $\texttt{Com}, \texttt{Com}^1, \ldots, \texttt{Com}^n$ are the commitments for these $n+1$ instances. We can combine the send, echo and ready messages for these instances to keep the message complexity same as that of VSS in [5]. Towards the end of the sharing phase of these $n+1$ AsynchVSS instances, all (honest) parties agree on $\texttt{Com}, \texttt{Com}^1, \ldots, \texttt{Com}^n$ and the fact that there are at least $t+1$ honest parties who received $F(i,y)$ and $F^k(x,i)$ for all $k \in [1,n]$ consistent with the commitments and $F^k(0,i)$'s for all k defines $F(i,y)$. This should hold because of the way D selected those polynomials. Now these $t+1$ honest parties can enable every P_k to reconstruct $F^k(x,y)$ and thereby $F(x,k) = F^k(x,0)$. This is done by running the reconstruction phase of AsynchVSS for $k \in [1,n]$, but sending shares to P_k only. Note that P_k can verify the validity of $F^k(x,i)$ sent to him with respect to his copy of $\texttt{Com}^k$, as agreement on all $\texttt{Com}, \texttt{Com}^1, \ldots, \texttt{Com}^n$ are done beforehand.

5 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. However, other VSS applications such as proactive share renewal and share recovery schemes [24] and distributed key generation [14, 34] 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., [10]) 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 protocols based on the definitional properties of commitment schemes are expensive (by a factor of n) in terms of communication complexity in comparison to the respective protocols employing homomorphic commitments. It is also worthwhile to study whether this gap in communication complexity is inevitable.

Acknowledgements. We thank Jonathan Katz for his coments and suggestions on an earlier draft of the paper. We also thank Ian Goldberg and and Mehrdad Nojoumian for interesting initial discussions.

References

[1] M. Abe and S. Fehr. Adaptively Secure Feldman VSS and Applications to Universally-Composable Threshold Cryptography. In *Advances in Cryptology—CRYPTO'04*, pages 317–334, 2004.

[2] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Ext. Abstract). In *ACM STOC'88*, pages 1–10, 1988.

[3] G. R. Blakley. Safeguarding Cryptographic Keys. In *the National Computer Conference*, pages 313–317, 1979.

[4] G. Bracha. An Asynchronous [(n-1)/3]-Resilient Consensus Protocol. In *ACM PODC'84*, pages 154–162, 1984.

[5] C. Cachin, K. Kursawe, A.Lysyanskaya, and R. Strobl. Asynchronous Verifiable Secret Sharing and Proactive Cryptosystems. In *ACM CCS'02*, pages 88–97, 2002.

[6] R. Canetti. *Studies in Secure Multiparty Computation and Applications.* PhD thesis, The Weizmann Institute of Science, 1996.

[7] R. Canetti and T. Rabin. Fast Asynchronous Byzantine Agreement with Optimal Resilience. In *ACM STOC'93*, pages 42–51, 1993.

[8] D. Chaum. Demonstrating That a Public Predicate Can Be Satisfied Without Revealing Any Information About How. In *Advances in Cryptology—CRYPTO'86*, pages 195–199, 1986.

[9] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults. In *IEEE FOCS'85*, pages 383–395, 1985.

[10] P. D'Arco and D. R. Stinson. On Unconditionally Secure Robust Distributed Key Distribution Centers. In *Advances in Cryptology—ASIACRYPT'02*, pages 346–363, 2002.

[11] P. Feldman. A Practical Scheme for Non-interactive Verifiable Secret Sharing. In *IEEE FOCS'87*, pages 427–437, 1987.

[12] M. Fitzi, J. A. Garay, S. Gollakota, C. Pandu Rangan, and K. Srinathan. Round-optimal and efficient verifiable secret sharing. In *TCC'06*, pages 329–342, 2006.

[13] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. The round complexity of verifiable secret sharing and secure multicast. In *ACM STOC'01*, pages 580–589, 2001.

[14] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. *J. of Cryptology*, 20(1):51–83, 2007.

