Verifiable Secret Sharing With Honest Majority

A PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

Master of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

by

Aakar Deora

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

JULY 2015

DEDICATED TO

My Family

for continuous support and encouragement

Signature of the Author: ....

Aakar Deora
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore

Signature of the Thesis Supervisor: ....

Arpita Patra
Assistant Professor
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore

Acknowledgments

I am deeply grateful to Prof. Arpita Patra for her excellent guidance, enthusiasm and supervision. She has always been a source of inspiration for me. I have been extremely lucky to work with her.

Also, I am thankful to Ajith S for all those long discussions and suggestions. It had been a great experience to work with him.

Thanks to the Department of Computer Science and Automation, along with all the faculty members, students, and non-teaching staff, who made my stay in the department a charm. Also I thank my CSA friends who made my stay at IISc pleasant, and for all the fun we had together.

Finally, I am indebted with gratitude to my parents and sister for their love and inspiration that no amount of thanks can suffice. This project would not have been possible without their constant support and motivation.

Abstract

Verifiable Secret Sharing (VSS) is a fundamental cryptographic primitive, which is used as a basic building block in almost every protocol for secure computation. It also serves as an important building block for Byzantine Agreement (BA) protocols. Informally, VSS allows a dealer to share a secret among several players which may later be uniquely reconstructed, even if some of the players are malicious (possibly including the dealer). Any VSS scheme consists of two phases: the sharing phase and the reconstruction phase and is implemented by a pair of protocols (Share, Rec). Here Share is the protocol for the sharing phase, while Rec is the protocol for the reconstruction phase. Protocol Share allows a special player called the dealer (denoted as D), to share a secret s among the n players in a way that later allows for a unique reconstruction of s by every player using the protocol Rec. Moreover, if D is honest, then the secrecy of s is preserved till the end of Share.

We focus on a standard setting of *statistical information-theoretic security* where VSS protocol exists if and only if $t < n/2$. In this model the standard assumption is that the players are connected to each other via point-to-point secure and authenticated channels. Furthermore, they have an access to a shared broadcast channel.

The round complexity of VSS protocols is defined as the number of rounds required to complete the sharing phase of VSS. Broadcast round complexity is another important complexity measure in VSS, which is estimated as the number of rounds in the protocol where a physical broadcast was required. Since a physical broadcast channel is an expensive resource, it is desirable to minimize the broadcast round complexity of a protocol. In this thesis, we have proposed a new VSS protocol with just two broadcast rounds in the sharing phase, inspired from the VSS protocol of Patra et al. [10] which has three broadcast rounds in the sharing phase. Our protocol has a overall round complexity 10 in the sharing phase. The only known protocol with two broadcast rounds is given by Garay et al. [6]. The overall round complexity of Garay et al. is 20. Our protocol is an improvement over the existing protocols in terms of either broadcast round complexity or overall round complexity.

We also focus on *information checking protocol* (ICP) which is used as a building block for VSS schemes. ICP is traditionally defined as an interactive protocol between three players namely, the dealer D, the intermediary INT, and the verifier $\mathcal{V}$ [10]. Initially, D hands over the secret $s \in \mathbb{F}$ to INT and passes some verification information to the $\mathcal{V}$. $\mathcal{V}$ learns nothing about s from the verification information. Later, INT passes the secret to $\mathcal{V}$ along with a proof that s is indeed the actual secret. Then, $\mathcal{V}$ confirms the validity of the secret using the verification information shared by D. In this work, we have modified the ICP given by Patra et al. [10] and reduced its broadcast round complexity from four to three. We use this modified ICP to come up with our new statistical VSS protocol with broadcast round complexity better than known protocols.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Secret sharing is one of the most important primitives used in secure multi-party computation protocols. In secret sharing, a designated player, called dealer, wants to share a secret s among n players in such a way that no set of t players are able to reconstruct the secret but any set of $t+1$ players should be able to reconstruct it by combining their respective shares.

Verifiable Secret Sharing (VSS) is an extension of secret sharing which is used in the presence of active corruption. Here a central adversary may corrupt upto t players (possibly including the dealer) in an arbitrary manner. The requirement of VSS is that no t players can reconstruct the secret in any way whereas the n players can reconstruct the secret successfully even if the malicious t players deliver incorrect information. Moreover, if the dealer is honest, then no information about the secret should be revealed to any of the t malicious players by the end of the *sharing phase*. Nevertheless, even a malicious dealer, by the end of the sharing phase, is irrevocably committed to *some* value which will be reconstructed by the honest players in the *reconstruction phase*. Furthermore, if the dealer is honest then this committed value should be same as the dealer's initial input.

1.1 Motivation

Secret sharing is required quite often in secure multi-party computation. Different values are required to be shared in a secret fashion to evaluate a multi-function, of many inputs, without revealing a party's own inputs to other players. And since few players can be malicious, hence the protocol for secret sharing should be verifiable.

Information Checking Protocol (ICP) is also a basic building block for various VSS protocols. In many cases, the complexity of VSS heavily depends upon the complexity of ICP. Hence we study ICP in detail too.

1.2 Contributions

Our focus is on *information theoretic* VSS, where the adversary possesses unbounded computational power. We also work on the case of statistical security where the security requirements hold *statistically* but can be violated with negligible probability. We assume a broadcast channel, which allows each player to send a message to all the players and ensures that the received message is identical. Using the broadcast channels along with private point to point channels is a standard model for secure multi-party computation protocols in information-theoretic setting. Without broadcast, VSS cannot be achieved in constant number of rounds. We try to reduce the broadcast round complexity of VSS, given in [10], by using gradecast technique in the ICP protocol given in the same paper. Gradecast allows us to simulate broadcast with weaker consistency guarantees. It enables us to reduce communication complexity if multiple broadcast rounds can be replaced by gradecast without affecting any of the characteristic property of the protocol. We will show that it is possible to do so, hence increasing the efficiency of the protocol.

1.3 Organization

The remainder of this thesis is organized as follows: Chapter 2 provides the background details that includes the model, definitions and other building blocks of VSS. Protocols like Weak broadcast and Gradecast are discussed in this chapter. In chapter 3, few previous works regarding VSS are reviewed. Chapter 4 discusses the contributions in this work in detail. The ICP protocol is studied in detail and thereafter modified in section 4.1. Once the ICP is established according to our model, we use it to come up with a improved VSS protocol in section 4.2.

Chapter 2

Background

In this chapter, we present necessary details about the model, definitions and tools related to Verifiable Secret Sharing.

2.1 VSS model

We consider the standard model of communication where all the players have an access to pairwise private channel and a common broadcast channel. The network is synchronous one consisting of n players $P_1, P_2, ..., P_n$. There exists a centralized adversary $\mathcal{A}$ with unbounded computational power. The adversary can corrupt upto t players, possibly including the dealer. The corrupted players may deviate from the protocol in an arbitrary way. $\mathcal{A}$ is also *rushing* in nature, which means that it first receives the messages of the honest players before deciding on the messages of corrupted players in a particular round.

Let $\mathbb{F}$ denotes a finite field and set $\kappa = \log|\mathbb{F}|$. The dealer's secret is supposed to lie in $\mathbb{F}$ and κ is the security parameter. In the statistical VSS, we allow an error probability of at most $\varepsilon = 2^{-\Theta(\kappa)}$.

2.2 Verifiable Secret Sharing

A two phase protocol for a set of n players $\mathcal{P} = P_1, P_2, ..., P_n$, where a designated dealer $D \in \mathcal{P}$ holds the initial input secret $s \in \mathbb{F}$, is a (1 - ε)-*statistical VSS protocol tolerating t corrupted players* if the following conditions hold for adversary $\mathcal{A}$ controlling upto t players:

Privacy: If the dealer D is honest, then the joint view of the corrupted players should be independent of the input secret value s at the end of the first phase (*sharing phase*).

Correctness/Commitment: Each honest player P_i outputs a value s_i at the end of the second phase (*reconstruction phase*). The following should hold except with probability at most ε:

- At the end of the sharing phase, there exists a value $s' \in \mathbb{F}$ which is defined by the joint view of the honest players. All the honest players output s' at the end of reconstruction phase. If D is honest, then $s' = s$.

2.3 Information Checking Protocol

An *information checking protocol (ICP)* was first introduced by Tal Rabin and Michael Ben-Or [13]. It traditionally involves three players, namely, the dealer D, an intermediary INT, and a verifier $\mathcal{V}$. The dealer initially holds the input secret value $s \in \mathbb{F}$ which he passes to INT. D also passes some verification information to $\mathcal{V}$. This verification information does not reveal anything about s. Later, INT passes s to $\mathcal{V}$ along with a "proof" that s is indeed the secret value shared by D to INT.

2.3.1 Multi-Verifier Information Checking Protocol

The traditional ICP protocol involve only a single verifier. Patra et al. [12, 11] gave the protocol involving multiple verifiers, i.e., all the players in the network can act as a verifier. They further gave a simplified version of ICP in [10]. Their version of the protocol consists of three sub protocols (Distr, AuthVal, RevealVal).

- Distr (D, INT, s): It is executed by D using some input value s. The algorithm generates some *authentic information*, which includes s, to give to INT. It also generates some *verification information* and gives it to each of the verifiers.

- AuthVal (D, INT, s): It is executed by INT after the Distr phase. The information held by INT is called D's *IC-signature* and is denoted by *ICSIG(D,*INT*,s)*.

- RevealVal (D, INT, s): In this phase, INT broadcasts *ICSIG(D,*INT*,s)*. Based on the values received in the previous two phases, *ICSIG(D,*INT*,s)* is either accepted or rejected by all the honest verifiers, with high probability.

The *ICP* protocol is required to satisfy the following properties:

Correctness:

- If D and INT are honest, then each honest verifier accepts *ICSIG(D,* INT*,s)* during RevealVal.
- If INT is honest, then he holds an *ICSIG(D,*INT*,s)* at the end of AuthVal which will be accepted by each of the honest verifiers with probability at least $1 - 2^{-\Theta(\kappa)}$.
- If D is honest, then *ICSIG(D,*INT*,s)* broadcasted as $s' \neq s$ by a corrupt INT should be rejected by each honest verifier with probability at least $1 - 2^{-\Theta(\kappa)}$, during RevealVal.

Secrecy: If both D and INT are honest, then the adversary should not learn any information about s till the end of AuthVal.

2.4 Gradecast

Graded broadcast or Gradecast was first given by Feldman and Micali [3]. Gradecast allows us to distribute a value to all the players but with weaker consistency guarantees than the standard broadcast. In the latter case, each player outputs the same value, but in gradecast, each player has to output a binary grade $g_i \in \{0, 1\}$ along with the value v_i.

A protocol achieves gradecast if it allows the dealer $D \in \mathcal{P}$ to distribute a value $v \in \mathbb{F}$ among the players $\mathcal{P}$ with every player P_i outputting a value $v_i \in \mathbb{F}$ along with a grade $g_i \in \{0, 1\}$ such that the following conditions hold:

Validity: If the dealer D is honest, then each honest player $P_i \in \mathcal{P}$ outputs $(v_i, g_i) = (v, 1)$.

Graded Consistency: If an honest player $P_i \in \mathcal{P}$ outputs (v_i, g_i) with $g_i = 1$, then every honest player $P_j \in \mathcal{P}$ outputs (v_j, g_j) with $v_j = v_i$.

Gradecast is achievable by private point-to-point channels in case of $t < n/3$ [4]. But for $t < n/2$, a communication model consisting of *2-cast* channels is required to achieve it. A *2-cast* channel allows a player to broadcast a value to two other players in the network. A construction is given by Hirt and Raykov [8, 6] to prepare a setup which allows to simulate 2-cast channels. A gradecast protocol using this setup is given by Garay et al. [6].

2.4.1 Gradecast Setup

A setup can be prepared allowing to simulate 2-cast channels over point-to-point channels [8, 6]. In order to do that, we need to implement protocols Setup_3 and $\mathsf{Broadcast}_3$ given by [8]. The setup protocol Setup_3 consists of three rounds, where in the first two rounds point-to-point channels are used, and a physical broadcast is required in the third round. The protocol $\mathsf{Broadcast}_3$ simulates the 2-cast channel from the prepared setup using point-to-point channels. $\mathsf{Broadcast}_3$ also consists of three rounds but does not use any physical broadcast.

Since gradecast can be achieved from 2-cast channels when $t < n/2$, we can consider the setup prepared for 2-cast channels as the setup for gradecast channels. Let SetupGradecast be the protocol used to generate the setup of 2-cast channels within the network, i.e. it executes the Setup_3 protocol given by [8].

Let the protocol Gradecast (Figure 2.2) be defined as the gradecast protocol given by Garay et al. [6]. Protocol Gradecast is a 6-round protocol which achieves gradecast from a setup and point-to-point channels tolerating $t < n/2$ corrupted players.

2.4.2 Weak broadcast

Weak broadcast (also known as Crusader agreement [2]) is another weak form of broadcast, where the recipients either can output the value sent by the broadcaster or a special symbol $\{\perp\}$. $\{\perp\}$ produced as the output indicates that the broadcaster is malicious. Weak broadcast can be achieved over point-to-point channels only.

A protocol achieves weak broadcast if it allows the dealer D to broadcast a value $v \in \mathbb{F}$ among the players $\mathcal{P}$ with every player P_i outputting a value $v_i \in \mathbb{F} \cup \{\perp\}$ such that the following conditions hold:

Validity: If the dealer D is honest, then each honest player $P_i \in \mathcal{P}$ outputs $v_i = v$.

Weak Consistency: If an honest player $P_i \in \mathcal{P}$ outputs $v_i \neq \perp$, then every honest player $P_j \in \mathcal{P}$ outputs either $v_j = v_i$ or $v_j = \perp$.

Please refer Figure 2.1 and Figure 2.2 for the protocols of weak broadcast and gradecast respectively given by Garay et al. [6].

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

Round 1-3: Dealer D 2-casts v to every pair of players in $\mathcal{P} \setminus \{D\}$. 2-cast is achieved by executing $\mathsf{Broadcast}_3$ over point-to-point channels only.

$\forall P_i \in \mathcal{P} \setminus \{D\}$: If all the values received in the 2-cast are same (equal to some $u \in \mathbb{F}$), then output $v_i = u$ else output $v_i = \perp$.
Dealer D outputs v.

Figure 2.1: Weak Broadcast

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

Round 1-3: Dealer D weak broadcasts v. Let the output of each player P_i be w_i.
Round 4-6: $\forall P_i \in \mathcal{P}$ weak broadcasts w_i. Let the output of each player P_j be w_{ij}.

$\forall P_i \in \mathcal{P}$: $\forall u \in \mathbb{F}$ let $T_i^u = \{P_j \in \mathcal{P} | w_{ji} = u\}$. Let v_i be u with maximal $|T_i^u|$ (break ties arbitrarily); if $|T_i^{v_i}| > n/2$ then $g_i = 1$, otherwise $g_i = 0$. Output (v_i, g_i).

Figure 2.2: Gradecast for $t < n/2$

Chapter 3

Literature Review

The concept of Verifiable Secret Sharing was introduced by Chor et al. [1]. Since then a number of researchers have come up with various VSS protocols for different communication models and under different assumptions. The protocols have been designed for many different models such as synchronous and asynchronous network, information-theoretic and computational adversary, perfect and statistical secret sharing etc. Different protocols have also been designed keeping number of players controlled by the adversary in mind. Though we have focused our research on $t < n/2$ setting where n is the total number of players in the network and t is the total number of malicious players. But here below we review few VSS protocols for $t < n/2$ as well as $t < n/3$ settings.

3.1 VSS Protocol by Gennaro et al. [7]

This paper was published in year 2000. The paper focused on the standard setting of perfect information-theoretic security, where all the players have access to secure point-to-point channels and a common broadcast medium. They gave a 3-round VSS protocol for $t < n/3$ case but the protocol was realized by exponential communication complexity.

In this protocol, the dealer shares the secret using replication-based secret-sharing [9] technique. It first creates the enumerations, $S_1, ..., S_K$ for all $\binom{n}{t}$ subsets of t players. Then it divides the secret as $s = \sum_{k=1}^{K} s_k$ where s_k are chosen at random subject to the summation condition only. Then it gives each share s_k to all the players who are not in S_k. In the reconstruction phase, each player reveals the $\binom{n-1}{t}$ shares in its own possession. The share s_k is decided as the value that is revealed most often by each player. The protocol is given in Figure 3.1 [7].

The secrecy property is achieved by the fact that all the players broadcast their shares

$\frac{n}{3}$**-EXP-VSS**

Sharing Phase

1. Let $S_1, ...S_K$ be an enumeration of all $K = \binom{n}{t}$ subsets of t players.

   $\mathcal{D}$ chooses K random values $s_1, .., s_K \in \mathbb{F}$ under the restriction that the secret s equals $\sum_{k=1}^{K} s_k$. Then, $\mathcal{D}$ sends to player P_i the values s_k for all k such that $P_i \notin S_k$.

   Simultaneously, each player P_i sends to each player P_j, a random pad $r_{ij}^{(k)} \in \mathbb{F}$ for each subset S_k such that $P_i, P_j \notin S_k$.

2. For each $i, j, i < j$: and each index k such that $P_i, P_j \notin S_k$.
   - P_i broadcasts $a_{ij}^{(k)} = s_k + r_{ij}^{(k)}$;
   - P_j broadcasts $a_{ji}^{(k)} = s_k + r_{ji}^{(k)}$.

3. For each index k for which there exists a pair i, j such that $a_{ij}^{(k)} \neq a_{ji}^{(k)}$ the dealer broadcasts the value s_k which is now taken by all the players $\notin S_k$ as the proper share.

Reconstruction phase

1. For each index k each player $P_i \notin S_k$ provides the share s_k it owns (either the one received in Step 1 or the one broadcasted by the dealer in Step 3). Take the value that appears most often as the proper share s_k. Set **Rec** $= \sum_{k=1}^{K} s_k$.

Figure 3.1: 3-Round VSS Protocol for $n > 3t$ with exponential communication

padded with random pads selected in Round 1. And there are more than one shares which are not received by any malicious player if the dealer is honest. The properties of correctness and consistency are achieved as follows: If the dealer is honest and one share is missed by any subset of t players, then they cannot prevent its reconstruction as it is given to $2t + 1$ players. In this case, even an incorrect value cannot be reconstructed as the correct share will appear with a majority in every case.

When the dealer is malicious, he can distribute different values of a share to honest players. But every pair of players compare their shares with each other and the dealer is forced to broadcast that share if an inconsistency is found. Since the secret is the summation of these shares, the dealer is now committed to the new secret if he reveals a new value of the share at the end of the sharing phase.

This protocol certainly was the first protocol to have just three rounds in the sharing phase, but the number of enumerations created by the dealer makes it inefficient. Later on, Fitzi et al. [5] came up with an efficient 3-round VSS protocol.

3.2 VSS Protocol by Fitzi et al. [5]

Fitzi et al. [5] gave an efficient 3-round VSS protocol in 2006. They focused on the same standard model of communication as in Gennaro et al. [7] and solved the then open problem of 3-round efficient VSS protocol for $t < n/3$.

In order to design 3-round VSS, they designed a 3-round Weak Verifiable Secret Sharing (WSS) protocol as well. In WSS, a property of weak commitment s desired which is as follows:

Weak Commitment: After the sharing phase, there is a unique $s^* \in \mathbb{F}$ such that either s^* or the default value $\perp \notin \mathbb{F}$ will be reconstructed in the reconstruction phase regardless of the views presented by the malicious players.

In WSS, the dealer chooses a bivariate polynomial $F(x, y)$ such that the secret is $s = F(0, 0)$. Each player P_i gets two polynomials $F(x, i)$ and $F(i, y)$. Then every pair of players compares their shares by binding them with a random pad and then broadcasting them. In the reconstruction phase, they use a concept of *consistency graph* to construct a CORE set of honest players. But it is possible that the cardinality of CORE turns out to be less than $n - t$ in which case $\perp$ is reconstructed. WSS protocol for $t < n/3$ appears in Figure 3.2 [5].

The VSS protocol proposed by them uses the same technique in the sharing phase. WSS protocol is run in parallel step by step. In the reconstruction phase, the random pad of each player is revealed and the secret is reconstructed. But in order to guarantee the consistency of the random pad, each P_i shares a random field element by WSS, and chooses his pads as points on the respective polynomial. The players whose WSS protocol reconstruct $\perp$ get discarded and the remaining players are put in CORE_{sh} set. The pads of players in CORE_{sh} are taken into account to reconstruct the dealer's secret. The full description of VSS protocol appears in Figures 3.3 and 3.4 [5]. Superscript "W" is used to denote the quantities corresponding to the $(\frac{n}{3})$-**WSS** protocols that are run in order to WSS the players' random pads.