[15] R. Gennaro, M. O. Rabin, and T. Rabin. Simplified vss and fact-track multiparty computations with applications to threshold cryptography. In *ACM PODC'98*, pages 101–111, 1998.

[16] O. Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. *J. Cryptology*, 9(3):167–190, 1996.

[17] O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, STOC '87, pages 218–229, 1987.

[18] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity for all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

[19] I. Haitner, O. Horvitz, J. Katz, C.-Y. Koo, R. Morselli, and R. Shaltiel. Reducing Complexity Assumptions for Statistically-Hiding Commitment. In *Advances in Cryptology—EUROCRYPT'05*, pages 58–77, 2005.

[20] I. Haitner and O. Reingold. A new interactive hashing theorem. In *IEEE Conference on Computational Complexity (CCC)*, pages 319–332, 2007.

[21] I. Haitner and O. Reingold. Statistically-hiding commitment from any one-way function. In *ACM STOC'07*, pages 1–10, 2007.

[22] S. Halevi and S. Micali. Practical and provably-secure commitment schemes from collision-free hashing. In *Advances in Cryptology—CRYPTO'96*, pages 201–215, 1996.

[23] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[24] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In *Advances in Cryptology—CRYPTO'95*, pages 339–352, 1995.

[25] J. Katz, C. Koo, and R. Kumaresan. Improving the round complexity of vss in point-to-point networks. In *ICALP(2)'08*, pages 499–510, 2008.

[26] R. Kumaresan, A. Patra, and C. Pandu Rangan. The round complexity of verifiable secret sharing: The statistical case. In *Advances in Cryptology—ASIACRYPT'10*, pages 431–447, 2010.

[27] M. Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[28] M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect Zero-Knowledge Arguments for P Using Any One-Way Permutation. *J. Cryptology*, 11(2):87–108, 1998.

[29] A. Patra, A. Choudhary, T. Rabin, and C. Pandu Rangan. The round complexity of verifiable secret sharing revisited. In *Advances in Cryptology—CRYPTO'09*, pages 487–504, 2009.

[30] A. Patra, A. Choudhary, and C. Pandu Rangan. Efficient Asynchronous Byzantine Agreement with Optimal Resilience. In *ACM PODC'09*, pages 92–101, 2009.

[31] A. Patra, A. Choudhary, and C. Pandu Rangan. Efficient Statistical Asynchronous Verifiable Secret Sharing with Optimal Resilience. In *ICITS'09*, pages 74–92, 2009.

[32] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27:228–234, 1980.

[33] T. P. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *Advances in Cryptology—Eurocrypt'91*, pages 522–526. Springer-Verlag, 1991.

[34] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Advances in Cryptology—CRYPTO'91*, pages 129–140, 1991.

[35] T. Rabin. Robust Sharing of Secrets when the Dealer is Honest or Cheating. *Journal of ACM*, 41(6):1089–1109, 1994.

[36] T. Rabin and M. Ben-Or. Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (Extended Abstract). In *ACM STOC'89*, pages 73–85. ACM Press, 1989.

[37] D. A. Schultz, B. Liskov, and M. Liskov. MPSS: Mobile Proactive Secret Sharing. *ACM Trans. Inf. Syst. Secur.*, 13(4):34, 2010.

[38] A. Shamir. How to Share a Secret. *Commun. ACM*, 22(11):612–613, 1979.

[39] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[40] L. Zhou, F. B. Schneider, and R. van Renesse. APSS: Proactive Secret Sharing in Asynchronous Systems. *ACM Trans. Inf. Syst. Secur. (TISSec)*, 8(3):259–286, 2005.

A 1-Round WSS for $n \geq 2t + 1$

In the literature, another primitive *weak* verifiable secret sharing (WSS), that is strictly weaker than VSS, is also been used [25, 35, 36]. It is generally defined as a step towards the final VSS protocol. The WSS setting is the same as for VSS and the definition satisfies the secrecy and correctness properties in Section 2; however, the commitment property is relaxed to the following definition:

Weak Commitment. If D is faulty then at the end of the sharing phase there is a value $s^* \in \mathbb{F}_p \cup \perp$ such that at the end of the reconstruction phase, each honest party will output either s^* or $\perp$.