Although their VSS is efficient and achieves all the required properties, but it may not be suitable for general multi-party computation. In a case where multiple VSS are invoked simul-

$(\frac{n}{3})$-WSS

Sharing Phase

1.
   - D chooses a random bivariate polynomial $F \in \mathbb{F}[x, y]$ of degree at most t in each variable, satisfying $F(0, 0) = s$. D sends to each player P_i the (univariate) polynomials $f_i(x) = F(x, i)$ and $g_i(y) = F(i, y)$.
   - Player P_i sends to each player P_j an independent random "pad" r_{ij} picked uniformly from $\mathbb{F}$.
2. Player P_i broadcasts:
   - $a_{ij} = f_i(j) + r_{ij}$ (r_{ij} is the pad P_i sent to P_j)
   - $b_{ij} = g_i(j) + r_{ji}$ (r_{ji} is the pad P_i received from P_j)
3. For each pair $a_{ij} \neq b_{ji}$, the following happens:
   - P_i broadcasts $\alpha_{ij} = f_i(j)$
   - P_j broadcasts $\beta_{ji} = g_j(i)$
   - D broadcasts $\gamma_{ij} = F(j, i)$

A player is said to be *unhappy* if the value which he broadcast does not match the dealer's value. If there are more than t unhappy players, disqualify the dealer and stop.

Reconstruction Phase

1. Every happy player P_i broadcasts his polynomials $f_i(x) = F(x, i)$ and $g_i(y) = F(i, y)$.

Local Computation

Each player P_i now constructs a *consistency graph* G over the set of happy players such that there exists an edge between P_j and P_k in G if and only if $f_j(k) = g_k(j)$ and $g_j(k) = f_k(j)$. Since these polynomials are broadcast, every player P_i constructs the same graph G.

Now each player P_i constructs a set $CORE$ of players as follows. Initially, all the players in G whose node degree is at least $n-t$ are inserted into the set. Next, players in $CORE$ consistent with less than $n - t$ other players in $CORE$ are removed. This process continues until no more players can be removed from the set. If the resulting $CORE$ set contains less than $n - t$ elements then P_i outputs $\perp$ otherwise, P_i reconstructs the polynomial $F^*(x, y)$ defined by any $t + 1$ players in $CORE$, and the secret $s^* = F^*(0, 0)$ is reconstructed.

Figure 3.2: Round-Optimal WSS for $n > 3t$

$(\frac{n}{3})$-VSS

Sharing Phase

1.
   - Dealer D chooses a random bivariate polynomial $F \in \mathbb{F}[x, y]$ of degree at most t in each variable satisfying $F(0,0) = s$. D sends to P_i the polynomials $f_i(x) = F(x,i)$ and $g_i(y) = F(i,y)$.
   - Player P_i, $i = 1, ..., n$, selects a random value r_i and starts an instance of **$(\frac{n}{3})$-WSS** acting as a dealer in order to share r_i by means of bivariate polynomial $F_i^W(x, y)$ ($F_i^W(0,0) = r_i$). We call this instance **$(\frac{n}{3})$-WSS**$_i$. Round 1 of **$(\frac{n}{3})$-WSS**$_i$ is run.

2. Player P_i broadcasts the following:
   - $a_{ij} = f_i(j) + F_i^W(0, j)$
   - $b_{ij} = g_i(j) + F_j^W(0, i)$

   Concurrently, round 2 of **$(\frac{n}{3})$-WSS**$_i$, $i = 1, ..., n$, also takes place.

3. For each pair $a_{ij} \neq b_{ji}$ the following happens:
   - P_i broadcasts $\alpha_{ij} = f_i(j)$
   - P_j broadcasts $\beta_{ji} = g_j(i)$
   - D broadcasts $\gamma_{ij} = F(j, i)$

   Concurrently, round 3 of **$(\frac{n}{3})$-WSS**$_i$, $i = 1, ..., n$, also takes place.

   A player is said to be *unhappy* if the value that he broadcast does not match the dealer's value. If there are more than t unhappy players, disqualify D and stop.

Local Computation

- Let $\mathcal{H}$ denote the set of happy players. Remove from $\mathcal{H}$ each player P_i who gets disqualified as the dealer in protocol instance **$(\frac{n}{3})$-WSS**$_i$. Now, if $|\mathcal{H}| < n - t$ then disqualify D and stop.
- For the remaining players, let $\mathcal{H}_i^W$ denote the set of happy players in instance **$(\frac{n}{3})$-WSS**$_i$. For each player $P_i \in \mathcal{H}$, check that there exist at least $n - t$ players in $\mathcal{H}$ who are also in $\mathcal{H}_i^W$; if not, remove P_i from $\mathcal{H}$. Let us call this final set $\mathsf{CORE}_{sh} := \mathcal{H}$. If $|\mathsf{CORE}_{sh}| < n - t$ then disqualify D and stop.

Figure 3.3: Round-Optimal VSS for $n > 3t$ (Sharing Phase)

$(\frac{n}{3})$-VSS

Reconstruction Phase

1. For each $P_i \in \mathsf{CORE}_{sh}$, run the reconstruction phase of $(\frac{n}{3})$-$\mathbf{WSS}_i$, concurrently.

Local Computation: Now each player P_i constructs a set CORE_{Rec} as follows. Initially, $\mathsf{CORE}_{Rec} := \mathsf{CORE}_{sh}$.

- Remove from CORE_{Rec} every player P_i such that the outcome of $(\frac{n}{3})$-$\mathbf{WSS}_i$ equals $\perp$.
- For every $P_i \in \mathsf{CORE}_{Rec}$, use the values a_{ij} he broadcast in round two of the sharing phase to compute $f_i(j) = a_{ij} - F_i^W(0, j)$, $1 \leqslant j \leqslant n$.
- Interpolate these points. Check that the resulting polynomial $f_i(x)$ is a polynomial of degree at most t. If not, remove P_i from CORE_{Rec}.
- Reconstruct the secret by taking any $t+1$ polynomials $f_i(x)$, $P_i \in \mathsf{CORE}_{Rec}$, to obtain $F^*(x, y)$, and compute $s^* = F^*(0, 0)$.

Figure 3.4: Round-Optimal VSS for $n > 3t$ (Reconstruction Phase)

taneously, we may end up with different CORE_{sh} sets.

3.3 VSS Protocol by Patra et al. [10]

In 2010, Patra et al. [10] gave a 4-round statistical information-theoretic VSS protocol for $t < n/2$ setting with polynomial complexity. Currently, this protocol consists of least number of overall rounds in the sharing phase for the mentioned model. They used a sub-protocol *Information Checking Protocol (ICP)* which was also proposed in the same paper. Details of ICP can be found in Chapter 2. The description of ICP with multiple verifiers appears in Figure 3.5 [10].

In the protocol, D selects a random symmetric bivariate polynomial $F(x, y)$ such that $F(0, 0) = s$ and sends $f_i(x)$ to P_i. If D is not discarded at the end of the sharing phase, then every honest P_i holds a t degree polynomial $f_i(x)$ which is deduced as $f_i(x) = F(i, x)$. Hence after D distributes the polynomial, a pair of honest players P_i and P_j possess the polynomials such that $f_i(j) = f_j(i)$. By the properties of ICP, no malicious P_i would be able to reveal $f'(x) \neq f(x)$ in the reconstruction phase. Hence irrespective of whether D is honest or malicious, reconstruction of $s = F(0, 0)$ is enforced. D gives the $ICSig$ to every player and every individual player gives the $ICSig$ to every other player to achieve the properties of VSS. The description of the protocol appears in the Figures 3.6, 3.7, and 3.8.

The dealer distributes $f_i(j)$'s to the player P_i in round 1. In addition to this, P_i also shares his random pad r_{ij} with player P_j and the dealer. The first round of AuthVal is executed for all the ICPs in progress in round 2. The players and the dealer broadcast the respective values received by them blinded with the random pad. If player P_i does not receive a polynomial of degree t, then he broadcasts a request to D to broadcast the polynomial. In the third round, the second round of AuthVal is executed for all the ICPs in progress. If any player had to broadcast due to inconsistency in AuthVal, he immediately executes the RevealVal of those ICPs in which he is an intermediary. Additionally, if D had to broadcast in AuthVal then he also broadcasts the corresponding polynomial. The second round of RevealVal is executed in the fourth round of VSS. D is discarded after some local computation if any inconsistency is found.

In the reconstruction phase, RevealVal of the uncompleted ICPs are executed. This takes two rounds.

3.4 VSS Protocol by Garay et al. [6]

Garay et al. [6] came up with a perfect information-theoretic VSS protocol for $t < n/2$ with only two broadcasts in the sharing phase. This is the least known complexity for this model in terms of number of broadcasts. They modified the existing protocols of Weak Broadcast and Gradecast for arbitrary domains.

Garay et al. [6] also used an ICP protocol which was proposed in the same paper. This protocol is a triple of protocols (ICSetup, ICValidate, ICReveal) which successfully achieves ICP for three players: a dealer D, intermediary I, and receiver R. This protocol is based upon the concept of $1_\alpha-$consistency. Let $s, y, z, \alpha \in \mathbb{F}$. We say that the triple (s, y, z) is $1_\alpha-$consistent provided that the three points $(0, s)$, $(1, y)$, and (α, z) are co-linear over $\mathbb{F}$. Based upon the information received by each player in the first two rounds, D may be *in conflict* with I and/or R. So an additional broadcast round is required in ICValidate to resolve the conflicts. This ICP protocol consists of two broadcast rounds in total. The protocol appears in Figure 3.9 [6].

VSS presented in this paper also uses a WSS-without-agreement protocol. WSS-without-agreement uses two broadcast rounds in its sharing phase WSS-Share$(\mathcal{P}, D, s)$ and uses no broadcast in its reconstruction phase WSS-Rec-NoBC$(\mathcal{P}, D, s)$ but without agreement over the reconstructed value. It uses ICP to achieve WSS-without-agreement. The protocol description appears in Figure 3.10 [6].

They first designed a VSS of 3-broadcast rounds (VSS$_{3bc}$). This protocol was inspired from Rabin and Ben-Or [13] which uses univariate polynomial. First D distributes the shares of a t-degree polynomial f and of additional random t-degree polynomials g_k where secret $s = f(0)$.