Notice that if the broadcast channel is not used during the reconstruction phase, it is not required that all honest parties output the same value between s^* and $\perp$; some may output s^*, while others may output $\perp$.

Protocol 1-Round-WSS$(D, \mathcal{P}, s)$

Sharing Phase: One Round

1. D selects two random polynomials $f(x)$ and $r(x)$ of degree one, such that $f(0) = s$.
2. For every i, $2 \leq i \leq n$, D sends $f_i = f(i)$ and $r_i = r(i)$ to P_i. D also broadcasts $\mathcal{C}_i = \mathsf{Commit}(f(i), r(i))$ for $2 \leq i \leq n$.

Reconstruction Phase: One Round

1. Each P_i broadcasts f_i and r_i.

Local Computation: For every party P_k,

1. Party $P_i \in \mathcal{P} \setminus \{D\}$ is said to be *confirmed* if $f_i \stackrel{?}{=} \mathsf{Open}(\mathcal{C}_i, f_i, r_i)$.
2. If there are more than t *non-confirmed* parties then output $\perp$. Otherwise, interpolate f_i of all *confirmed* parties to a polynomial $f'(x)$. If $\deg(f') \leq t$ then output $s = f'(0)$ else output $\perp$.

Figure 5: 1-Round WSS for $n \geq 2t + 1$

Here, we present a 1-round WSS protocol with $n \geq 2t + 1$. The idea of our WSS protocol is very similar to 1-round VSS presented in Section 3.4. However, we allow the dealer D to take part in reconstruction phase in our WSS protocol. We present our WSS protocol in Figure 5. We note that our protocol achieves optimal fault tolerance as WSS can be shown to be impossible for honest minority ($n \leq 2t$).

B Proofs

Here, we provide more elaborate analyses and proofs of our theorems in the paper.

B.1 Proof for Theorem 3.1: 2-Round VSS for $n \geq 2t + 1$

We analyze the secrecy, correctness and strong commitments properties of VSS to prove Theorem 3.1.

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 α_{ij}, β_{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 [34] 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 $P_i \in \mathbb{Q}$ 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 α_{ik}, β_{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.

B.2 Impossibility Results for 1-Round VSS Schemes

We show our impossibility results assuming the underlying network is complete; i.e., every party can communicate to everybody else. The same impossibility will definitely hold on the weaker network model that we consider i.e. only the dealer is connected to every other party.

B.2.1 Proof for Theorem 3.2: Impossibility for $t \geq 2$

Without loss of generality, we assume $t = 2$. The proof is by contradiction. Let the set of parties be $\{P_1, \ldots, P_n\}$, and assume there exists a 1-round sharing protocol Π for VSS with D being any party other than P_1 (this can be assumed without loss of generality).

Let us now look at the structure of the sharing phase of Π. Let party P_i start with random coin r_i.[2] In the first round, private messages are exchanged between parties and also parties broadcast messages individually. The private messages and broadcast messages of P_i are function of its random coin r_i. We denote the private message that P_i sends to P_j by r_{ij}, and the broadcast of party P_i by α_i. So given r_i, we assume that P_i's private messages for round one can be deterministically generated. Similarly, we may write

[2] r_i's are actually random variables here. For different executions of Π, they may take different values.

$\alpha_i(r_i)$ to mean that given r_i, the broadcast message α_i can be generated deterministically. At the end of the sharing phase, each party locally outputs his *view* of the sharing phase i.e. all the information (broadcasted as well as private) seen by that party so far. Figure 6 provide a formal definition of view V_i of a party P_i in protocol Π.

The *view* of a party P_i denoted by V_i in protocol Π consists of the random coin r_i of P_i and all the messages (private messages and broadcasts) received by him during the sharing phase of Π.

1. $P_1, \ldots, P_n$ participate in protocol Π with random coins $r_1, \ldots, r_n$, respectively. D has input s (implicitly defined by r_D,where $r_D = r_i$ if P_i is D).
2. Round 1
   (a) Private messages communicated by the parties.
      i. Private messages of P_1: $r_{12}, r_{13}, \ldots, r_{1n}$.
      ii. .....
      iii. Private messages of P_n: $r_{n1}, r_{n2}, \ldots, r_{n(n-1)}$.
   (b) Broadcasts: $\alpha_1(r_1), \ldots, \alpha_n(r_n)$.
3. Local outputs:
   (a) $V_1 = (r_1, r_{21}, \ldots, r_{n1}, \alpha_2, \ldots, \alpha_n)$.
   (b) .....
   (c) $V_n = (r_n, r_{1n}, \ldots, r_{(n-1)n}, \alpha_1, \ldots, \alpha_{n-1})$.

Figure 6: A formal description of the sharing phase of Π

Without loss of generality, we assume that dealer's secret s is implicitly contained in r_D (i.e., the dealer's random coin). So far we have discussed about the structure of the sharing phase of protocol Π.

Now let us fix how a reconstruction phase of Π would look like. According to the definition of VSS protocol, the reconstruction phase can be simulated by a function, say REC, which takes the views of the parties generated at the end of sharing phase. In other words, given the views of the parties at the end of the sharing phase, we can always define a function REC to simulate the actual reconstruction phase (that may require any number of rounds in our context). Let us now define REC formally.

Definition B.1. The reconstruction function REC takes as input the set of views of all the parties that participate in the reconstruction phase of protocol Π and outputs D's committed secret irrespective of whether D is honest or corrupted. Since all the honest parties participate in the reconstruction phase, REC will have at least $n-2$ input views. The corrupted parties may input anything as their view. Let $V_H = \{V_i | P_i \text{ is honest}\}$ and let $V_C = \{V_i | P_i \text{ is corrupted}\}$. Let s be the fixed secret that D is committed to in the sharing phase. Then REC satisfies the following,

- For every possible value of V_C, $\mathsf{REC}(V_H, V_C) = s$ (follows from correctness property when D is honest; follows from commitment property when D is corrupted).

For our purpose, we allow REC to internally simulate the behavior of all the parties in the actual reconstruction phase of Π. That is, REC assumes that all the parties (including those that deviated from the protocol in the sharing phase) act honestly in the reconstruction phase. Of course this assumption does not stop a corrupted party to input junk view to REC. What we mean by the previous statements is that once all the inputs are fed to REC function, REC internally simulates the honest behavior of the parties with the inputs.

In Figure 7, we present a real execution G of Π, where D's secret is s^G. We denote the view of P_i by $V_i(G)$ in execution G. By the property of REC, we have the following claim:

Claim B.2. $\mathsf{REC}(V_1(G), \ldots, V_n(G)) = s^G$.

Let $V_i^\star(G)$ is defined to be same as $V_i(G)$ with r_{Di}^G is replaced by any value $\overline{r_{Di}^G}$. Now we show the following:

Claim B.3. $\mathsf{REC}(V_1(G), \ldots, V_n^\star(G)) = s^G$.

1. $P_1, \ldots, P_n$ participate in execution G with random coins $r_1^G, \ldots, r_n^G$, respectively. D has input s^G (implicitly defined by r_D^G).
2. Round 1
   (a) Private messages communicated by the parties.
      i. Private messages of P_1: $r_{12}^G, r_{13}^G, \ldots, r_{1n}^G$.
      ii. .....
      iii. Private messages of P_n: $r_{n1}^G, r_{n2}^G, \ldots, r_{n(n-1)}^G$.
   (b) Broadcasts: $\alpha_1(r_1^G), \ldots, \alpha_n(r_n^G)$.
3. Local outputs:
   (a) $V_1(G) = (r_1^G, r_{21}^G, \ldots, r_{n1}^G, \alpha_2^G, \ldots, \alpha_n^G)$.
   (b) .....
   (c) $V_n = (r_n^G, r_{1n}^G, \ldots, r_{(n-1)n}^G, \alpha_1^G, \ldots, \alpha_{n-1}^G)$.