Protocol	No. of Rounds in Sharing Phase	No. of Broadcast Rounds in Sharing Phase	No. of Malicious Players	Perfect/ Statistical security	Communication complexity
$\frac{n}{3}$-EXP-VSS by Gennaro et al. [7]	3	2	$t < n/3$	Perfect	Exponential
$\frac{n}{3}$-VSS by Fitzi et al. [5]	3	2	$t < n/3$	Perfect	Polynomial
4-round VSS by Patra et al. [10]	4	3	$t < n/2$	Statistical	Polynomial
VSS-Share$_{2bc}$ by Garay et al. [6]	20	2	$t < n/2$	Perfect	Polynomial

Table 3.1: Comparison of different protocols in literature

Each player P_i commits to all shares via WSS. All players then jointly carry out cut-and-choose process in which the players have to reconstruct either g_k or $f + g_k$ for each k, which must be of degree t. Players who cannot reconstruct their shares have them broadcasted by D. The description of VSS_{3bc} is given in Figures 3.12 and 3.13 [6].

They also presented another sub-protocol Moderast. This protocol allows gradecast to take place under the supervision of a designated *moderator*. Each time Moderast is invoked, all the players update their flag f_i to indicate whether the broadcast simulation, i.e. gradecast was successful. In the paper, they have proved that if there exists a constant-round VSS protocol Π which uses a broadcast channel and tolerates t malicious players, then there exists a moderated VSS protocol Π' which uses a gradecast channel and tolerates same number of malicious players. Modercast protocol is given in Figure 3.11 [6].

In the 2-broadcast VSS protocol (VSS_{2bc}) consisting of protocols $\mathsf{VSS\text{-}Share}_{2bc}$ and VSS-Rec, the players first generate the setup required for Gradecast by executing protocol SetupGradecast which consists of one broadcast. Protocols Gradecast and SetupGradecast are discussed in detail in Section 2.4. Then they run the moderated version of VSS_{3bc} where the dealer himself acts as the moderator. But the broadcasts are replaced by Moderast. Another broadcast, the second one, is required to confirm the honesty of the moderator (who is same as dealer). All the players broadcast their flags f_i's indicating whether they trust the moderator or not. If the true flags come out in majority then the sharing phase is successful, otherwise the dealer is disqualified. The description of VSS_{2bc} is given in Figures 3.15 and 3.14 [6].

Though their protocol is optimal in terms of broadcast rounds, but they have left a lot of scope to minimize the overall number of rounds.

*ICP(D,*INT*,s)* **with multiple verifiers**

Distr (D, INT, s):
Round 1:

1. D sends the following to INT:
   (a) A random degree-t polynomial $F(x)$ over $\mathbb{F}$, with $F(0) = s$. Let INT receive $F'(x)$ as the polynomial with $F'(0) = s'$. [1]
   (b) A random degree-t polynomial $R(x)$ over $\mathbb{F}$. Let INT receive $R(x)$ as a t-degree polynomial $R'(x)$.
2. D privately sends the following to each verifier P_i:
   (a) (α_i, v_i, r_i), where $\alpha_i \in \mathbb{F} \setminus \{0\}$ is random (all α_i's are distinct), $v_i = F(\alpha_i)$ and $r_i = R(\alpha_i)$.

AuthVal (D, INT, s):
Round 1: INT chooses a random $d \in_R \mathbb{F} \setminus \{0\}$ and broadcasts $(d, B(x))$ where $B(x) = dF'(x) + R'(x)$.
Round 2: D checks $dv_i + r_i \stackrel{?}{=} B(\alpha_i)$ for $i = 1, ..., n$. If D finds any inconsistency, he broadcasts $s^D = s$.
The polynomial $F'(x)$ (when D does not broadcast s^D in round 2 of AuthVal) or s^D (broadcast by D in round 2 of AuthVal) as held by INT is denoted by *ICSIG(D,*INT*,s)*.

RevealVal (D, INT, s):
Round 1: INT broadcasts *ICSIG(D,*INT*,s)* (i.e., he reveals D's secret contained in *ICSIG(D,*INT*,s)* as $s' = s^D$ or as $s' = F'(0)$).

Round 2: Verifier P_i broadcasts Accept if one of the following conditions holds. (Otherwise, P_i broadcasts Reject.)

1. *ICSIG(D,*INT*,s)* $= s'$, and $s' = s^D$.
2. *ICSIG(D,*INT*,s)* $= F'(x)$, and one of the following holds.
   (a) C_1**:** $v_i = F'(\alpha_i)$; OR
   (b) C_2**:** $B(\alpha_i) \neq dv_i + r_i$ ($B(x)$ was broadcasted by INT during AuthVal).

Local Computation (By Every Verifier): If at least $t + 1$ verifiers have broadcasted Accept during round 2 of RevealVal then accept *ICSIG(D,*INT*,s)* and output s' or $F'(0)$ (depending on whether *ICSIG(D,*INT*,s)* is s' or $F'(x)$). Else reject *ICSIG(D,*INT*,s)*.

[1] If INT is honest, then $F'(x) = F(x)$.

Figure 3.5: ICP with multiple verifiers

4-round VSS: Sharing Phase

Inputs: The dealer has a secret s. Let D be the dealer and let $F(x, y)$ be a symmetric bivariate polynomial of degree t in each variable. Let $F(0, 0) = s$.

Round 1: Let $f_i(x)$ be defined as $F(i, x)$. Let $r_{ij} \in_R \mathbb{F}$ for each P_i, P_j.

1. Execute $\mathsf{Distr}(D, P_i, f_i(j))$. Let the corresponding value received by P_i be $f'_i(j)$.
2. Execute $\mathsf{Distr}(P_i, P_j, r_{ij})$. Let the corresponding value received by P_j be r'_{ij}.
3. Execute $\mathsf{Distr}(P_i, D, r_{ij})$. Let the corresponding value received by D be r^D_{ij}.

Round 2:

1. Execute $\mathsf{AuthVal}^{(1)}(D, P_i, f_i(j))$, $\mathsf{AuthVal}^{(1)}(P_i, P_j, r_{ij})$, and $\mathsf{AuthVal}^{(1)}(P_i, D, r_{ij})$.
2. P_i broadcasts $a_{ij} = f'_i(j) + r_{ij}$ and $b_{ij} = f'_i(j) + r'_{ji}$.
3. D broadcasts $a^D_{ij} = f_i(j) + r^D_{ij}$ and $b^D_{ij} = f_i(j) + r^D_{ji}$.
4. If P_i received $f'_i(x)$ which is not a polynomial of degree t, then P_i broadcasts a request asking D to broadcast a t-degree $f^D_i(x)$.

Round 3:

1. Execute $\mathsf{AuthVal}^{(2)}(D, P_i, f_i(j))$. If D broadcasted the secret in $\mathsf{AuthVal}^{(2)}(D, P_i, f_i(j))$, then he broadcasts the corresponding polynomial $f^D_i(x) = f_i(x)$ and executes $\mathsf{RevealVal}^{(1)}(P_i, D, r_{ik})$ and $\mathsf{RevealVal}^{(1)}(P_k, D, r_{ki})$ for all k.
2. Execute $\mathsf{AuthVal}^{(2)}(P_i, P_j, r_{ij})$. If P_i broadcasted the secret in $\mathsf{AuthVal}^{(2)}(P_i, P_j, r_{ij})$, then he also executes $\mathsf{RevealVal}^{(1)}(D, P_i, f_i(j))$ and $\mathsf{RevealVal}^{(1)}(P_j, P_i, r_{ji})$.
3. Execute $\mathsf{AuthVal}^{(2)}(P_i, D, r_{ij})$. If P_i broadcasted the secret in $\mathsf{AuthVal}^{(2)}(P_i, D, r_{ij})$, then he also executes $\mathsf{RevealVal}^{(1)}(D, P_i, f_i(j))$ and $\mathsf{RevealVal}^{(1)}(P_j, P_i, r_{ji})$.
4. If P_i requested D to broadcast $f^D_i(x)$, then D broadcasts $f^D_i(x) = f_i(x)$.
5. If $a_{ij} \neq a^D_{ij}$ or $a_{ij} = \perp$, then D broadcasts $f^D_i(x) = f_i(x)$ and executes $\mathsf{RevealVal}^{(1)}(P_i, D, r_{ik})$ and $\mathsf{RevealVal}^{(1)}(P_k, D, r_{ki})$ for all k.
6. If $a_{ij} \neq b_{ji}$ or $a_{ji} \neq b_{ij}$ or $a_{ij} \neq a^D_{ij}$ or $b_{ij} \neq b^D_{ij}$, then P_i executes $\mathsf{RevealVal}^{(1)}(D, P_i, f_i(j))$ and $\mathsf{RevealVal}^{(1)}(P_j, P_i, r_{ji})$.

Figure 3.6: 4-round VSS: Sharing Phase

Round 4: Corresponding $\mathsf{RevealVal}^{(2)}$ executions are completed in this round.

Local Computation: D is **discarded** if for some P_i, P_j:

1. D broadcasted more than t shares (i.e. polynomials of the form $f_i^D(x)$).
2. $f_i^D(j) \neq f_j^D(i)$.
3. $a_{ij}^D \neq b_{ji}^D$.
4. P_i revealed $f_i'(j)$ and $f_i'(j) \neq f_i^D(j)$ or $f_i'(j) \neq f_j^D(i)$.
5. P_i, P_j revealed $f_i'(j)$, $f_j'(i)$ (respectively) and $f_i'(j) \neq f_j'(i)$.
6. D did not broadcast $f_i^D(x)$ and P_i did not broadcast the secret in $\mathsf{AuthVal}^{(2)}(P_i, D, r_{ij})$ and D executed $\mathsf{RevealVal}(D, P_i, r_{ij})$ and $a_{ij}^D - r_{ij}^D \neq f_j^D(i)$.

Figure 3.7: 4-round VSS: Sharing Phase continued

4-round VSS: Reconstruction Phase

Round 1-2:

1. Execute $\mathsf{RevealVal}(D, P_i, f_i(j))$.
2. Execute $\mathsf{RevealVal}(P_j, P_i, r_{ji})$.

Local Computation: Let $P_i \in UNHAPPY$ if D broadcasted $f_i^D(x)$. Construct REC in the following way:

1. $P_i \in REC$ if $P_i \in UNHAPPY$. In this case, define $f_i'(x) = f_i^D(x)$.
2. $P_i \in REC$ if he successfully executed $\mathsf{RevealVal}(D, P_i, f_i(j))$ for all j. The values $\{f_i'(j)\}_j$ must lie on a t-degree polynomial $f_i'(x)$.