Figure 7: A Real Execution G of Π

PROOF: Let in G, the dealer D was honest and P_n was corrupted. At the end of sharing phase, let P_n replaces r_{Dn}^G (that he has received from D) by any value $\overline{r_{Dn}^G}$ in his view $V_n(G)$ and inputs it to REC. By correctness of Π, function REC should output s^G. This proves our claim. □

Claim B.4. $\mathsf{REC}(V_1(G), \ldots, V_{n-1}^\star(G), V_n^\star(G)) = s^G$.

PROOF: Let in G, the dealer D was corrupted and distributed r_{Di}^G for all $i = 1, \ldots, n-1$ and $\overline{r_{Dn}^G}$ (this can be any value) to P_n. Now if every party (including D) behaves properly and inputs correct views then by Claim B.3, s^G will be reconstructed. On the other hand, let P_{n-1} becomes corrupted at the end of sharing phase (apart from D) and replaces $r_{D(n-1)}^G$ (that he has received from D) by any value $\overline{r_{D(n-1)}^G}$ in his view $V_{n-1}(G)$ and inputs it to REC. By commitment property of Π, function REC should still output s^G. This shows that our claim is true. Our assumption that two parties can be corrupted has been used in this claim. □

Like this we can proceed and prove the following claim in the same way as in Claim B.4:

Claim B.5. $\mathsf{REC}(V_1(G), \ldots, V_{n-2}^\star(G), V_{n-1}^\star(G), V_n^\star(G)) = s^G$.

Therefore we can prove a series of claims in this way and finally land up in the following claim.

Claim B.6. $\mathsf{REC}(V_1(G), V_2^\star(G), \ldots, V_{n-1}^\star(G), V_n^\star(G)) = s^G$.

Finally the above Claim clearly shows a violation of the secrecy property of Π because it states that in any execution, where D gives message r_{D1}^G to P_1, will always output the secret s^G at the end of the reconstruction phase. So if D is honest and adversary passively corrupts P_1 in such an execution, he will come to know that the shared secret is s^G, which is a violation of secrecy property. This implies that Π does not exist and our theorem is correct.

Note that Theorem 3.2 does not hold for WSS due to the fact that WSS requires only weak commitment, this prevents the argument that all sequences of messages sent to the parties need to be reconstructed to the same secret.

B.2.2 Proof for Theorem 3.3: Impossibility for $n \leq 3t$

Due to the player partitioning technique, it is enough to show that 1-round sharing VSS is impossible for $n \leq 3$ and $t = 1$. Our proof is by contradiction. Let the set of parties be $\{P_1, P_2, P_3\}$ with $D = P_1$, and assume there exists a 1-round sharing protocol Π for VSS.

The general structure of the sharing phase of Π will be same as the description presented in the proof of Theorem 3.2 restricted to three parties. Also the description for the reconstruction phase of Π will be very much same as presented in Theorem 3.2. In Figure 8, we present a real execution G of Π, where D's secret is s^G. We may denote the view of P_i by $V_i(G)$ in execution G.

By the property of REC, we have the following claim:

1. P_1, P_2, P_3 participate in execution G with random coins r_1^G, r_2^G, r_3^G, respectively. D has input s^G (implicitly defined by r_D^G).
2. Round 1
   (a) Private messages communicated by the parties.
      i. Private messages of P_1: r_{12}^G, r_{13}^G.
      ii. Private messages of P_2: r_{21}^G, r_{23}^G.
      iii. Private messages of P_3: r_{31}^G, r_{32}^G.
   (b) Broadcasts: $\alpha_1(r_1^G), \alpha_1(r_2^G), \alpha_n(r_3^G)$.
3. Local outputs:
   (a) $V_1(G) = (r_1^G, r_{21}^G, r_{31}^G, \alpha_2^G, \alpha_3^G)$.
   (b) $V_2(G) = (r_2^G, r_{12}^G, r_{32}^G, \alpha_1^G, \alpha_3^G)$.
   (c) $V_3(G) = (r_3^G, r_{13}^G, r_{23}^G, \alpha_1^G, \alpha_2^G)$.