Delete $P_i \notin UNHAPPY$ from REC if

1. P_i revealed $f_i'(j)$ and $f_i'(j) \neq f_j^D(i)$ for some $P_j \in UNHAPPY$.
2. P_j revealed r_{ij}' and $f_i'(j) + r_{ij}' \neq a_{ij}$.
3. If for some P_j, P_j did not broadcast in $\mathsf{AuthVal}^{(2)}(P_j, P_i, r_{ji})$ and $b_{ij} - r_{ji}' \neq f_i'(j)$.
4. If for some P_j, P_i successfully executed $\mathsf{RevealVal}(D, P_i, f_i(j))$ in the sharing phase but in the reconstruction phase reconstructed a different value for $f_i'(j)$.

Reconstruct a symmetric bivariate polynomial $F'(x, y)$ of degree t from $\{f_i'(x)\}_{P_i \in REC}$. Output $s' = F'(0, 0)$.

Figure 3.8: 4-round VSS: Reconstruction Phase

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

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

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

Round 1: I chooses a random value $d \in \mathbb{F}$ and sends it to D.

Round 2: D sends the triple $(d, s' + ds, y' + dy)$ to R.

Round 3: Each player broadcasts the values he sent and received in the previous two rounds. I broadcasts his view of the triple $(d, s' + ds, y' + dy)$. Additionally, R checks that $(s' + ds, y' + dy, z' + dz)$ is $1_\alpha-$consistent; if not R broadcasts "reject values."
Based on these broadcasts D may be *in conflict* with I and/or R:

1. D and I are in conflict if they disagree about the value of the triple $(d, s' + ds, y' + dy)$. [Or if they conflict in a parallel instance.]
2. D and R are in conflict if they disagree about what D sent in round 2, or if D is *not* in conflict with I, and R broadcast "reject values." [Or if they conflict in a parallel instance.]

If no such conflicts arise, then all players are satisfied and the phase ends here. Otherwise continue to round 4.

Round 4: If D, I are in conflict, then D broadcasts (s, y) and R adjusts z if necessary so that (s, y, z) is $1_\alpha-$consistent. This is done regardless whether D, R are in conflict or not, and the phase ends here.
Otherwise, it must be that D, R are in conflict, but D, I are not. In this case D broadcasts (z, α) and I adjusts y if necessary so that (s, y, z) is $1_\alpha-$consistent.

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

Round 1: I sends (s, y) to R, who accepts s if and only if (s, y, z) is $1_\alpha-$consistent.

Figure 3.9: ICP

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

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

Round 2-5: For each $P_i, P_j \in \mathcal{P} - \{D\}$, run ICValidate($D, P_i, P_j, s_i$).

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

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

Figure 3.10: WSS with two broadcast rounds

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

Round 1-6: P_i gradecasts the message m.

Round 7-12: The moderator P^{**} gradecasts the message he output in the previous step.

- Let (m_j, g_j) and (m'_j, g'_j) be the outputs of player P_j in steps 1 and 2, respectively. Within the underlying execution of Π', player P_j will use m'_j as the message "broadcast" by P_i.
- Furthermore, P_j sets $f_j := 0$ if (1) $g'_i \neq 1$, or (2) $m'_i \neq m_i$ and $g_i = 1$.

Figure 3.11: Modercast

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

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

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

Round 6: The players have just completed WSS-Share step 4/ICValidate step 3. In the next step (corresponding to WSS-Share step 5/ICValidate step 4) the WSS/IC dealer will resolve conflicts. Instead of doing so immediately, let BC_i denote the broadcast which P_i would make. P_i first sends-to-all BC_i.

Also, if D conflicted with any P_i in the previous step (namely in ICValidate step 3) then in the following round D will broadcast *all* the values $(s_i, \{t_{ki}\}_k)$. For now, D sends-to-all these values, which we call *public pieces*.

Round 7: Now P_i broadcasts BC_i, which completes WSS-Share step 5/ICValidate step 4, and D broadcasts the values $(s_i, \{t_{ki}\}_k)$ which he sent-to-all in the previous step. Of course each player broadcasts his view of the previous step; if it is not the case that at least $t+1$ players agree that P_i's broadcast this round matches what he told them in the previous round, then P_i is disqualified.
Additionally, each $P_i \neq D$ broadcasts a random challenge $C_i \in \{0,1\}^\kappa$ for D and for the other P_j's. The challenge indicates, for each index $k \in [\kappa n]$ assigned to P_i (κ such in total), whether:

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

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

Round 8: $\forall k \in [\kappa n]$, $j \in [n]$, P_i participates in $\mathsf{WSS\text{-}Rec\text{-}NoBC}(D, v_{kj})$ and $\mathsf{WSS\text{-}Rec\text{-}NoBC}(P_j, v_{kj}^*)$. P_i's outputs from these protocols are $v_{kj}^{(i)}$ and $v_{kj}^{*(i)}$, respectively.

Figure 3.12: VSS-Share$_{3bc}$

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

Round 9: Each P_i broadcasts his view of the previous round - namely, the reconstructed shares $v_{kj}^{(i)}$ and $v_{kj}^{*(i)}$, for all k, j.
If a majority of players agrees on a non-$\perp$ reconstructed value for v_{kj} (resp. v_{kj}^{*}), then such value is the *broadcast (BC) consensus* for the given commitment, and the players who agree *participate in the consensus*. If no BC consensus exists, or if the player who shared the value does not participate, then the sharing player is disqualified. Consequently, if D is not so disqualified, then there exists a BC consensus (which D endorses) for all v_{kj}. Assuming this is the case, then D is nevertheless disqualified if for any k, the set of shares $\{v_{kj}\}_j$, together with appropriate public pieces, does not lie on a polynomial of degree $\leqslant t$.
In addition to broadcasting his view as described above, D also accuses player P_j, by publicly broadcasting the shares $(s_j, \{t_{kj}\}_k)$, if either of the following occurred:

1. D output $\perp$ in any WSS-Rec-NoBC instance for which P_j was dealer; or
2. D reconstructed an incorrect value for P_j's share of any challenge polynomial ($v_{kj}^{*(D)} \neq v_{kj}$).

If any such public pieces fail to lie on the appropriate degree-t polynomial, or if D neglects to accuse P_j when there exists a BC consensus that ($v_{kj}^{*} \neq v_{kj}$), then D is disqualified.
Let HAPPY denote the set of **non-disqualified** players who were **not accused** by D. If $|\mathsf{HAPPY}| < n - t$, then D is disqualified.

Figure 3.13: VSS-Share$_{3bc}$ continued

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

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

Figure 3.14: VSS-Rec

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

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

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

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

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

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

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

Figure 3.15: VSS-Share$_{2bc}$

Chapter 4

New ICP and VSS Protocol

We now turn our attention to contributions of this work which describes the modified ICP and VSS protocol tolerating $t < n/2$ malicious players. In the following sections, these protocols are discussed in detail with their proofs. The protocols below discussed are:

- ICP using Gradecast
- VSS using modified ICP

4.1 ICP using Gradecast

In Patra et al. [10], there are two broadcast rounds each in AuthVal and RevealVal. We show how we can reduce its broadcast round complexity in the ICP using few additional rounds, including just one broadcast. Using one additional round of broadcast, we can simulate sufficiently many gradecast channels later on.

Gradecast is a very important technique to improve the efficiency of various protocols. It can be used to replace broadcasts in many places without losing the properties of the protocol. We propose the modified ICP which uses gradecast, [6] by which we can reduce the number of broadcasts from four to only two. We eventually use this ICP to come up with a new VSS protocol.

In Figure 4.1 and 4.2, we present the Information Checking Protocol using gradecast. Firstly, all the three rounds of protocol SetupGradecast are executed to create the setup for 2-cast. Then, one broadcast in the AuthVal sub-protocol is replaced by gradecast and one broadcast in RevealVal of [10] is replaced by communication over point-to-point channels. In RevealVal, INT sends the *ICSIG(D,*INT*,s)* over point-to-point channels to all the verifiers instead of broadcasting. All the honest verifiers accept or reject INT accordingly as shown in the Figure 4.1. All the verifiers also broadcast their outputs at the end and reach an agreement after some local computation.

ICP(D,INT*,s)*

SetupGradecast:
Round 1-3: Players execute all three rounds of the protocol SetupGradecast.

Distr (D, INT, s):
Round 1:

1. D sends the following to INT:
   (a) A random degree-t polynomial $F(x)$ over $\mathbb{F}$, with $F(0) = s$. Let INT receive $F'(x)$ as the polynomial with $F'(0) = s'$. [1]
   (b) A random degree-t polynomial $R(x)$ over $\mathbb{F}$. Let INT receive $R(x)$ as a t-degree polynomial $R'(x)$.
2. D privately sends the following to each verifier P_i:
   (a) (α_i, v_i, r_i), where $\alpha_i \in \mathbb{F} \setminus \{0\}$ is random (all α_i's are distinct), $v_i = F(\alpha_i)$ and $r_i = R(\alpha_i)$.

AuthVal (D, INT, s):
Round 1: INT chooses a random $d \in_R \mathbb{F} \setminus \{0\}$ and broadcasts $(d, B(x))$ where $B(x) = dF'(x) + R'(x)$.
Round 2-7: D checks $dv_i + r_i \stackrel{?}{=} B(\alpha_i)$ for $i = 1, ..., n$. If D finds any inconsistency, he gradecasts $s^D = s$, else gradecasts $\perp$ (indicating no issues).

RevealVal (D, INT, s):
Round 1: Let P_i and INT received messages m_i^D and m_{INT}^D from D during AuthVal with grade g_i and g_{INT} respectively.
If $m_{\mathsf{INT}}^D \neq \perp$ then INT sets $s' = m_{\mathsf{INT}}^D$ else sets $s' = F'(x)$. Here, INT actually reveals D's secret to P_i as s'.
INT sends *ICSIG(D,*INT*,s)* $= s'$ to P_i over point-to-point channel.

If $g_i = 0$, P_i blames D and accepts INT. If not, i.e. $g_i = 1$, following cases can happen:
Case 1 ($m_i^D = s^D$):

1. P_i accepts INT if $m_i^D =$ *ICSIG(D,*INT*,s)* $= s'$.
2. P_i rejects INT otherwise.

Figure 4.1: ICP using gradecast

Case 2 ($m_i^D = \perp$):
P_i accepts INT if either of the following condition holds:

1. C_1: $v_i = F(\alpha_i)$; OR
2. C_2: $B(\alpha_i) \neq dv_i + r_i$.

P_i rejects INT otherwise.