Figure 8: A Real Execution G of Π for P_1, P_2 and P_3

Claim B.7. $\mathsf{REC}(V_1(G), V_2(G), V_3(G)) = s^G$.

We now show that $\mathsf{REC}(\star, V_2(G), \star) = s^G$, where $\star$'s can be replaced by any arbitrary view.

Claim B.8. $\mathsf{REC}(\star, V_2(G), \star) = s^G$, where $\star$'s can be replaced by any arbitrary view.

Proof: First we show that $\mathsf{REC}(V_1(G), V_2(G), \star) = s^G$. If it is not the case, then Π does not obey correctness when D is honest and P_3 is corrupted (and inputs any view to REC). Now consider the case when D was corrupted and had sent some arbitrary $\overline{r_{13}^G}$ to honest P_3. So the views of D and P_2 remain the same. The view of P_3 becomes $V_3^\star(G) = (r_3^G, \overline{r_{13}^G}, r_{23}^G, \alpha_1^G, \alpha_2^G)$. Then $\mathsf{REC}(V_1(G), V_2(G), V_3^\star(G)) = s^G$ holds by substituting $\star = V_3^\star(G)$ in $\mathsf{REC}(V_1(G), V_2(G), \star) = s^G$. Now if D inputs $V_1^\star(G)$ (which is arbitrary) to REC, still REC should output s^G by commitment property of Π. So we have $\mathsf{REC}(\star, V_2(G), \star) = s^G$. □

The above claim shows that the view of P_2 is enough to reconstruct the secret. This means that if D was honest and P_2 was corrupted then P_2 can break the secrecy of Π, which is a contradiction to the fact that Π is a VSS scheme.

Similar to Theorem 3.2, Theorem 3.3 does not hold for a WSS scheme as WSS requires only weak commitment. So the argument for the proof of claim $\mathsf{REC}(\star, V_2(G), \star) = s^G$ will fall apart in this case. In fact we can design a 1-round sharing, 1-round reconstruction $(2t+1, t)$ WSS protocol.

B.3 Proof for Theorem 3.6: 1-Round VSS for $t = 1$ and $n \geq 4$

We analyze the secrecy, correctness and commitment properties of VSS to prove Theorem 3.6. We assume any commitment function that provides unconditional (or statistical) hiding and computational binding under some standard hardness assumption (e.g., Pedersen commitment scheme)

Secrecy. We prove secrecy by showing that given the public information and view V_i comprising of (f_i, r_i) at some index i, the adversary has no information about the shared secret s. Formally,

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

This follows from the unconditional hiding property of the underlying commitment function using the standard techniques [34].

Correctness. If D is honest, there is at most one corrupted party in $\mathcal{P} \setminus \{D\}$. As every honest party will always be *confirmed*, there cannot be more than one *non-confirmed* parties. Now if the sole corrupted party is *confirmed*, then his published values f_i and r_i are indeed correct point on $f(x)$ and $r(x)$. If it is not true then the adversary (corrupted party) finds some $f_i^\star$ and $r_i^\star$ such that $f_i^\star = \mathsf{Open}(\mathcal{C}_i, f_i^\star, r_i^\star)$. We can

then devise an algorithm to break the computational binding property of the commitment function using this adversary. Hence given that the commitment function achieves computational binding, all the *confirmed* parties disclose proper f_i and r_i values and therefore the interpolated polynomials are $f(x)$, $r(x)$, and the reconstructed secret is $s = f(0)$.

Strong Commitment. If D is corrupted, then all the remaining parties in $\mathcal{P} \setminus \{D\}$ are honest. We say that a corrupted D has committed $\perp$ if he does either one of the following:

1. if there are more than one party in $\mathcal{P} \setminus \{D\}$ who received f_i and r_i that are inconsistent with corresponding public commitments;
2. if the underlying polynomial as implied by the public commitments is of degree more than 1.

If none of the above is the case, then D's committed secret s is the constant term of degree-1 polynomial that is implied by the public commitments. Now it is easy to see that $\perp$ will be reconstructed when D commits to $\perp$ and likewise, s will be reconstructed when D commits so. We also note that strong commitment holds unconditionally. That is no matter how much computational power the adversary has, he cannot make a value that is not committed to be reconstructed by controlling D. This holds because all the participating parties in reconstruction phase are honest.

B.4 Proofs for the VSS using Homomorphic Commitment

Now we prove the following properties of 2-Round-VSS-Hm.

Secrecy. The secrecy of the scheme mainly follows from the unconditional hiding property of Pedersen commitment function and the property of degree-t polynomial. In the first round of the protocol, the adversary receives t points on $f(x)$ and $r(x)$ polynomials and the public commitments on their coefficients. We claim that the communication in second round does not add any extra information to the adversary's view in the first round. In second round, an honest D broadcasts f_i and r_i only when P_i is corrupted. Otherwise, D broadcasts $\alpha_i = f_i + p_i$ and $\beta_i = r_i + g_i$. Now we note that p_i and g_i are randomly chosen and they remain information theoretically secure even the commitments $\texttt{PCom}_i$ and $\texttt{GCom}_i$ are public (from the perfect hiding property of Pedersen's Commitment function). When p_i and g_i are used to hide f_i and r_i (while revealing α_i and β), f_i and r_i remain as secure as prior to the broadcast of α_i and β_i. This completes our claim that round two communication does not add anything extra to the information obtained by the adversary in the first round of the protocol. Now the perfect secrecy of the secret s follows from the proof of Petersen [33] (see Theorem 4.4 of [33]) when the adversary receives t points on $f(x)$ and $r(x)$ polynomials and the public commitments on their coefficients. In brief the secrecy follows from the fact that both $f(x)$ and $r(x)$ polynomials are of degree t and D's public commitments $\texttt{Com}_i$ will be uniformly distributed given the unconditional hiding property of Pedersen commitment function.

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 $P_i \in \mathbb{Q}$ is considered as *confirmed* only when he broadcasts correct values in the reconstruction phase. Assume that corrupted P_i is considered to be *confirmed* even when he broadcasts f_i' and r_i' where these values are not equal to f_i and r_i (as given by the dealer). By homomorphism of commitment scheme, $\prod_{j=0}^{t}(\texttt{Com}_i)^{i^j}$ is the commitment for both pairs (f_i, r_i) and (f_i', r_i'). This implies corrupted P_i is able to break the computational binding property of the commitment function. Hence given that the commitment function achieves computational binding, all the *confirmed* parties disclose proper f_i and r_i and therefore $f(x)$ and $s = f(0)$ will be reconstructed correctly.

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. If D is not discarded, then the polynomials, say $f(x)$ and $r(x)$, defined by the public commitments of D are degree-t polynomials. We note that an honest party will never be discarded. Now we need to ensure that at the end of sharing phase honest P_i will output f_i and r_i, the i^{th} shares of $f(x)$ and $r(x)$, respectively. We consider two cases:

P_i **is** `happy`**:** P_i verified that $\mathsf{Commit}(f_i, r_i) = \prod_{j=0}^{t}(\mathtt{Com}_i)^{i^j}$. This implies that the pair (f_i, r_i) used by P_i are the correct ones, unless corrupted D had broken the binding property of the commitment function.

P_i **is** `unhappy`**:** We have two cases: (1) D has broadcasted f_i and r_i; (2) D has broadcasted α_i, β_i and P_i computed $f_i = \alpha_i - p_i$, $r_i = \beta_i - g_i$. In both the above cases, f_i and r_i satisfies $\mathsf{Commit}(f_i, r_i) = \prod_{j=0}^{t}(\mathtt{Com}_i)^{i^j}$. So unless corrupted D had broken the binding property of the commitment function, the pairs (f_i, r_i) and $(f(i), r(i))$ are identical.

Now in the reconstruction phase, every honest party will be considered as *confirmed*. However, a corrupted party P_i will be considered as *confirmed* if he broadcasts $f_i = f(i)$ and $r_i = r(i)$ (assuming he does not break binding of commitment function). Now it follows that the parties will reconstruct D's committed secret $s = f(0)$ in the reconstruction phase.