Round 2: Verifier P_i broadcasts Accept if he accepts INT in the previous round, otherwise broadcasts Reject.
Local Computation (By Every Verifier): If at least $t+1$ verifiers have broadcasted Accept during round 2 of RevealVal then accept *ICSIG(D,INT,s)* and output s' or $F'(0)$ (depending on whether *ICSIG(D,INT,s)* is s' or $F'(x)$). Else reject *ICSIG(D,INT,s)*.

[1] If INT is honest, then $F'(x) = F(x)$.

Figure 4.2: ICP using gradecast continued

Notably, the number of overall rounds here are increased due to gradecast but the broadcast rounds have come down from four to three.

In our protocols, we use <SubProtocolName>$^{(i)}$ to denote the i^{th} round of the <SubProtocolName> sub-protocol. For example, AuthVal$^{(1)}$ denotes the first round of AuthVal.

The ICP with all the broadcasts satisfies all the required properties [10]. We now prove that ICP with gradecasts also satisfies those properties.

4.1.1 Proof of Correctness of ICP

Claim 4.1 *If D and* INT *are honest then D will always gradecast $\perp$ during* AuthVal*.*

Proof:

Since INT is honest, he will correctly broadcast $(d, B(x))$ in round 1 of AuthVal. So during round 2-7 of AuthVal, D will find that $B(\alpha_i) = dv_i + r_i$ is satisfied for all $i = 1, ..., n$. Thus D will never gradecast s, but $\perp$ during AuthVal. □

Lemma 4.1 *If D and* INT *are honest, then* ICSIG(D,INT,s) *produced by* INT *during* RevealVal *will be accepted by each honest verifier.*

Proof: Since INT is honest, then $F'(x) = F(x)$, Also an honest verifier will have $v_i = F(\alpha_i)$ and $r_i = R(\alpha_i)$. Moreover by Claim 4.1, D will gradecast $\perp$ during AuthVal. Hence INT will set *ICSIG(D,INT,s)* $= F'(x) = F(x)$. Now an honest verifier will accept INT as the condition C_1 will hold. Since there are at least $t+1$ honest verifiers, *ICSIG(D,INT,s)* will be accepted by every honest verifier. □

Claim 4.2 *If $(F(x), R(x))$ held by an honest* INT *and* (α_i, v_i, r_i) *held by an honest verifier* P_i *satisfies* $F(\alpha_i) \neq v_i$ *and* $R(\alpha_i) \neq r_i$*, then except with probability* $2^{-\Theta(\kappa)}$, $B(\alpha_i) \neq dv_i + r_i$*.*

Proof: The proof of this claim is given in Claim 3 of [10]. The claim will hold as it is independent of the distribution scheme. Hence this claim is true in case of ICP with gradecast as well. □

Lemma 4.2 *If* INT *is honest then at the end of* AuthVal*,* INT *possesses an* ICSIG(D,INT,s)*, which will be accepted in* RevealVal *by each honest verifier, except with probability* $2^{-\Theta(\kappa)}$*.*

Proof: Let P_i be an honest verifier. If D is honest, the lemma follows from Lemma 4.1. When D is malicious, there can be two cases as described below:

1. *ICSIG(D,*INT*,s)* $= m^D_{\mathsf{INT}}$. In this case an honest INT will send $s' = m^D_{\mathsf{INT}}$ to the verifier. If $g_i = 0$, then P_i will blame D and accept INT. Moreover, if $g_i = 1$, then m^D_{INT} will be equal to m^D_i, due to the graded consistency property of Gradecast, and INT will be accepted by P_i. Hence the lemma will hold without any error.

2. *ICSIG(D,*INT*,s)* $= F'(x)$. An honest INT will have $F'(x) = F(x)$ and $R'(x) = R(x)$. We have the following cases depending upon the values held by INT (i.e. $F(x), R(x)$) and P_i (i.e. (α_i, v_i, r_i)):

   (a) If $F(\alpha_i) = v_i$: Here P_i will broadcast Accept without any error probability as C_1 (i.e. $F(\alpha_i) = v_i$) will hold.

   (b) If $F(\alpha_i) \neq v_i$ and $R(\alpha_i) = r_i$: Here P_i will broadcast Accept without any error probability as C_2 (i.e. $B(\alpha_i) \neq dv_i + r_i$) will hold.

   (c) If $F(\alpha_i) \neq v_i$ and $R(\alpha_i) \neq r_i$: Here P_i will broadcast Accept except with probability $2^{-\Theta(\kappa)}$, as C_2 will hold from Claim 4.2.

Hence each honest verifier will broadcast Accept during RevealVal except with probability $2^{-\Theta(\kappa)}$. This completes the proof. □

Lemma 4.3 *If* D *is honest then during* RevealVal*, with probability at least* 1 - $2^{-\Theta(\kappa)}$*, every* ICSIG(D,INT,s) *revealed as* $s' \neq s$ *or by a corrupted* INT *will be rejected by each honest verifier.*

Proof: Let P_i be an honest verifier. Let $s' = F'(0)$. Here again, the proof can be divided into two cases based upon the value of *ICSIG(D,*INT*,s)* as described below:

1. *ICSIG(D,*INT*,s)* $= m^D_{\mathsf{INT}}$. The lemma will hold as an honest dealer D would have gradecasted $s^D = s$ with grade $g = 1$ to all the honest players. Hence P_i will broadcast Accept if and only if $s' = s$.

2. *ICSIG(D,*INT*,s)* $= F'(x)$. Here P_i will broadcast Accept only in below mentioned two cases:

   (a) $F'(\alpha_i) = v_i$. Since P_i and D are honest, INT has no information about α_i and r_i. The only way for INT to ensure that $F'(\alpha_i) = v_i = F(\alpha_i)$ is by guessing α_i correctly. The probability of that is at most $\frac{1}{|\mathbb{F}|-1} = 2^{-\Theta(\kappa)}$.

   (b) $B(\alpha_i) \neq dv_i + r_i$. This case will never happen since an honest D would have gradecasted s during AuthVal if this was the case.

   This proves that P_i will broadcast Accept with a probability of at most $2^{-\Theta(\kappa)}$ if $F'(x) \neq F(x)$. Since there are only $t < n/2$ malicious players, the malicious INT's *ICSIG(D,*INT*,s)* will be rejected.

This ends the proof. □

Lemma 4.4 *If D and* INT *are honest, then at the end of* AuthVal*, s is information theoretically secure from the adversary $\mathcal{A}$ (that controls at most t verifiers in $\mathcal{P}$).*

Proof: If both D and INT are honest, then D will gradecast $\perp$ during AuthVal. A corrupt verifier P_i will have knowledge of no more than α_i, r_i, d and $B(x)$. So the central adversary $\mathcal{A}$ will have knowledge of at most t points on the polynomials $F(x)$ and $R(x)$. Since $F(x)$ and $R(x)$ are independent, the constant coefficient of $F(x)$ will be information theoretically secure even after the knowledge of d and $dF(x) + R(x)$. Hence the lemma. □

Theorem 4.1 *Proposed ICP with gradecast satisfies all the properties required by an Information Checking Protocol.*

Proof: The theorem follows from Lemmas 4.1, 4.2, 4.3, 4.4. □

4.2 VSS using modified ICP

Now we can plug in the modified ICP into the VSS protocol given by Patra et al. [10]. The protocol being discussed here is for the statistical case, i.e. it may not achieve VSS with a negligible probability. We here try to come up with a protocol with lesser broadcast rounds than the one in [10] using gradecast. [6] have used the gradecast technique to come up with a

VSS protocol in constant number of rounds. We are able to come up with even lesser number of overall rounds, though introducing a negligible probability of error.

In this section we present our new VSS protocol for $t < n/2$ which consists of two broadcasts and constant number of overall rounds. This protocol uses the modified ICP with gradecast which is discussed in Section 4.1.

This protocol is inspired by the VSS protocol of [10]. In the protocol, firstly, SetupGradecast is run to create the setup for gradecast required by the ICP protocol. D selects a random symmetric bivariate polynomial $F(x, y)$ such that $F(0, 0) = s$ and sends $f_i(x)$ to P_i in round 1. In round 3, the first step of AuthVal is executed. SetupGradecast completes its execution by this round and the setup is ready for gradecasting. The steps 2-4 in round 2 of [10] are run in parallel with $\mathsf{AuthVal}^{(1)}$. The last six steps of AuthVal are merged with round 4 of [10] in the same manner. If D is not discarded at the end of the sharing phase, then every honest P_i holds a t degree polynomial $f_i(x)$ which is deduced as $f_i(x) = F(i, x)$. Hence after D distributes the polynomial, a pair of honest players P_i and P_j possess the polynomials such that $f_i(j) = f_j(i)$. By the properties of ICP, no malicious P_i would be able to reveal $f'(x) \neq f(x)$ in the reconstruction phase. Hence irrespective of whether D is honest or malicious, reconstruction of $s = F(0, 0)$ is enforced. D gives the $ICSig$ to every player and every individual player gives the $ICSig$ to every other player to achieve the properties of VSS. Now we present the VSS protocol in Figures 4.3, 4.4, 4.5.

4.2.1 Proof of Correctness

Claim 4.3 *In our* 10-Round-VSS *protocol,* **Correctness1, Correctness2, Correctness3** *hold for concurrent executions of ICP(P_i, P_j, r_{ij}) and ICP(P_i, D, r_{ij}).*

Proof: This is because the polynomials used in ICP(P_i, P_j, r_{ij}) and ICP(P_i, D, r_{ij}) are random and independent of each other. Also when D is honest, Secrecy holds for concurrent executions of ICP(P_i, P_j, r_{ij}) and ICP(P_i, D, r_{ij}). □

Lemma 4.5 (Secrecy) *Protocol* 10-round-VSS *satisfies perfect secrecy.*

Proof: When D is honest, for every honest P_i, P_j, the values $f_i(j), f_j(i)$ are never broadcasted in the clear during the sharing phase. Therefore, the adversary knows at most t values on $f_i(x)$ for an honest P_i. Therefore, he does not have any information about $f_i(0)$. As a result, the adversary has exactly t polynomials $\{f_j(x)|P_j \text{ is dishonest}\}$ and no information on the set $\{f_i(0)|P_i \text{ is honest}\}$. Hence $F(0, 0) = s$ is information theoretically private. □

10-round VSS: Sharing Phase

Inputs: The dealer has a secret s. Let D be the dealer and let $F(x, y)$ be a symmetric bivariate polynomial of degree t in each variable. Let $F(0, 0) = s$.

Round 1: Let $f_i(x)$ be defined as $F(i, x)$. Let $r_{ij} \in_R \mathbb{F}$ for each P_i, P_j.

1. Execute $\mathsf{Distr}(D, P_i, f_i(j))$. Let the corresponding value received by P_i be $f'_i(j)$.
2. Execute $\mathsf{Distr}(P_i, P_j, r_{ij})$. Let the corresponding value received by P_j be r'_{ij}.
3. Execute $\mathsf{Distr}(P_i, D, r_{ij})$. Let the corresponding value received by D be r^D_{ij}.
4. Execute $\mathsf{SetupGradecast}^{(1)}$.

Round 2: Execute $\mathsf{SetupGradecast}^{(2)}$.

Round 3:

1. Execute $\mathsf{AuthVal}^{(1)}(D, P_i, f_i(j))$, $\mathsf{AuthVal}^{(1)}(P_i, P_j, r_{ij})$, and $\mathsf{AuthVal}^{(1)}(P_i, D, r_{ij})$.
2. P_i broadcasts $a_{ij} = f'_i(j) + r_{ij}$ and $b_{ij} = f'_i(j) + r'_{ji}$.
3. D broadcasts $a^D_{ij} = f_i(j) + r^D_{ij}$ and $b^D_{ij} = f_i(j) + r^D_{ji}$.
4. If P_i received $f'_i(x)$ which is not a polynomial of degree t, then P_i broadcasts a request asking D to broadcast a t-degree $f^D_i(x)$.
5. Execute $\mathsf{SetupGradecast}^{(3)}$.

Figure 4.3: 10-round VSS: Sharing Phase

Let us define the set $UNHAPPY$ to consist of players P_j whose share (the polynomial $f_j(x)$) was broadcasted by D in the sharing phase.

Claim 4.4 *If D is not discarded and P_i is honest, then for every $P_j \in UNHAPPY$, $f'_i(j) = f^D_j(i)$.*

Proof: If $P_i \in UNHAPPY$, then $f'_i(x) = f^D_i(x)$, and since D is not discarded, we have $f'_i(j) = f^D_j(i)$ for every $P_j \in UNHAPPY$.

Now let $P_i \notin UNHAPPY$. We have two cases:

Case 1: $P_j \in UNHAPPY$ because P_j received an incorrect polynomial from D. In this case, P_j would not have broadcasted anything for a_{ji} and b_{ji}. Hence, in round 4, $a_{ij} \neq b_{ji}$. Consequently,

Round 4:

1. Execute $\mathsf{AuthVal}^{(2)}(D, P_i, f_i(j))$. If D gradecasted the secret in $\mathsf{AuthVal}^{(2)}(D, P_i, f_i(j))$, then he executes $\mathsf{RevealVal}^{(1)}(P_i, D, r_{ik})$ and $\mathsf{RevealVal}^{(1)}(P_k, D, r_{ki})$ for all k.

2. Execute $\mathsf{AuthVal}^{(2)}(P_i, P_j, r_{ij})$. If P_i gradecasted the secret in $\mathsf{AuthVal}^{(2)}(P_i, P_j, r_{ij})$, then he also executes $\mathsf{RevealVal}^{(1)}(D, P_i, f_i(j))$ and $\mathsf{RevealVal}^{(1)}(P_j, P_i, r_{ji})$.

3. Execute $\mathsf{AuthVal}^{(2)}(P_i, D, r_{ij})$. If P_i gradecasted the secret in $\mathsf{AuthVal}^{(2)}(P_i, D, r_{ij})$, then he also executes $\mathsf{RevealVal}^{(1)}(D, P_i, f_i(j))$ and $\mathsf{RevealVal}^{(1)}(P_j, P_i, r_{ji})$.

4. If $a_{ij} \neq a_{ij}^D$ or $a_{ij} = \perp$, then D executes $\mathsf{RevealVal}^{(1)}(P_i, D, r_{ik})$ and $\mathsf{RevealVal}^{(1)}(P_k, D, r_{ki})$ for all k.

5. If $a_{ij} \neq b_{ji}$ or $a_{ji} \neq b_{ij}$ or $a_{ij} \neq a_{ij}^D$ or $b_{ij} \neq b_{ij}^D$, then P_i executes $\mathsf{RevealVal}^{(1)}(D, P_i, f_i(j))$ and $\mathsf{RevealVal}^{(1)}(P_j, P_i, r_{ji})$.

Round 5-9: Execute $\mathsf{AuthVal}^{(3-7)}(D, P_i, f_i(j))$, $\mathsf{AuthVal}^{(3-7)}(P_i, P_j, r_{ij})$, and $\mathsf{AuthVal}^{(3-7)}(P_i, D, r_{ij})$.

Round 10:

1. Corresponding $\mathsf{RevealVal}^{(2)}$ executions are completed in this round.

2. If D gradecasted the secret in $\mathsf{AuthVal}^{(2)}(D, P_i, f_i(j))$, then he broadcasts the corresponding polynomial $f_i^D(x) = f_i(x)$.

3. If P_i requested D to broadcast $f_i^D(x)$, then D broadcasts $f_i^D(x) = f_i(x)$.

4. If $a_{ij} \neq a_{ij}^D$ or $a_{ij} = \perp$, then D broadcasts $f_i^D(x) = f_i(x)$.

Local Computation: D is **discarded** if for some P_i, P_j:

1. D broadcasted more than t shares (i.e. polynomials of the form $f_i^D(x)$).

2. $f_i^D(j) \neq f_j^D(i)$.

3. $a_{ij}^D \neq b_{ji}^D$.

4. P_i revealed $f_i'(j)$ and $f_i'(j) \neq f_i^D(j)$ or $f_i'(j) \neq f_j^D(i)$.

5. P_i, P_j revealed $f_i'(j)$, $f_j'(i)$ (respectively) and $f_i'(j) \neq f_j'(i)$.

6. D did not broadcast $f_i^D(x)$ and P_i did not gradecast the secret in $\mathsf{AuthVal}^{(2-7)}(P_i, D, r_{ij})$ and D executed $\mathsf{RevealVal}(D, P_i, r_{ij})$ and $a_{ij}^D - r_{ij}^D \neq f_j^D(i)$.

Figure 4.4: 10-round VSS: Sharing Phase continued

10-round VSS: Reconstruction Phase

Round 1-2:

1. Execute $\mathsf{RevealVal}(D, P_i, f_i(j))$.
2. Execute $\mathsf{RevealVal}(P_j, P_i, r_{ji})$.

Local Computation: Let $P_i \in UNHAPPY$ if D broadcasted $f_i^D(x)$. Construct REC in the following way:

1. $P_i \in REC$ if $P_i \in UNHAPPY$. In this case, define $f_i'(x) = f_i^D(x)$.
2. $P_i \in REC$ if he successfully executed $\mathsf{RevealVal}(D, P_i, f_i(j))$ for all j. The values $\{f_i'(j)\}_j$ must lie on a t-degree polynomial $f_i'(x)$.

Delete $P_i \notin UNHAPPY$ from REC if

1. P_i revealed $f_i'(j)$ and $f_i'(j) \neq f_j^D(i)$ for some $P_j \in UNHAPPY$.
2. P_j revealed r_{ij}' and $f_i'(j) + r_{ij}' \neq a_{ij}$.
3. If for some P_j, P_j gradecasted $\perp$ in $\mathsf{AuthVal}^{(2-7)}(P_j, P_i, r_{ji})$ and $b_{ij} - r_{ji}' \neq f_i'(j)$.
4. If for some P_j, P_i successfully executed $\mathsf{RevealVal}(D, P_i, f_i(j))$ in the sharing phase but in the reconstruction phase reconstructed a different value for $f_i'(j)$.

Reconstruct a symmetric bivariate polynomial $F'(x, y)$ of degree t from $\{f_i'(x)\}_{P_i \in REC}$. Output $s' = F'(0, 0)$.

Figure 4.5: 10-round VSS: Reconstruction Phase

P_i would execute $\mathsf{RevealVal}(D, P_i, f_i(j))$ and if $f_i'(j) \neq f_j^D(i)$, D would have been disqualified. Hence, the claim holds.

Case 2: $P_j \in UNHAPPY$ because for some player P_k, D broadcasted $f_j^D(x)$ because he gradecasts the secret in $\mathsf{AuthVal}^{(2-7)}(D, P_j, f_j(k))$ or because $a_{jk} \neq a_{jk}^D$ or $a_{jk} = \perp$.

In this case, D also executes $\mathsf{RevealVal}(P_i, D, r_{ij})$ (in Steps 1,4 of round 4). There are two subcases to consider now. First, if P_i gradecasted the secret (with grade 1 since he is honest) in $\mathsf{AuthVal}^{(2-7)}(P_i, D, r_{ij})$, then he also executes $\mathsf{RevealVal}(D, P_i, f_i(j))$ and a contradiction (if one exists) is visible to all players, and D would be discarded. On the other hand, if P_i gradecasted $\perp$ in $\mathsf{AuthVal}^{(2-7)}(P_i, D, r_{ij})$, then D has to reveal the correct value of r_{ij} (follows from **Correctness3**), i.e. $r_{ij}^D = r_{ij}$. Since $P_i \notin UNHAPPY$, we have $a_{ij}^D = a_{ij}$. Therefore, for an honest P_i, we have $a_{ij}^D - r_{ij}^D = a_{ij} - r_{ij} = f_i'(j)$. If $a_{ij}^D - r_{ij}^D \neq f_j^D(i)$, then D is discarded (in Step 6 of Local Computation). Therefore, $f_i'(j) = f_j^D(i)$. □

Claim 4.5 *If D is not discarded and P_i is honest, then $P_i \in REC$.*

Proof: If $P_i \in UNHAPPY$, then $P_i \in REC$. Assume $Pi \notin UNHAPPY$. Honest P_i successfully executes $\mathsf{RevealVal}(D, P_i, f_i(j))$, and player P_j can successfully reveal r'_{ij} in $\mathsf{RevealVal}(P_i, P_j, r'_{ij})$ only for $r'_{ij} = r_{ij}$ (follows from **Correctness3**). We now show that none of rules that delete P_i from REC apply to an honest P_i.