B.5 Asynchronous VSS with strong commitment

We now proceed to prove the properties of protocol AsynchVSS.

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 4.2.

Correctness. Correctness follows from Lemma 4.2 and 4.3. 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 4.3. 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 $F(x, y)$: $f_j(k) = \mathsf{Open}(\mathtt{Com}_{jk}, f_j(k), r_j(k))$ 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.

In Figure 9, we present an asynchronous computational VSS scheme that satisfies the strong commitment properties for $n \geq 3t + 1$ as discussed in Section 4.3

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

Sharing Phase:
Code for D**:**

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

Code for P_i**:**

- On receiving $(\texttt{send}, C, f_i(y))$ and $(\texttt{send}, C^k, f^k_i(x), r^k_i(x))$ for all $k \in [1,n]$ from D, send $(\texttt{echo}, C, C^1, \ldots, C^n)$ to every P_j if
  1. $C, C^1, \ldots, C^n$ are $n \times n$ symmetric matrices and
  2. $f_i(k) \stackrel{?}{=} f^k_i(0)$ for all $k \in [1,n]$ and
  3. $f_i(j) \stackrel{?}{=} \mathsf{Open}(C_{ij}, f_i(j), r_i(j))$ and
  4. $f^k_i(j) \stackrel{?}{=} \mathsf{Open}(C^k_{ij}, f^k_i(j), r^k_i(j))$ for $j,k \in [1,n]$.
- On receiving $(\texttt{echo}, C, C^1, \ldots, C^n)$ from at least $n-t$ parties satisfying that $(C, C^1, \ldots, C^n)$ received from P_j is same as received from D, send $(\texttt{ready}, \texttt{share-holder}, C, C^1, \ldots, C^n)$ to every P_j, if you have already sent out `echo` messages.
- If you have not sent out any `ready` signal before:
  1. on receiving `ready` messages from at least $n-2t$ P_j's satisfying that $(C, C^1, \ldots, C^n)$ received from P_j is same as received from D, send $(\texttt{ready}, \texttt{share-holder}, C, C^1, \ldots, C^n)$ to every P_j, if you have already sent out `echo` messages.
  2. on receiving $(\texttt{ready}, \texttt{share-holder}, C, C^1, \ldots, C^n)$ from at least $n-2t$ P_j's such that all the $C, C^1, \ldots, C^n$ are same but do not match with the copies received from D, update your $C, C^1, \ldots, C^n$ with these new matrices, delete everything else received from D and send $(\texttt{ready}, \star, C^1, \ldots, C^n)$ to every P_j.
- On receiving `ready` signals from at least $(n-t)$ parties such that all of them contain same $(C, C^1, \ldots, C^n)$ as yours and at least $(n-2t)$ `ready` signals contain `share-holder`, agree on $(C, C^1, \ldots, C^n)$ and send $(\texttt{final}, f^j_i(x), r^j_i(x))$ to P_j if you had sent $(\texttt{ready}, \texttt{share-holder}, C, C^1, \ldots, C^n)$.
- Wait for $t+1$ $(\texttt{final}, f^i_j(x), r^i_j(x))$ messages such that $f^i_j(x)$ is degree-t polynomial, $r^i_j(x)$ is degree-$(n-1)$ polynomial and $f^i_j(k) = \mathsf{Open}(C^i_{jk}, f^i_j(k), r^i_j(k))$ for all $k \in [1,n]$, interpolate $F^i(x,y)$ using those $t+1$ $f^i_j(x)$ polynomials, compute $f_i(x) = F^i(x,0)$ and terminate with $s_i = f_i(0)$ as the share of secret.

Reconstruction Phase:
Code for P_i**:**

1. Each P_i sends s_i to every P_j.
2. On receiving the shares from the parties, apply online error correction of [6] to get back the secret.

Figure 9: Asynchronous VSS with strong commitment for $n \geq 3t+1$