1. By Claim 4.4, we have that for each $P_j \in UNHAPPY$, $f'_i(j) = f_j^D(i)$.
2. Since revealed r'_{ij} is always equal to r_{ij} (by **Correctness2**), $a_{ij} = f'_i(j) + r'_{ij}$.
3. If P_j gradecasted $\perp$ to P_i in $\mathsf{AuthVal}^{(2-7)}(P_j, P_i, r_{ji})$, then an honest P_i will be successful in revealing the pad r'_{ji} (irrespective of the grade output by P_i) which he used while broadcasting a_{ij}, b_{ij}. Hence $b_{ij} - r'_{ji} = f'_i(j)$.
4. If P_i is honest, he will reveal the same values as he had done before in the sharing phase.

□

Claim 4.6 *If D is not discarded, then $f'_i(j) = f'_j(i)$ for every honest P_i, P_j.*

Proof: We have 4 cases:
Case 1: $P_i, P_j \in UNHAPPY$.
In this case, $f'_i(x) = f_i^D(x)$ and $f'_j(x) = f_j^D(x)$. Since D was not discarded, the claim holds.
Case 2: $P_i, P_j \notin UNHAPPY$.
For honest P_i, P_j, if $f'_i(j) \neq f'_j(i)$, then $a_{ij} \neq b_{ji}$ and $a_{ji} \neq b_{ij}$. Consequently, P_i would execute $\mathsf{RevealVal}(D, P_i, f_i(j))$ and P_j would execute $\mathsf{RevealVal}(D, P_j, f_j(i))$. If $f'_i(j) \neq f'_j(i)$, then D is discarded (Step 5 of Local Computation). Since we assume that D is not discarded, the claim follows.
Case 3: $P_i \notin UNHAPPY, Pj \in UNHAPPY$.
If $P_j \in UNHAPPY$, then $f'_j(x) = f_j^D(x)$. If $f'_i(j) \neq f_j^D(i)$, then P_i would have been deleted from REC. But by Claim 4.5, we have honest $P_i \in REC$. Therefore, the claim must hold.
Case 4: $P_i \in UNHAPPY, P_j \notin UNHAPPY$.
Switching P_i and P_j in the previous case, we see that the claim holds for this case. □

Claim 4.7 *If D is not discarded then all honest players are consistent with an unique t-degree symmetric bivariate polynomial.*

Proof: Note that there are at least $t+1$ honest players. Hence their shares (which are consistent by Claim 4.6) are sufficient to reconstruct a t-degree symmetric bivariate polynomial. □

Let us call this t-degree polynomial $F^H(x,y)$.

Claim 4.8 *If D is not discarded and $P_i \in REC$, then $f'_i(x)$ is consistent with $F^H(x,y)$.*

Proof: By Claim 4.4, for every $P_i \in UNHAPPY$, P_i's share is consistent with all the honest players' shares. This implies that $f'_i(x)$ is consistent with $F^H(x,y)$ and we are done. Now let $P_i \notin UNHAPPY$. Since $P_i \in REC$, we have $f'_i(j) = f^D_j(i)$ for every $P_j \in UNHAPPY$ (otherwise, P_i is deleted from REC). Therefore, if $f'_i(x)$ isn't consistent with $F^H(x,y)$, then $f'_i(j) \neq f'_j(i)$ must hold for some honest $P_j \notin UNHAPPY$. If $a_{ij} \neq b_{ji}$ or $a_{ji} \neq b_{ij}$, then P_i would execute $\mathsf{RevealVal}(D, P_i, f_i(j))$ and P_j would execute $\mathsf{RevealVal}(D, P_j, f_j(i))$ and a contradiction (if one exists) would have been detected. Since D was not discarded, we have $f'_i(j) = f'_j(i)$. In the reconstruction phase, P_i and P_j would have to reveal the same values as before (otherwise, they are deleted from REC) and hence, the claim holds. On the other hand, if $a_{ij} = b_{ji}$ and $a_{ji} = b_{ij}$, then we have two possible cases:
Case 1: P_i gradecasted the secret in $\mathsf{AuthVal}^{(2-7)}(P_i, P_j, r_{ij})$.
In this case, P_i reveals $f'_i(j)$. If P_j had gradecasted the secret in $\mathsf{AuthVal}^{(2-7)}(P_i, P_j, r_{ij})$, then P_j would have revealed $f'_j(i)$ and a contradiction (if one exists) would have been detected. Since D was not discarded, we have $f'_i(j) = f'_j(i)$. On the other hand, if P_j had not gradecasted $\perp$ in $\mathsf{AuthVal}^{(2-7)}(P_j, P_i, r_{ji})$, then P_i would have to reveal $r'_{ji} = r_{ji}$ (follows from **Correctness3**) in $\mathsf{RevealVal}(P_j, P_i, r_{ji})$. Since P_j is honest, $b_{ij} - r_{ji} = f'_j(i)$. If $P_i \in REC$, then $b_{ij} - r'_{ji} = f'_i(j)$. Since $r'_{ji} = r_{ji}$, this shows that $f'_i(j) = f'_j(i)$. Hence $f'_i(x)$ is consistent with $F^H(x,y)$.
Case 2: P_i gradecasted $\perp$ in $\mathsf{AuthVal}^{(2-7)}(P_i, P_j, r_{ij})$.
In this case, an honest P_j would successfully reveal r'_{ij} in $\mathsf{RevealVal}(P_i, P_j, r_{ij})$. Since $a_{ij} = b_{ji} = f'_j(i) + r'_{ij}$, P_i would have to reveal $f'_i(x)$ such that $f'_i(j) = f'_j(i)$, otherwise $a_{ij} \neq f'_i(j) + r'_{ij}$, and P_i will be deleted from REC. □

Claim 4.9 *If D is not discarded, then $F^H(x,y)$ will be reconstructed in the reconstruction phase. Moreover, this $F^H(x,y)$ is fixed at the end of the sharing phase.*

Proof: By Claim 4.8, every $P_i \in REC$ reveals $f'_i(x)$ that is consistent with $F^H(x,y)$. Hence in the reconstruction phase, $F^H(x,y)$ will be reconstructed. $F^H(x,y)$ can be computed from the joint view of the honest players at the end of the sharing phase. Hence it is fixed at the end of the sharing phase. □

Claim 4.10 *If D is honest, then D will not be discarded.*

Proof: We prove that none of the rules for "discarding" D, apply to an honest D.

1. Since D is honest, he will give correct t-degree polynomials to every player. Hence he will never have to broadcast more than t polynomials.

2. Since D is honest, all polynomials broadcasted are consistent with a symmetric bivariate degree t polynomial $F(x, y)$.

3. For an honest D, $a_{ij}^D = f_i(j) + r_{ij}^D = f_j(i) + r_{ij}^D = b_{ij}^D$.

4. For every P_j, no player P_i can successfully reveal $f_i'(j) \neq f_i(j)$ (follows from **Correctness3**). And since D is honest, all his broadcasted polynomials are consistent with successfully revealed values.

5. For every P_j, no player P_i can successfully reveal $f_i'(j) \neq f_i(j)$. This follows from **Correctness3**.

6. By **Correctness2**, D successfully reveals $r_{ij}^D = r_{ij}$. For an honest D, $a_{ij}^D - r_{ij}^D = f_i^D(j) = f_j^D(i)$.

This completes the proof. □

Lemma 4.6 (Correctness) *Protocol* 10-Round-VSS *satisfies* $(1 - \varepsilon)$*-correctness property.*

Proof: Correctness follows from Claims 4.9 and 4.10. It is to be noted that $F^H(0, 0) = s$. □

Lemma 4.7 (Strong Commitment) *Protocol* 10-Round-VSS *satisfies* $(1-\varepsilon)$*-strong commitment property.*

Proof: The proof follows from Claim 4.9. □

Efficiency of the protocol is obvious. Hence the theorem follows from Lemmas 4.5, 4.6, and 4.7.

Chapter 5

Conclusions and Future Work

In this work, we studied ICP and VSS protocols for $t < n/2$ setting. We also focused on the statistical versions of ICP and VSS. We reduced the number of broadcast in ICP as well as VSS using the gradecast technique. The modified ICP was found to be satisfying all the required properties of an Information Checking Protocol. The proposed new VSS protocol is also found to be better than the VSS given by Garay et al. [6] in terms of overall round complexity. Our VSS protocol consists of ten rounds in the sharing phase as compared to 20 rounds in their VSS. Though the broadcast round complexity is same in both the protocols.

Currently, VSS is being studied for all kinds of complexities, i.e. round, broadcast and overall communication complexities. Hence it is worthwhile to pursue some research to bring down all the types of existing complexities simultaneously.

Bibliography

[1] Benny Chor, Shafi Goldwasser, Silvio Micali, and Baruch Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 383–395. IEEE, 1985.

[2] Danny Dolev. The byzantine generals strike again. *Journal of algorithms*, 3(1):14–30, 1982.

[3] Paul Feldman and Silvio Micali. Optimal algorithms for byzantine agreement. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 148–161. ACM, 1988.

[4] Matthias Fitzi. *Generalized communication and security models in Byzantine agreement*. PhD thesis, Swiss Federal Institute of Technology, 2002.

[5] Matthias Fitzi, Juan Garay, Shyamnath Gollakota, C Pandu Rangan, and Kannan Srinathan. Round-optimal and efficient verifiable secret sharing. In *Theory of Cryptography*, pages 329–342. Springer, 2006.

[6] Juan Garay, Clint Givens, Rafail Ostrovsky, and Pavel Raykov. Broadcast (and round) efficient verifiable secret sharing. In *Information Theoretic Security*, pages 200–219. Springer, 2014.

[7] Rosario Gennaro, Yuval Ishai, Eyal Kushilevitz, and Tal Rabin. The round complexity of verifiable secret sharing and secure multicast. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 580–589. ACM, 2001.

[8] Martin Hirt and Pavel Raykov. On the complexity of broadcast setup. In *Automata, Languages, and Programming*, pages 552–563. Springer, 2013.

[9] Mitsuru Ito, Akira Saito, and Takao Nishizeki. Secret sharing scheme realizing general access structure. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, 72(9):56–64, 1989.

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

[11] Arpita Patra, Ashish Choudhary, and Chandrasekharan Pandu Rangan. Simple and efficient asynchronous byzantine agreement with optimal resilience. In *Proceedings of the 28th ACM symposium on Principles of distributed computing*, pages 92–101. ACM, 2009.

[12] Arpita Patra, Ashish Choudhary, and C Pandu Rangan. Round efficient unconditionally secure multiparty computation protocol. In *Progress in Cryptology-INDOCRYPT 2008*, pages 185–199. Springer, 2008.

[13] Tal Rabin and Michael Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 73–85. ACM, 1989.